



UNIVERSITÄT  
KOBLENZ · LANDAU

Fachbereich 4: Informatik



# Untersuchungen zum automatisierten Tracking der Bewegungsabläufe von Sportkletterern

Bachelorarbeit  
zur Erlangung des Grades  
BACHELOR OF SCIENCE  
im Studiengang Computervisualistik

vorgelegt von

Eva Brumme

**Betreuer:** Dipl.-Inform. Peter Decker, Institut für Computervisualistik,  
Fachbereich Informatik, Universität Koblenz-Landau

**Erstgutachter:** Prof. Dr.-Ing. Dietrich Paulus, Institut für  
Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

**Zweitgutachter:** Dipl.-Inform. Peter Decker, Institut für  
Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Koblenz, im Dezember 2010



## **Kurzfassung**

Im Institut MTI Mittelrhein wird untersucht, wie sich Training und Erfahrung auf die Bewegung und Bewegungsplanung im Sport auswirken. Im Rahmen dieser Untersuchungen soll das Bewegungsverhalten eines Kletterers an einer Sportkletterwand analysiert werden. Das Ziel der Bachelorarbeit ist es, die Bewegungen eines Kletterers in Videoaufnahmen zu segmentieren und markerbasiert zu detektieren. Die segmentierten Marker aus jedem Bild werden mit einem Trackingverfahren über die Zeit verfolgt und die Bewegungsspuren in einem geeigneten Format für die Bewegungsanalyse-Software Simi Motion exportiert, um dort weitere Analysen zu ermöglichen.

## **Abstract**

In the “Institut MTI Mittelrhein” the impact of training and experience on motion and motion planning in sports is examined. As a part of these investigations the motion behaviour of a sports climber is analyzed on a sports climbing wall. The aim of this Bachelor’s thesis is the segmentation and the marker-based detection of the movement of a climber in video recordings. The segmented markers from each image are tracked by a tracking procedure over time. The motion tracks will be exported to the motion analysis software Simi Motion in a format suitable for further analysis.



## Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja  nein

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja  nein

Koblenz, den 30. Dezember 2010



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>13</b>
1.1	Aufbau der Arbeit . . . . .	14
<b>2</b>	<b>Stand der Technik</b>	<b>15</b>
2.1	Marker-Erkennung . . . . .	15
2.2	Simi Motion . . . . .	16
2.3	Tracking mit Partikelfiltern . . . . .	18
<b>3</b>	<b>Bayes-Tracking mit Partikelfiltern</b>	<b>19</b>
3.1	Einführung . . . . .	19
3.2	Mathematischer Hintergrund . . . . .	19
3.3	Der Condensation-Algorithmus . . . . .	21
3.3.1	Bewegungsmodell . . . . .	21
3.3.2	Beobachtungsmodell . . . . .	22
3.3.3	Ablauf des Algorithmus . . . . .	22
<b>4</b>	<b>Bewegungsverfolgung eines Kletterers</b>	<b>25</b>
4.1	Videoaufnahmen an der Kletterwand . . . . .	25
4.2	Markerdetektion . . . . .	26
4.2.1	Vorverarbeitung . . . . .	27
4.2.2	Konturfindung . . . . .	28
4.2.3	Aussortierung . . . . .	29
4.2.4	Berechnung des Schwerpunktes . . . . .	29
4.3	Tracking . . . . .	31
4.3.1	Zustandsmodell . . . . .	31
4.3.2	Bewegungsmodell . . . . .	31
4.3.3	Beobachtungsmodell . . . . .	32
4.3.4	Tracking-Funktionalität . . . . .	32
4.4	Export für Simi Motion . . . . .	33

<b>5</b>	<b>Experimente und Evaluation</b>	<b>39</b>
5.1	Evaluation der Markerdetektion . . . . .	39
5.1.1	Erster Test . . . . .	40
5.1.2	Zweiter Test . . . . .	42
5.2	Tracking . . . . .	43
5.2.1	Vergleich der Trackingergebnisse . . . . .	43
5.2.2	Bewertung . . . . .	45
<b>6</b>	<b>Zusammenfassung</b>	<b>51</b>
6.1	Grenzen der Verfahren . . . . .	51
6.2	Ausblick . . . . .	52
<b>A</b>	<b>Herstellung von Reflektionsmarkern</b>	<b>55</b>
<b>B</b>	<b>Programmstrukturen und Benutzungshinweise</b>	<b>57</b>
B.1	Struktur . . . . .	57
B.2	Vorbereitung der Bilddaten . . . . .	58
B.3	Schnittstellen . . . . .	59
B.4	Bedienung der Tracking-Software . . . . .	59



# Tabellenverzeichnis

5.1	Parameterwerte der Markerdetektion bei der Evaluation . . . . .	40
5.2	Statistiken des ersten Tests . . . . .	42
5.3	Statistiken des zweiten Tests . . . . .	45



# Abbildungsverzeichnis

2.1	Bewegungsanalyse in Simi Motion . . . . .	17
2.2	Falsche Marker-Erkennung in Simi Motion . . . . .	18
3.1	Zeitschritt des Condensation-Algorithmus . . . . .	24
4.1	Systemübersicht . . . . .	26
4.2	Vergleich zweier Bilder, mit und ohne Infrarotfilter . . . . .	27
4.3	Vergleich zweier Binärbilder, mit und ohne angewandtem Closing . . . . .	28
4.4	Vergleich zweier Konturbilder, vor und nach der Verwerfung ungewünschter Konturen . . . . .	30
4.5	Screenshot des Tracking-Programms, Initialisierung . . . . .	34
4.6	Screenshot des Tracking-Programms, korrektes Tracking mit zwei Partikelfiltern . . . . .	35
4.7	Screenshot des Tracking-Programms, fehlerhaftes Tracking mit vier Partikelfiltern . . . . .	36
4.8	Import einer Textdatei in Simi Motion . . . . .	37
5.1	Erscheinungsformen von Markern im Klettervideo . . . . .	40
5.2	Videoframe mit Ergebnis der Markerdetektion . . . . .	41
5.3	Erkennungsrate von Markern in Bildern des Videos aus Test 1 . . . . .	43
5.4	Videoframe mit Ergebnis der Markerdetektion . . . . .	44
5.5	Erkennungsrate von Markern in Bildern des Videos aus Test 2 . . . . .	46
5.6	Positionsdaten aller in Simi Motion detektierten Marker . . . . .	47
5.7	Vergleich von in Simi Motion erfassten Markerdaten mit guten Ergebnissen aus dem Tracking-Programm . . . . .	48
5.8	Vergleich von in Simi Motion erfassten Markerdaten mit unsicheren Ergebnissen aus dem Tracking-Programm . . . . .	48
5.9	Vergleich von in Simi Motion erfassten Markerdaten mit Ergebnissen aus dem Tracking-Programm, basierend auf einer Hochgeschwindigkeitsaufnahme . . . . .	49
A.1	Verschiedene Reflektionsmarker . . . . .	56



# Kapitel 1

## Einleitung

Das Institut für Medizintechnik und Informationsverarbeitung Mittelrhein (MTI) ist ein Kooperationsinstitut der Universität Koblenz, der FH Koblenz und den medizinischen Kliniken in Koblenz. Im MTI Mittelrhein wird untersucht, wie sich Training und Erfahrung auf die Bewegung und Bewegungsplanung im Sport auswirken.

Im Rahmen dieser Bachelorarbeit wird ein Ansatz zum Verfolgen der Bewegungen eines Kletterers in Videoaufnahmen vorgestellt. Langfristig soll es möglich sein, Bewegungsmuster des Kletterers an der Kletterwand zu erkennen und zu Trainingszwecken zu analysieren. Für die Bewegungsverfolgung werden die Bewegungen markerbasiert detektiert und die Verfolgung der Marker auf Basis eines Partikelfilters realisiert. Durch die Nutzung des stochastischen Verfahrens der Partikelfilter können bei der Markerverfolgung auch fehlerhafte bzw. unvollständige Daten verarbeitet werden. Damit eröffnen sich in der Praxis vielfältige Möglichkeiten zur Bewegungsanalyse mit begrenztem Aufwand, d.h. die Anforderungen an Kamertechnik, Lichtverhältnisse und Markerqualität sind weniger hoch als bei klassischen Verfahren.

Diese Arbeit konzentriert sich auf die Detektion und Verfolgung der Marker. Im Ergebnis werden die segmentierten Marker aus jedem Bild für die Software Simi Motion [Sim] exportiert, um dort weitere Bewegungsanalysen zu ermöglichen.

Als Datenbasis für die durchgeführten Untersuchungen dienen Videosequenzen eines Kletternden, die im Rahmen der Arbeit aufgenommen wurden. Nach der Zerlegung der Aufnahmen in Einzelbilder wird die weitere Verarbeitung in folgenden Schritten durchgeführt:

1. Detektion der Marker mittels Binarisierung, Untersuchung der Konturparameter und Berechnung der Schwerpunkte
2. Tracking mittels Partikelfiltern

3. Export der Bewegungen in einem geeigneten Format für die Bewegungsanalyse-Software Simi Motion

Die Implementierung erfolgt mit Hilfe von OpenCV in C++ unter GNU/Linux.

## 1.1 Aufbau der Arbeit

Im Folgenden wird ein kurzer Überblick über die einzelnen Kapitel der Arbeit gegeben.

Nach diesem einleitenden Kapitel wird in Kapitel 2 der Stand der Technik im Bereich der Markerdetektion und Objektverfolgung beschrieben. Die Grundlagen zum Verständnis des Tracking mit Partikelfiltern werden anschließend in Kapitel 3 erklärt. Eine detaillierte Vorstellung des eigenen Ansatzes erfolgt in Kapitel 4. Zum Ende wird das entwickelte Verfahren zur Markerdetektion evaluiert und experimentelle Vergleiche zwischen dem Ergebnis der Markerverfolgung mit Partikelfiltern und dem Ergebnis der automatischen Markerverfolgung in Simi Motion gezogen. Die Evaluation und Experimente werden in Kapitel 5 beschrieben. Es folgt eine abschließende Zusammenfassung sowie ein kurzer Ausblick in Kapitel 6.

# Kapitel 2

## Stand der Technik

Dieses Kapitel stellt die in der Arbeit verwendeten Verfahren vor und beschreibt ihre Einsatzmöglichkeiten anhand anderer Arbeiten.

### 2.1 Marker-Erkennung

Die markerbasierte Verfolgung menschlicher Bewegungen in Bildsequenzen kann in die Verwendung von aktiven und passiven Markern unterteilt werden [FLB03], auch bezeichnet als aktive und passive optische Systeme. Aktive Marker basieren auf LEDs und benötigen eine Stromversorgung. Da so im Ergebnis Marker und Hintergrund einen großen Kontrast zueinander haben, wird die Detektion vereinfacht. Auch die Verfolgung der Marker ist einfacher, weil nur eine LED auf einmal aktiviert ist und so keine Unterscheidung der Marker notwendig ist. Ein Nachteil bleibt damit die notwendige Stromversorgung und der Aufwand zur Ansteuerung der LED, was letztlich in einem hohen Preis der aktiven Marker im Vergleich zu passiven Markern resultiert.

Passive optische Systeme verwenden kugelförmige Marker, die mit einem retro-reflektierenden Material überzogen sind, um Licht zu reflektieren, das neben der Kameralinse generiert wurde. Passive Systeme erfordern im Gegensatz zu aktiven Systemen kein elektronisches Equipment am Körper und dennoch zeigen die gelieferten Bilder einen hohen Kontrast zwischen Markern und Hintergrund. Durch die simultane Erscheinung der Marker wird für die Verfolgung eine Identifikation der Marker in den Bildsequenzen notwendig.

In dieser Arbeit wird ein passives optisches System zur Bewegungsverfolgung verwendet. Der Ansatz zur Detektion von Markern besteht hier zunächst in der Anwendung einer Binarisierung mit einem flexiblen Schwellwert, der sich aus dem kumulativen Histogramm des zu untersuchenden Bildes berechnet. Der angewandte Schwellwert orientiert sich pro Bild demnach an dessen Helligkeit. [Han08] ver-

wendet zur Markerdetektion einen adaptiven Schwellwert, um ähnlich hell reflektierende Bereiche von den Markern zu trennen. Er wird berechnet aus dem Mittelwert einer Nachbarschaft mit einer Konstanten. Zugleich wird ein normaler, niedrig gewählter Schwellwert auf das gaußgeglättete Bild angewandt, um den Hintergrund auszuschließen. Die Ergebnisbilder werden mit einer Durchschnittsoperation zusammengeführt.

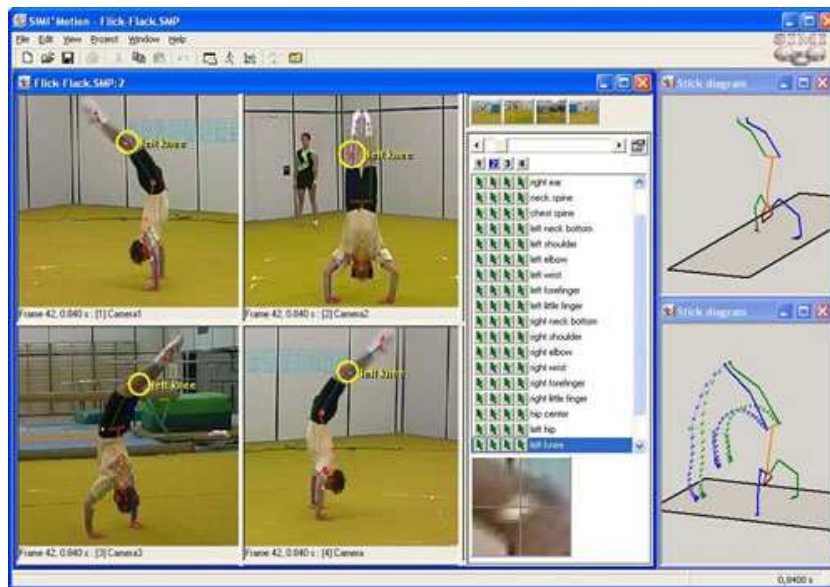
Nach der Schwellwertbestimmung wird im hier vorgestellten Verfahren Closing angewandt. Morphologische Operationen sind häufig verwendete Methoden bei der Markerdetektion, die die Form von Strukturen in einem Bild verändern. In [FLB03] werden sie neben einfacher Vorverarbeitung verwendet, um ein Gradientenbild zu berechnen. Auf dieses wird eine Wasserscheiden-Transformation zur Segmentierung angewandt. Die sich ergebenden Konturen werden auf verschiedene Kriterien wie Größe und Umfang untersucht und gefiltert. Die Untersuchung der sogenannten Konturparameter erfolgt bei [FLB03] analog zum Ansatz in dieser Arbeit, nur wird die Kontur jeweils unterschiedlich erzeugt. Nach der Bewertung der Kontureigenschaften werden ihre Schwerpunkte als resultierende Markerkoordinaten berechnet.

## 2.2 Simi Motion

*Simi Motion* ist ein Produkt der Firma Simi Reality Motion Systems GmbH [Sim]. Das Unternehmen bietet Software zur Bewegungsanalyse mit Kinetik und 2D/3D-Kinematik, automatische Objektverfolgung, sowie Software zur Kommunikations- und Verhaltensanalyse für Anwendungen in Wissenschaft, Medizin, Sport, Industrie und Unterhaltung. Das in dieser Arbeit zur Bewegungsanalyse verwendete Produkt *Simi Motion* ermöglicht, 2D oder 3D Bewegungen videobasiert zu erfassen und zu analysieren. Die Software besteht aus einzelnen Modulen, die in einer gemeinsamen Benutzeroberfläche integriert sind. Die Basisfunktionen umfassen Algorithmen zur Objektverfolgung mit und ohne Markern sowie die synchrone Anzeige von Kameraaufzeichnungen und Daten. Weitere Module bieten unter anderem Messungen zur Druckverteilung sowie die Erstellung von Modellen zur Berechnung des Körperschwerpunkts.

Nach dem Import der zu analysierenden Videoaufnahmen in *Simi Motion*, wird ein Tracking-Verfahren angewandt, um die Bewegungsdaten zu erfassen. Dafür ist es zunächst erforderlich, die Marker per Mausklick anzuwählen und somit zu initialisieren. Bei der Verwendung von kugelförmigen Markern bietet sich das *automatische Tracking* an. Es sucht pro gewähltem Marker, ausgehend vom Zentrum der letzten bekannten Position, runde, markerartige Flächen im Bild radial und iterativ über mehrere Farbräume. Hierzu gibt es eine Option zur Anpassung von Tracking-Kriterien wie Farbe, Fläche und Weg, deren Einflussnahme prozentual



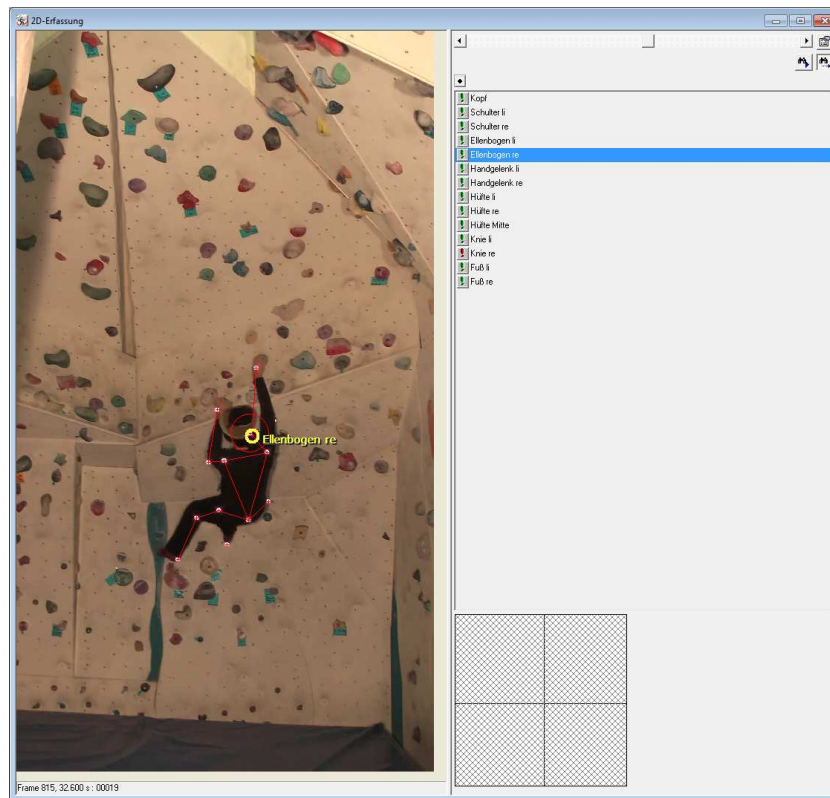


**Abbildung 2.1:** Mögliche Bewegungsanalyse in Simi Motion. Hier die Untersuchung der Bewegungen beim Flick-Flack, aufgenommen mit 4 Kameras. Die Daten können in Strichdarstellung animiert werden. Quelle: [Sim]

angegeben werden kann. Eine andere Möglichkeit zur Erfassung der Bewegungsdaten ist die Anwendung des *Patternmatching*-Algorithmus, der beliebige durch den Benutzer als Muster festgelegte Bildausschnitte erkennt. Auch hier können verschiedene Einstellungen vorgenommen werden. So kann die Berechnung der Ähnlichkeit entweder über eine normierte Kreuzkorrelation erfolgen oder über die Abweichung vom Mittelwert, und bestimmt werden, ob das vorhergehende oder das erste Bild als Referenz verwendet wird.

Die erfassten Daten können anschließend auf verschiedene Weise analysiert und visualisiert werden, beispielsweise über Diagramme, Strichdarstellungen und 3D-Darstellungen. Zur Weiterverarbeitung können die Daten für Statistik- und Animationsprogramme wie Matlab, Excel, Maya und Cinema 4D exportiert werden.

Um aussagekräftige Analyseergebnisse zu erhalten, wird empfohlen, die Bewegungen mit mehreren Kameras aus verschiedenen Blickrichtungen aufzunehmen, so dass alle Marker erfasst und nicht verdeckt werden. Die hier verwendete Version von Simi Motion ist nicht dafür ausgelegt, mit Verdeckungen von Markern umgehen zu können. Die Funktion zur automatischen Erfassung sucht ausschließlich zuvor ausgewählte Marker in dem Bild und betrachtet sie endgültig als verloren, sobald sie nicht mehr sichtbar sind. Dies trifft bereits für Teilverdeckungen von Markern sowie Bewegungsunschärfen zu. In dem Fall werden Marker falsch erkannt



**Abbildung 2.2:** Falsche Erkennung des Ellenbogens, da der zugehörige Marker nicht vollständig sichtbar ist. Der Marker des rechten Knies wird komplett verdeckt.

und falsch weiter verfolgt (vgl. Abbildung 2.2). Eine Korrektur der Markerposition muss manuell erfolgen.

## 2.3 Tracking mit Partikelfiltern

Als Alternative zum Ansatz von Simi Motion wird die Markerverfolgung in dieser Arbeit mit Partikelfiltern durchgeführt. Partikelfilter sind ein probabilistisches Verfahren zur Verfolgung von Objekten. Durch die Schätzung potentieller Objektzustände ist es mit Partikelfiltern möglich, Objekte sicher zu verfolgen, auch wenn sie kurzzeitig verdeckt werden. In [Sch05] wird beispielsweise mit Hilfe eines Partikelfilters ein Objektverfolgungssystem implementiert, das Autos, Fahrradfahrer, Fußgänger und Vögel in einem Video erkennen und verfolgen soll.

Partikelfilter sind in der Lage, eine Wahrscheinlichkeitsaussage über einen zukünftigen Objektzustand zu treffen. In [Wir07] wird die Lage und Ausrichtung von Unterwasser-Kabeln in Kamerabildern mit Hilfe eines Partikelfilters vorhergesagt.

Die zuverlässige automatische Detektion der Kabel ermöglicht es, Unterwasserfahrzeuge autonome Kontrollfahrten durchführen zu lassen.

Im nächsten Kapitel werden Partikelfilter detailliert vorgestellt.



# Kapitel 3

## Bayes-Tracking mit Partikelfiltern

Dieses Kapitel erläutert das probabilistische Tracking mit Partikelfiltern. Mathematischer Hintergrund ist das Bayes-Theorem, das zu Beginn erklärt werden soll. Es folgt eine Beschreibung des Condensation-Algorithmus, der das Prinzip des Partikelfilters implementiert.

### 3.1 Einführung

In dieser Arbeit wird untersucht, wie die Bewegungen eines Kletterers markerbasiert in Videos erkannt, verfolgt und einem Gelenk zugeordnet werden können. Die Erkennung und Verfolgung der Marker wird auf Einzelbildern realisiert. Beim Tracking mit Partikelfiltern wird der Objektzustand durch Wahrscheinlichkeitsvorhersagen geschätzt, wofür verschiedene Parameter des Systems betrachtet werden. Anhand eines objektspezifischen Bewegungsmodells wird versucht, die nächste Position des Objektes im Bild zu schätzen. Da dies niemals genau vorhergesagt werden kann, werden eine Vielzahl von Hypothesen angenommen. Sie beschreiben mögliche Objektzustände und versuchen, jede mögliche Veränderung gegenüber letzten Messung vorherzusagen. Eine Hypothese wird dabei als ein Partikel bezeichnet.

### 3.2 Mathematischer Hintergrund

Das Theorem von Bayes stellt die mathematische Grundlage des Partikelfilters dar und beschreibt, wie mit bedingten Wahrscheinlichkeiten zu rechnen ist. Zur näheren Erläuterung werden dafür zwei Ereignisse  $A$  und  $B$  angenommen. Die Wahrscheinlichkeit für das Eintreten des Ereignisses  $A$ , angenommen es beschreibt eine mögliche Objektposition, ist  $P(A)$  und wird A-priori-Wahrscheinlichkeit genannt. Dies ist ein Wahrscheinlichkeitswert, der aufgrund von Vorwissen gewonnen

werden kann.  $P(B)$  ist die A-priori-Wahrscheinlichkeit für das Eintreten eines Ereignisses B, und beschreibt beispielsweise die gelieferte Bildinformation.

Man möchte nun die Wahrscheinlichkeit von Ereignis A wissen, unter der Bedingung, dass Ereignis B eingetreten ist. Sie wird mit  $P(A|B)$  notiert und repräsentiert die Wahrscheinlichkeit, mit der sich das Objekt an einer bestimmten Stelle im Bild befindet. Da es schwierig ist, die Wahrscheinlichkeit  $P(A|B)$  direkt zu bestimmen, wird dies durch die einfachere Bestimmung, ob die Bildinformation der vermuteten Position entspricht,  $P(B|A)$  ausgedrückt und folgendermaßen im Bayes-Theorem formuliert [Koc00]:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (3.1)$$

$P(B|A)$  beschreibt demnach die Wahrscheinlichkeit für das Ereignis B unter der Bedingung, dass A eingetreten ist. Sie wird als A-posteriori-Wahrscheinlichkeit bezeichnet, da die Wahrscheinlichkeit nicht durch Vorwissen, sondern aus einer Reihe von Zufallsexperimenten ermittelt wird.  $P(A)$  gibt das Vorwissen mit der Auftrittswahrscheinlichkeit von A ohne weiteres Wissen um B an.

In der allgemeinen Form des Bayes-Theorems werden statt wie bisher konkrete Ereignisse Verteilungen stetiger und diskreter Zufallsvektoren untersucht. Sie werden für die Verwendung in Bildsequenzen formuliert, wobei jedes Bild der Kamera einem Zeitschritt entspricht. Objektzustände zum Zeitpunkt  $t$  werden ab sofort in einem Zustandsvektor  $\mathbf{x}_t$  zusammengefasst und die Messungen im Zeitschritt  $t$  im Messvektor  $\mathbf{z}_t$  angegeben. Die A-posteriori-Wahrscheinlichkeit wird als  $p(\mathbf{x}_t|\mathbf{z}_t)$  notiert, mit der zugleich Informationen über Objektzustände im Folgebild einer Sequenz ermittelt werden. Nach dem Bayes-Theorem erhält man somit [DFG01]:

$$p(\mathbf{x}_t|\mathbf{z}_t) = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{z}_t)} = \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t)}{\int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t)d\mathbf{x}_t} \quad (3.2)$$

$p(\mathbf{z}_t)$  sind hier die Beobachtungen im aktuellen Zeitschritt und  $p(\mathbf{z}_t|\mathbf{x}_t)$  ist die vordefinierte Ähnlichkeitsfunktion. Die Bestimmung von  $p(\mathbf{x}_t|\mathbf{z}_t)$  geschieht rekursiv in zwei Schritten, der *Prädiktion* und *Aktualisierung* [AMGC02]. Im Prädiktions-schritt werden die neuen Parameterwerte anhand eines erstellten Bewegungsmodells geschätzt. Die potentiellen Veränderungen des Objektzustandes  $\mathbf{x}$  im Verlauf eines Zeitschritts werden als Hypothesen mit  $p(\mathbf{x}_t|\mathbf{x}_{t-1})$  aufgestellt. Ergebnis der Prädiktion ist das ungewichtete A-priori  $p(\mathbf{x}_t|\mathbf{z}_{t-1})$ , das sich berechnet durch

$$p(\mathbf{x}_t|\mathbf{z}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{t-1})d\mathbf{x}_{t-1} \quad (3.3)$$

Beim anschließenden Aktualisierungsschritt wird die aus dem Prädiktions-schritt berechnete A-priori-Wahrscheinlichkeit durch die Ähnlichkeitsfunktion gewichtet.

Man erhält die A-posteriori-Verteilung, die Grundlage zur Bildung der neuen A-priori-Verteilung ist. Der Aktualisierungsschritt erfolgt nach dem Bayes-Gesetz mit

$$p(\mathbf{x}_t | \mathbf{z}_t) = \frac{p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{t-1})}{\int p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{t-1}) dx_t} \quad (3.4)$$

### 3.3 Der Condensation-Algorithmus

In der vorliegenden Arbeit wird der Partikelfilter mit Hilfe des Condensation-Algorithmus implementiert. Der Algorithmus ist eine von [IB98] entwickelte Technik zum Verfolgen von Objektkonturen in Videos. Er verwendet definierte dynamische Modelle, um anhand aktueller Beobachtungen eine Aussage über die Wahrscheinlichkeitsverteilung möglicher zukünftiger Objektzustände zu treffen.

Der Zustand des zu untersuchenden dynamischen Systems wird anhand aller erhältlichen Informationen geschätzt, beispielsweise Form und Geschwindigkeit. Die Wahrscheinlichkeitsverteilung wird dabei durch eine diskrete Anzahl von gewichteten Proben approximiert, sogenannten Partikeln. Das Gewicht eines Partikels gibt seinen Anteil an der Gesamtwahrscheinlichkeit wieder und beschreibt einen bestimmten Objektzustand, das heißt es wird eine Aussage darüber getroffen, wie wahrscheinlich dieser Objektzustand für das aktuelle Bild ist. Von der Anzahl der Partikel ist abhängig, wie gut die Wahrscheinlichkeitsverteilung approximiert wird: Je mehr Partikel, um so exakter ist die Annäherung. Die optimale Anzahl an Partikeln hängt letztlich jedoch von der Menge zu schätzender Zustände ab, der sogenannten Dimension der Wahrscheinlichkeitsverteilung.

Der Condensation-Algorithmus ist problemunabhängig, da er nur für die Verwaltung der Partikel zuständig ist. Für die zeitabhängige Variation der Partikelwerte ist das dynamische Modell der Objektbewegung verantwortlich. Die Zuweisung einer Gewichtung an jedes Partikel wird weiterhin in einem Beobachtungsmodell verwaltet. Nachfolgend sollen die Prinzipien dieser Modelle erläutert werden.

#### 3.3.1 Bewegungsmodell

Das Bewegungsmodell beschreibt die Veränderung der Objektparameter über die Zeit und wird als eine Transformationsgleichung mit einem deterministischen und einem stochastischen Teil angegeben [Hom07]. Die Wahrscheinlichkeitsverteilung im aktuellen unbekanntem Objektzustand  $\mathbf{x}_t$  kann damit aus dem vorhergehenden Zustand  $\mathbf{x}_{t-1}$  ermittelt werden. Dies wird auch im Prädiktionsschritt in Gleichung 3.3 verwendet. In der deterministischen Komponente der Gleichung werden die erwarteten Parameterveränderungen sowie dessen Intensitäten beschrieben. Durch die stochastische Komponente wird anschließend eine Zufallsgröße addiert, mittels welcher die Zustandsvektoren um die prognostizierte Position gestreut werden.

In [IB98] wird das Bewegungsmodell als autoregressiver Prozess mit folgender formaler Gleichung modelliert:

$$\mathbf{x}_t - \bar{\mathbf{x}} = \mathbf{A}(\mathbf{x}_{t-1} - \bar{\mathbf{x}}) + \mathbf{B}\mathbf{w}_t \quad (3.5)$$

Danach ergibt sich die Abweichung des aktuellen Objektzustandes  $\mathbf{x}_t$  gegenüber dem Mittelwert der betrachteten Zustände  $\bar{\mathbf{x}}$  als Transformation aus dem vorhergehenden Zustand  $\mathbf{x}_{t-1}$ , der mittels der Matrix  $\mathbf{A}$  umgeformt wird. Zusätzlich erfolgt die Überlagerung eines zufälligen Rauschanteils  $\mathbf{B}\mathbf{w}_t$ . Durch die Überlagerung des Rauschanteils werden systematische Modellierungsfehler kompensiert. Dazu können beispielsweise Modellabweichungen aufgrund beschleunigter Bewegungsanteile oder Fehler des Messsystems gehören.

### 3.3.2 Beobachtungsmodell

Das Beobachtungsmodell beschreibt den Vergleich der geschätzten Zustände mit den Ergebnissen der Auswertung des aktuellen Bildes, das heißt den Ergebnissen des verwendeten Sensors. Es beurteilt die Aussagen des Bewegungsmodells auf ihre Übereinstimmung mit der realen Messung. Aus der Abweichung der beobachteten Zustände gegenüber den Schätzungen werden die bedingten Wahrscheinlichkeiten des Aktualisierungsschrittes aus Gleichung 3.4 ermittelt.

Wenn Beobachtung und Schätzung eine hohe Übereinstimmung zeigen, erhalten die entsprechenden Zustände im Modell ein hohes Gewicht. Zustände mit großen Abweichungen zwischen Schätzung und Beobachtung werden dementsprechend gering gewichtet. Die ermittelten Gewichte gehen in den folgenden Iterationsschritt zur Vorhersage des nächsten aktuellen Zustandes ein.

### 3.3.3 Ablauf des Algorithmus

Im Folgenden werden die einzelnen Schritte des Condensation-Algorithmus beschrieben, die sich iterativ wiederholen (vgl. Abbildung 3.1) und auf jedem Partikel angewandt werden.

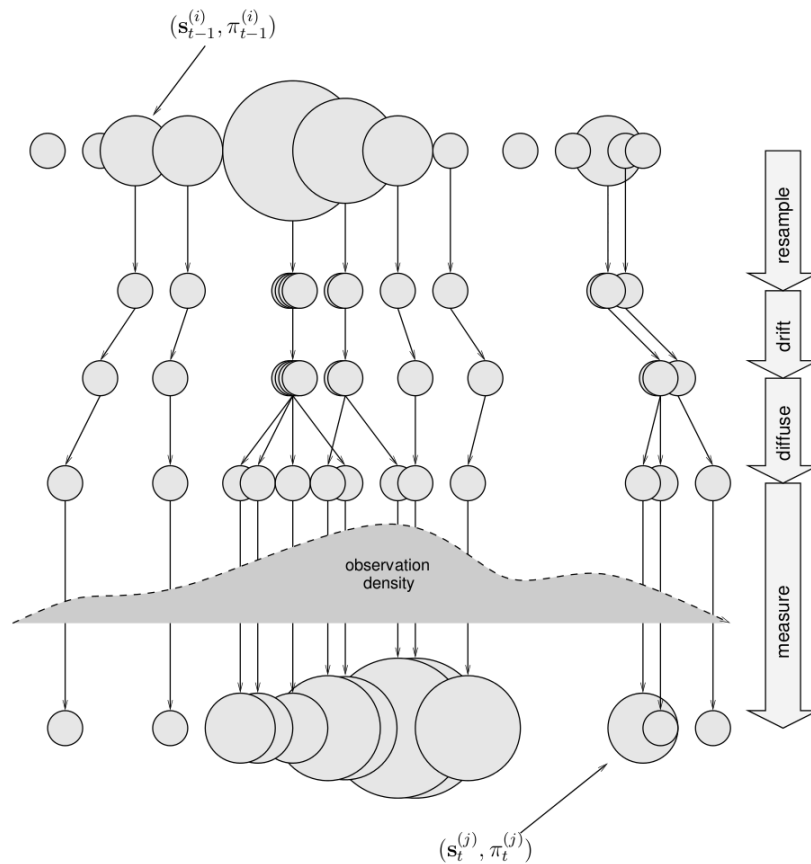
**Resampling** Im Resampling-Schritt wird eine neue Menge an Samples durch Ziehen mit Zurücklegen aus der alten Partikelmenge ausgewählt. Hoch gewichtete Partikel können dabei mehrmals, niedrig gewichtete Partikel auch überhaupt nicht ausgewählt werden. Partikel mit hoher Gewichtung bekommen dann mehrere Nachkommen, Partikel mit geringem Gewicht werden verworfen.

**Drift und Diffuse** Im Drift- und Diffuse-Schritt wird der nächste Zustand eines Partikels aus seinem aktuellen Zustand prognostiziert. Beim Drifting findet



eine Abweichung der Partikelzustände gemäß des deterministischen Teils des Bewegungsmodells statt und im Diffuse-Schritt werden die Partikel anschließend mit einem normalverteilten Rauschen überlagert. So werden eventuelle Messfehler des Bewegungsmodells ausgeglichen.

**Measure** Die prognostizierten Partikelzustände werden anhand von aktuellen Messdaten mit Hilfe des Beobachtungsmodells rekursiv neu gewichtet.



**Abbildung 3.1:** Ein Zeitschritt des Condensation-Algorithmus. Die Gewichtung bzw. Wahrscheinlichkeit  $\pi$  eines Partikels  $\mathbf{s}$  wird durch die Kreisgröße repräsentiert. Die erste Zeile an Kreisen entspricht der Partikelrepräsentation zur Startverteilung. Die zweite Zeile zeigt die Partikel nach dem Resampling, wobei Partikel mit hoher Gewichtung vervielfacht werden. In der dritten Zeile wird der Drift-Schritt angewandt und in der vierten Zeile der Diffuse-Schritt. Im letzten Schritt wird die neue Messung (observation density) verwendet, um die Partikel neu zu gewichten. Quelle: [Wir07]

# Kapitel 4

## Bewegungsverfolgung eines Kletterers

Nachdem im letzten Kapitel eine allgemeine Einführung in die Grundlagen des Partikelfilters gegeben wurde, soll nun das entwickelte System zur markerbasierten Verfolgung der Bewegungen eines Kletterers vorgestellt werden. Es teilt sich in ein Programm zur Markererkennung und zur Markerverfolgung durch Partikelfilter-Tracking (vgl. Abbildung 4.1). Bevor jedoch näher auf die Module eingegangen wird, erfolgt ein kurzer Einblick in die Vorbereitung der Videoaufnahmen gegeben.

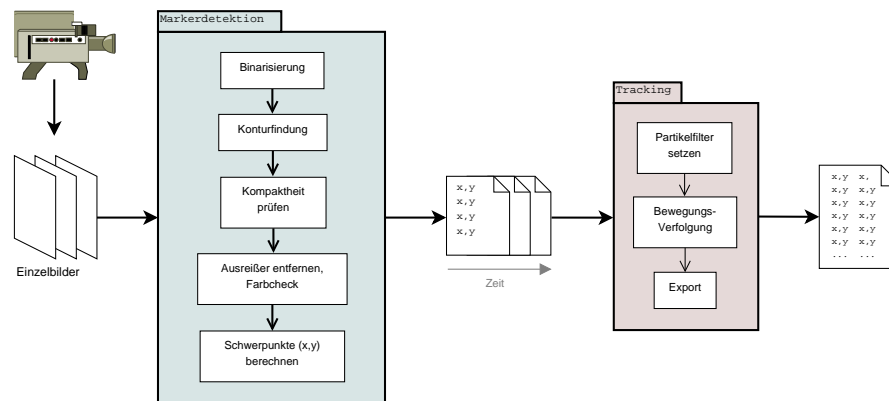
### 4.1 Videoaufnahmen an der Kletterwand

Um mit passendem Videomaterial arbeiten zu können, wurden verschiedene Aufnahmen eines Kletterers im Boulderbereich der Kletterhalle Kanditurm in Andernach durchgeführt [Kan].

Bei der Kamera handelte es sich um eine Panasonic HDC-SD300 mit einer Auflösung von 1920 x 1080 Pixeln. Sie wurde auf einem Stativ in etwa 1,5m Höhe angebracht und so platziert, dass der Kletterbereich (keine spezifische Route) vollständig im Bild war. Der Kletterer wurde rückseitig mit kugelförmigen, lichtreflektierenden Markern an den Gelenken prepariert und beim Klettern von zwei starken Scheinwerfern links und rechts der Kamera angestrahlt. Durch die schwarze Kleidung des Kletterers ergab sich ein starker Kontrast zwischen Hintergrund und Markern.

Es wurden an verschiedenen Wänden Videos mit folgenden Einstellungen aufgenommen:

- Leeraufnahme des Raumes
- Kletter-Sequenzen mit und ohne Marker



**Abbildung 4.1:** Übersicht der entwickelten Systeme zur Markerdetektion und Marker-  
verfolgung. Eine zu verarbeitende Videoaufnahme wird in Einzelbilder zerlegt und jedes  
Bild im Programm zur Markerdetektion verarbeitet. Dieses speichert im Bild gefundene  
Markerkoordinaten in einer Textdatei ab. Die Textdateien aller Bilder werden anschlie-  
ßend zusammen in das Tracking-Programm geladen, um die Markerkoordinaten zu visua-  
lisieren und mit Partikelfiltern zu verfolgen. Die Spuren der Partikelfilter werden über  
die Zeit in einer weiteren Textdatei abgespeichert.

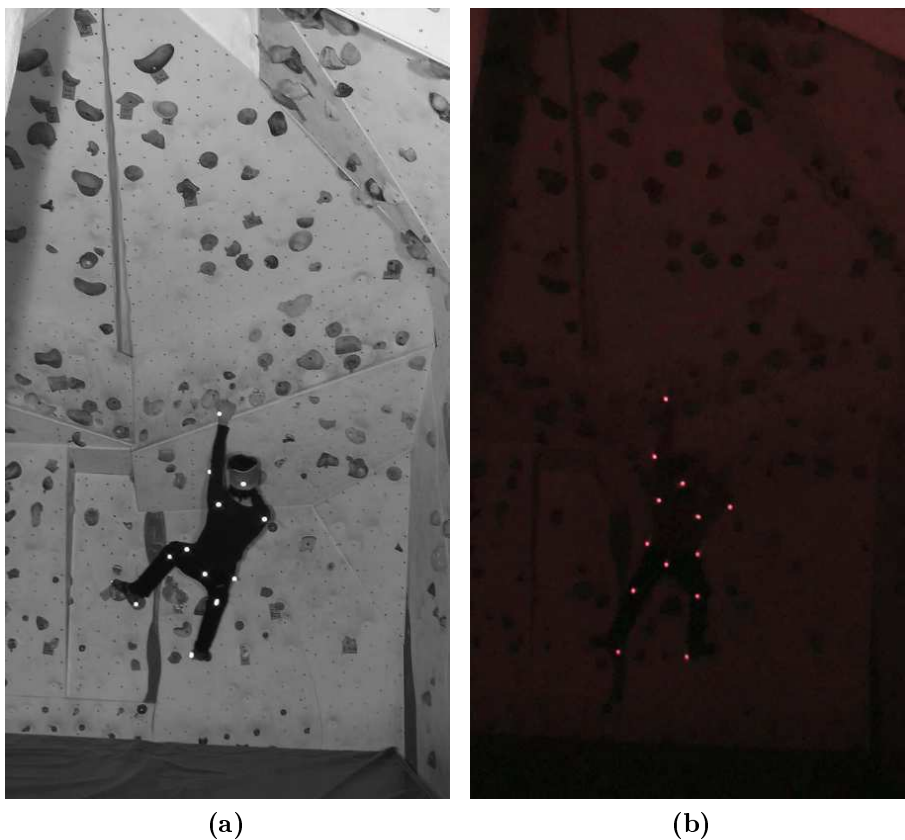
- Markeraufnahmen mit zusätzlichem Infrarotfilter, um den Kontrast zwischen reflektierenden Markern und der Umgebung zu erhöhen

Der Infrarotfilter bestand aus einem Stück Fotofilm, der vor die Linse geklemmt wurde. Bei der Auswertung der Aufnahmen zeigte sich, dass bereits die Aufnahmen ohne Infrarotfilter hervorragende Ergebnisse liefern (vgl. Abbildung 4.2).

Beim Klettern ergibt sich eine natürliche temporäre Verdeckung von Markern, beispielsweise wenn ein Griff direkt vor dem Oberkörper gewählt wird und somit der Marker am Handgelenk verdeckt wird. Später soll gezeigt werden, dass Partikelfilter in der Lage sind, die Bewegungen temporär verdeckter Objekte abzuschätzen.

## 4.2 Markerdetektion

Die Erkennung der Marker erfolgt über Binarisierung der Einzelbilder und anschließende Berechnungen auf den entsprechenden Konturbildern. Ausgabe des Programms ist, wie in Abbildung 4.1 dargestellt, für jeden Videoframe eine Textdatei mit den Koordinaten gefundener Marker.



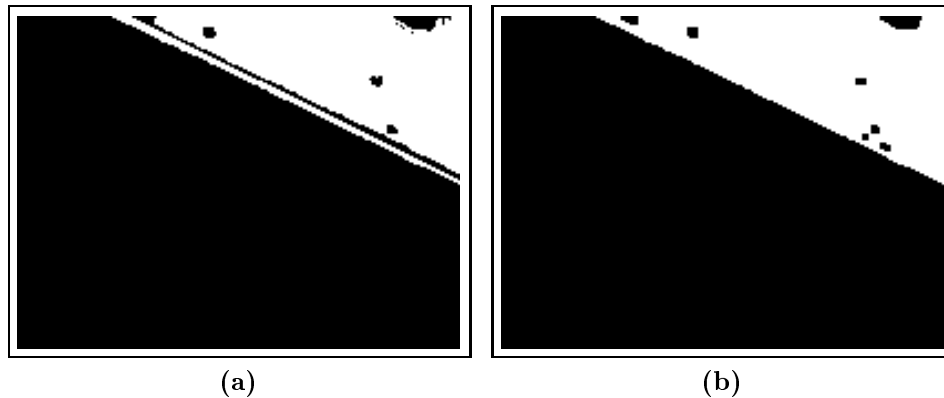
**Abbildung 4.2:** Extrahierte Einzelbilder aus zwei verschiedenen Klettervideos, (a) ohne Infrarotfilter und (b) mit Infrarotfilter.

### 4.2.1 Vorverarbeitung

Die Bildverarbeitungs-Operationen werden auf Einzelbildern durchgeführt. Eine Erklärung der vorbereitenden Schritte zum Benutzen des Programms zur Markerdetektion befindet sich in Anhang B.2.

Als erstes wird eine Binarisierung durchgeführt, um die hellen Marker vom Hintergrund zu trennen. Farbbilder werden hierfür intern in Graustufenbilder umgewandelt. Der Schwellwert wird über das kumulative Histogramm des Bildes bestimmt. Ein kumulatives Histogramm ist eine Abbildung, das die Häufigkeiten der einzelnen Grauwerte aus dem normalen Histogramm, beginnend mit der Häufigkeit von schwarz, bis zu einem gewählten Wert aufaddiert. Das heißt, das kumulative Histogramm  $M_i$  von einem Histogramm  $M_j$  ist definiert als

$$M_i = \sum_{j=1}^i m_j \quad (4.1)$$



**Abbildung 4.3:** Der obere rechte Bereich des Bildes aus Abbildung 4.2 nach der Binarisierung. In dem Wandstück wird der Effekt des Closings gut erkennbar. Zum Vergleich das Bild (a) ohne anschließendem Closing und (b) mit anschließendem Closing. Es ist zu erkennen, dass durch das Closing kleine Lücken geschlossen werden.

Zur Bestimmung des Schwellwertes wird im kumulierten Histogramm die 99%-ige Anzahl der Gesamtzahl an Pixeln betrachtet. Dieser Wert wurde empirisch für die Kletteraufnahmen ermittelt. Die auf diese Weise ausgewählten Pixel beschreiben die hellsten Grauwerte im Bild und somit den optimalen Schwellwert zum Trennen der Marker vom Rest des Bildes. Voraussetzung ist allerdings, dass sich die Marker stark im Bild hervorheben, da der Schwellwert sonst zu niedrig bestimmt wird. Auf das Binärbild wird Closing angewandt, um zusammenhängende Flächen, die kleine Löcher enthalten, zu homogenisieren. So ergeben sich bei der Konturfindung nicht unnötig viele Konturen (vgl. Abbildung 4.3).

### 4.2.2 Konturfindung

Um mehr Informationen aus dem Binärbild zu gewinnen, wird anschließend ein Verfahren zur Konturfindung nach [SA85] angewandt. Der Algorithmus analysiert die topologische Struktur eines Binärbildes. Er findet Konturen, indem er das Bild zeilenweise durchläuft und Farbwechsel sucht. Durch Verfolgung eines Farbübergangs wird die Kontur erfasst und durch Unterscheidung von Eltern- und Kind-Konturen bzw. äußeren und inneren Grenzen in eine Baumstruktur eingefügt.

In dieser Arbeit wird die Implementierung des Algorithmus in OpenCV verwendet. Dort wurde das ursprüngliche Verfahren unter anderem um eine Methode erweitert, bei der die gefundenen Konturen in einer Liste repräsentiert werden. Für die Darstellung der als Freeman-Code repräsentierten Konturen werden die Werte in Punkte übersetzt. Die Konturen beschreiben letztlich jeweils den inneren Umfang einer Komponente.

### 4.2.3 Aussortierung

Nach der Konturfindung liegen Konturen in verschiedener Form und Größe vor (vgl. Abbildung 4.4a). Das Ziel ist, die einen Marker beschreibenden Konturen von den übrigen Konturen zu trennen. Als Anhaltspunkt kann hierbei gelten, dass die Markerkonturen ähnlich groß sind und eine vergleichsweise runde Form haben. Die Rundheit einer Kontur kann durch die Formel für Kompaktheit berechnet werden [Ste08]:

$$c = 4\pi \frac{A}{U^2} \in [0, 1] \quad (4.2)$$

Der Wert  $c = 1$  beschreibt demnach einen Kreis und der Wert  $c = 0,785$  entspricht einem Quadrat. Die Kompaktheit einer Kontur wird um so kleiner, je größer ihr Umfang  $U$  im Verhältnis zum Flächeninhalt  $A$  ist. In dieser Arbeit wird angenommen, dass Konturen mit einer Kompaktheit kleiner 0,72 keine Marker beschreiben und daher verworfen werden können. Damit sind stark rechteckige Konturen, wie sie durch ein reflektierendes Wandstück hervorgerufen werden können, entfernt worden. Leicht ovale Konturen werden noch akzeptiert.

Die erhalten gebliebenen Konturen haben zueinander eine ähnliche Form und variieren nur noch in ihrer Größe. Eine sehr kleine runde Kontur kann hier beispielsweise durch eine reflektierende Schraube in der Kletterwand entstehen. Diese Ausreißer, die einen viel größeren oder kleineren Flächeninhalt haben, lassen sich gut mit Hilfe eines Medians über die Flächeninhalte der Konturen ausgrenzen. Toleriert werden ausschließlich Konturen, die dem Median zu 80% entsprechen, das heißt eine gewisse Abweichung vom Median ist zulässig.

Eine letzte Aussortierung erfolgt durch das Überprüfen der Farbe. Entspricht der Schwerpunkt der Kontur im Binärbild einem schwarzen Pixel, so kann die Kontur keinen Marker beschreiben und wird verworfen.

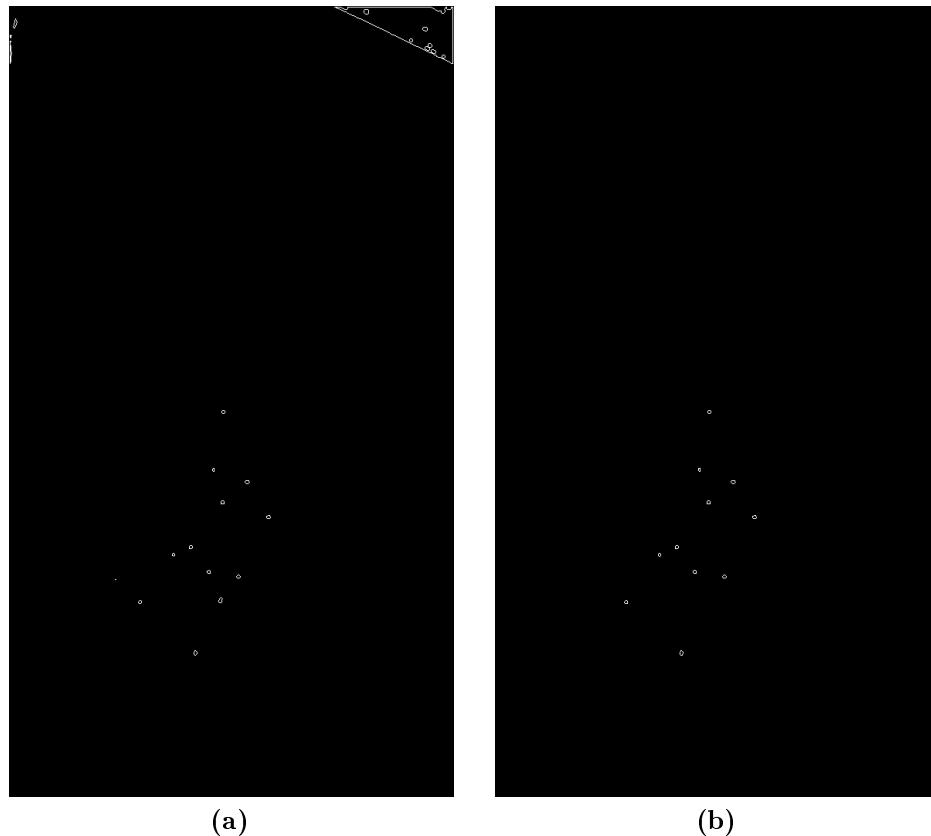
### 4.2.4 Berechnung des Schwerpunktes

Im letzten Schritt wird von den übrigen Konturen, optimalerweise den Markern, der geometrische Schwerpunkt berechnet, um die Koordinaten bestimmen zu können. Dies ist mit Momenten möglich. Das Moment  $m_{p,q}$  der Ordnung  $p + q$  über eine Funktion  $f(x, y)$  als die Dichte eines Objektes wird allgemein definiert als [Pap02]:

$$m_{p,q} = \iint x^p y^q f(x, y) dx dy \quad (4.3)$$

Aus dieser Gleichung berechnet sich das Moment nullter Ordnung als

$$m_{0,0} = \iint f(x, y) dx dy \quad (4.4)$$



**Abbildung 4.4:** (a) Konturen des binarisierten Bildes aus Abbildung 4.3b, (b) Ergebnis nach Verwerfung derjenigen Konturen, die die Kriterien (vgl. Abschnitt 4.2.3) nicht erfüllten.

und die Momente erster Ordnung als

$$\begin{aligned} m_{1,0} &= \iint x f(x, y) dx dy \\ m_{0,1} &= \iint y f(x, y) dx dy \end{aligned} \quad (4.5)$$

Das Moment nullter Ordnung beschreibt die Fläche des Objektes. Aus den Momenten erster Ordnung kann der Schwerpunkt  $(x_c, y_c)$  von  $f(x, y)$  wie folgt berechnet werden:

$$x_c = \frac{m_{1,0}}{m_{0,0}} \quad y_c = \frac{m_{0,1}}{m_{0,0}} \quad (4.6)$$

Die somit berechneten x und y-Koordinaten der Marker eines Bildes werden vom Programm in eine Textdatei geschrieben. Dies erfolgt für jeden Frame des Videos. So ist es möglich, die Marker für ihre Verfolgung durch Partikelfilter im Tracking-Programm zu visualisieren.



Für den Import nach Simi Motion müssen die Koordinaten nicht nur frameweise, sondern auch nach Gelenk getrennt werden, um eine sinnvolle Analyse zu ermöglichen. Bei dieser Zuordnung der Marker zu Gelenken setzt der Partikelfilter ein.

## 4.3 Tracking

In dieser Arbeit basiert das Tracking auf der Beobachtung der Positionen der Marker. Hierfür wurden die Markerpositionen im Programm zur Markerdetektion berechnet. Zur Verfolgung der detektierten Marker werden die Markerdaten vom Detektions-Programm an das Tracking-Programm übergeben. Eine detailliertere Erklärung hierzu befindet sich in Anhang B.3.

Das Tracking-System wurde auf Basis eines in [Wir07] implementierten Demo-Partikel-Filters weiterentwickelt, in dem ein durch den Nutzer steuerbares Auto mit einem Partikelfilter verfolgt werden konnte. Das Programm wurde in dieser Arbeit für mehrere Partikelfilter und mehrere zu verfolgende Objekte angepasst und außerdem um eine Funktion zum Export der aufgezeichneten Pfade erweitert.

Im Folgenden werden die verwendeten Parameter und die Implementierung der dynamischen Modelle des Condensation-Algorithmus für den Anwendungsfall des Trackings von Kletterern beschrieben.

### 4.3.1 Zustandsmodell

Generell werden Objekte zur Zeit  $t$  mit einem Zustandsvektor  $\mathbf{x}_t$  beschrieben, der verschiedene Informationen wie Position, Ausrichtung und Geschwindigkeit in x- und y-Richtung des Objekts im Bild kodiert. In dieser Arbeit wird nur die Position als Parameter verwendet. Der Zustandsvektor hat daher die folgende Form:

$$\mathbf{x}_t = \begin{pmatrix} x \\ y \end{pmatrix} \quad (4.7)$$

### 4.3.2 Bewegungsmodell

Aufgrund der geringen Frame-Rate des Videos ergeben sich beispielsweise durch dynamisches Weitergreifen zum nächsten Griff Bewegungsunschärfen, deren Konturen bei der Markerdetektion nicht zuverlässig erkannt wurden. Bei der Darstellung der Marker im Tracking-Programm kommt es dann zu einem Flackern bzw. kleinen Sprüngen. Die niedrige Frame-Rate und die dadurch zu schnellen Bewegungen machen es daher unmöglich, die Geschwindigkeiten der Marker realistisch zu schätzen. Aus diesem Grund wird auf die Implementierung der deterministischen Komponente des Bewegungsmodells verzichtet. Um die Bewegungen dennoch vorherzusagen

zu können, werden die Partikel stark gaussverteilt gestreut. Die Erkennungswahrscheinlichkeit für Marker wird dabei um so geringer, je größer der Sprung ist, den sie gegenüber dem Ausgangszustand gemacht haben.

### 4.3.3 Beobachtungsmodell

Im Beobachtungsmodell werden die Ergebnisse des Programmes zur Markerdetektion ausgewertet. Generell wird in diesem Modell gemessen, wie gut ein Partikel dem tatsächlichen Objektzustand entspricht. In dieser Arbeit erfolgt dies über eine einfache Abstandsberechnung: Für jedes Partikel wird der Abstand zu allen Markerzuständen gemessen und der geringste Abstand gespeichert. Der Partikel, der unter allen Partikeln den geringsten Abstand zu einem Marker hat bzw. den zutreffendsten Zustand, repräsentiert den Markerzustand am Besten und erhält die höchste Gewichtung. In Abhängigkeit vom Ergebnis dieser Messung werden die Partikel neu gewichtet.

### 4.3.4 Tracking-Funktionalität

Beim Starten des Programms wird für jeden zu verfolgenden Marker per Mausklick ein Partikelfilter angelegt und beim Trackingvorgang die Positionen der Partikelfilter über die Zeit gespeichert (vgl. Abbildung 4.5). Bei gewünschtem Export für Simi Motion werden sie in eine Textdatei geschrieben. Nähere Hinweise zur Bedienung des Programms werden in Anhang B.4 gegeben.

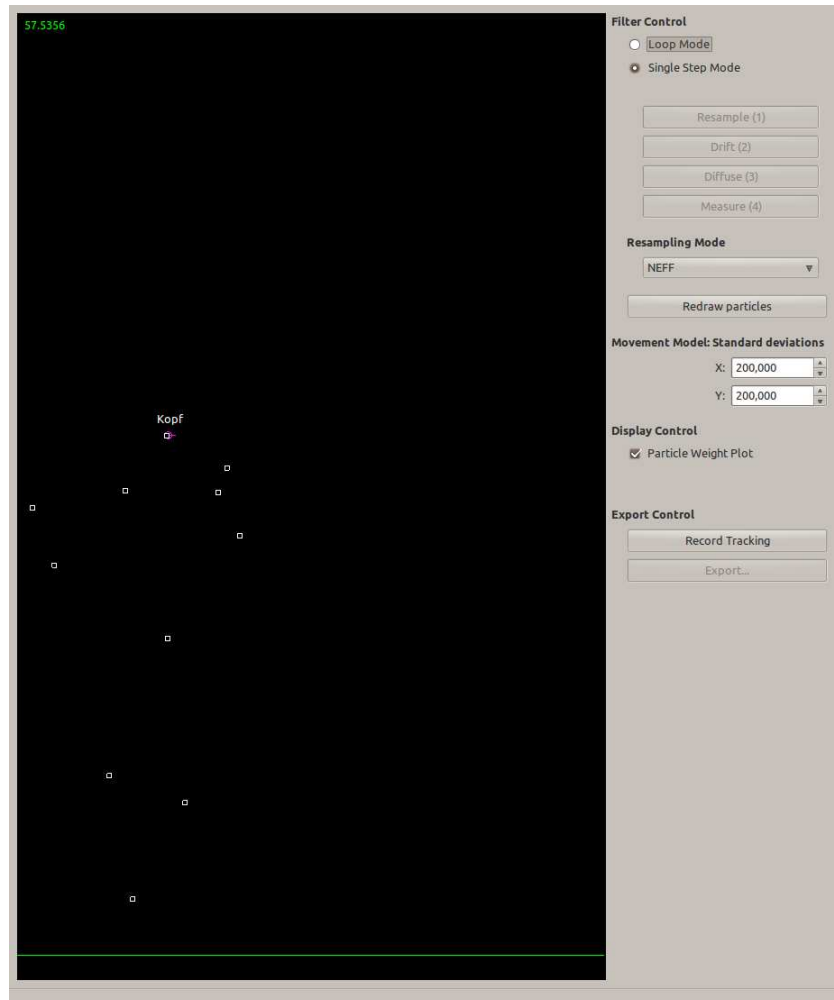
Ein Partikelfilter arbeitet korrekt, wenn er den ihm zugewiesenen Marker (vgl. Abbildung 4.5) die ganze Zeit über verfolgt und nicht verliert. Im entwickelten Programm zur Markerverfolgung ist dies primär abhängig von der Anzahl der verwendeten Partikel sowie von der im Bewegungsmodell verwendeten Streuung und wird erleichtert, wenn sich keine weiteren Marker in der Nähe befinden. Ist dies jedoch der Fall, besteht die Möglichkeit, dass durch die Abstandsmessung im Beobachtungsmodell eine Gewichtung der Partikel zu Gunsten eines Markers in der Nähe stattfindet und der Partikelfilter auf den anderen Marker springt und weiter verfolgt. Dies ist ein bekanntes Problem bei der Arbeit mit Partikelfiltern, ebenso der Fall wenn mehrere Partikel gemeinsam einen Marker belegen und verfolgen. Hierzu kann es beispielsweise kommen, wenn ein zu verfolgender Marker temporär verdeckt wird und ein anderer Marker in der Nähe ist. Das Problem des übereinander springens wurde in der Implementation dadurch gelöst, dass ein Marker von einem Partikelfilter in Besitz genommen werden kann, und in dem Fall für andere Partikelfilter als vergeben deklariert wird. In Abbildung 4.6 und 4.7 werden zwei verschiedene Verhaltensweisen der Partikelfilter dargestellt.

## 4.4 Export für Simi Motion

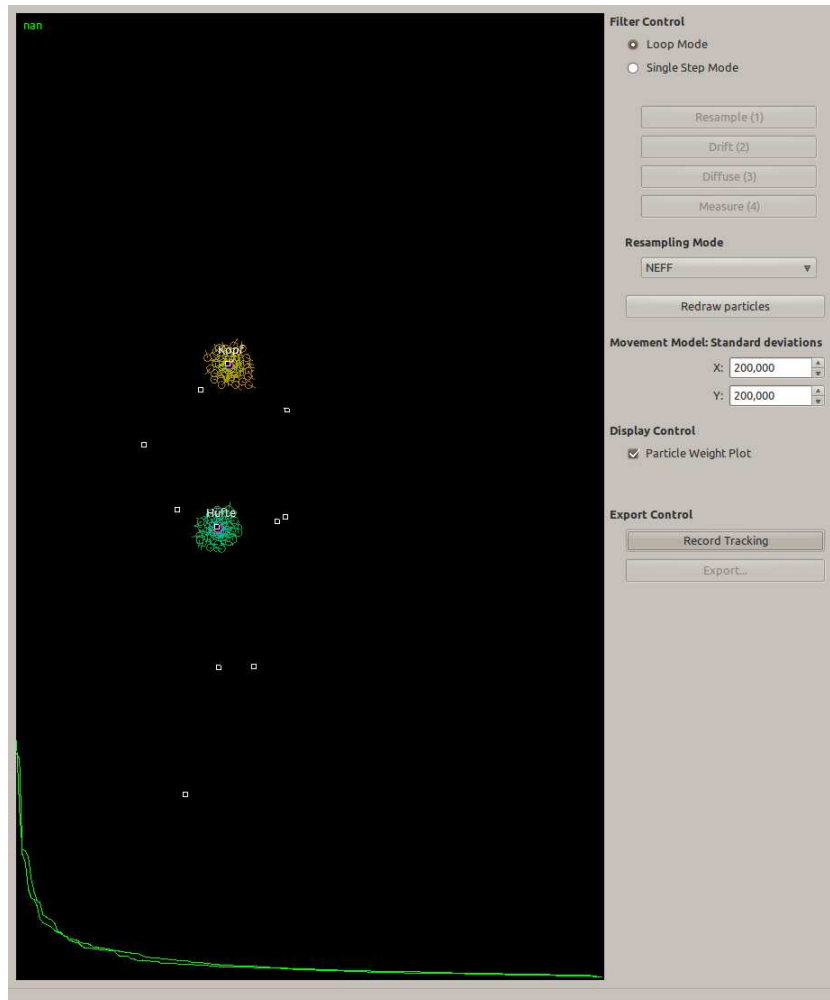
Simi Motion bietet die Möglichkeit, Textdateien mit Daten zu importieren und in die Analyse mit einzubeziehen. Sämtliche Daten können in Diagrammen dargestellt und so untereinander verglichen werden.

Für eine vergleichbare Darstellung der Markerdaten wurde die zu schreibende Exportdatei nach Markern getrennt und über die Zeit die X- und Y-Koordinaten der Marker aufgelistet. In den Einstellungen des Programms beim Importieren von Textdateien kann dann angegeben werden, wie die Daten in der Textdatei interpretiert werden sollen (vgl. Abbildung 4.8).

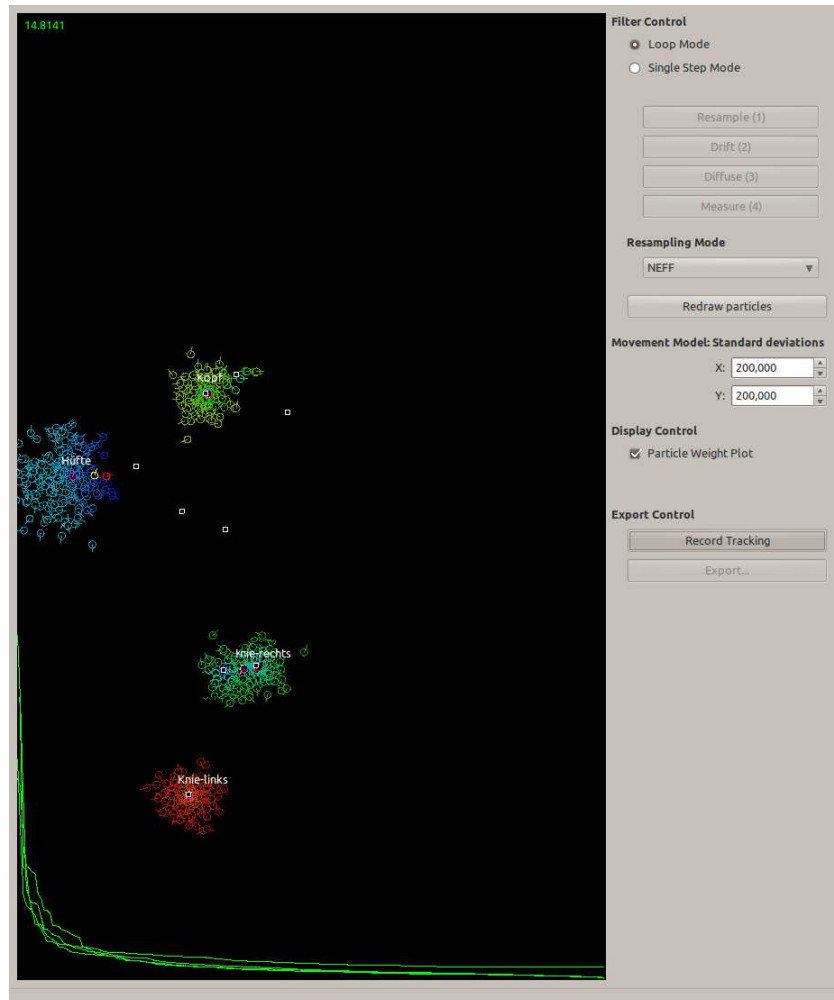
In gleichem Format wie beim Import können Koordinaten von Markern, die mit den Trackingfunktionen in Simi Motion gewonnen wurden, auch exportiert werden. Diese Möglichkeit wird im nächsten Kapitel genutzt, um die Ergebnisdaten des Markerverfolgung mit Partikelfiltern mit den Daten des Trackings in Simi Motion zu vergleichen.



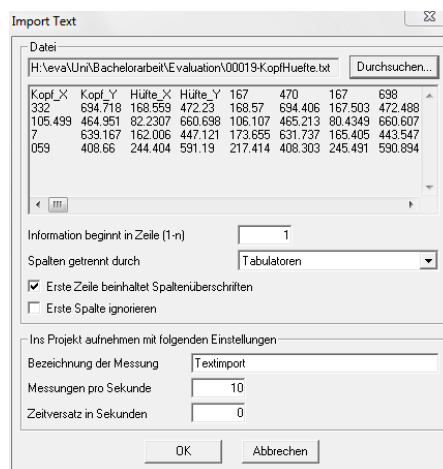
**Abbildung 4.5:** Screenshot des Tracking-Programms. Partikelfilter können mit der Maus gesetzt werden und sind anschließend mit Namen in der Anwendungsoberfläche sichtbar. Das Tracking wird im *Loop Mode* durchgeführt, bzw. beim klicken von *Record Tracking*, wenn die Wege der Partikelfilter für Simi Motion aufgenommen werden sollen.



**Abbildung 4.6:** In dieser Situation wurden zwei Partikelfilter im Tracking-Programm gesetzt, um jeweils den Kopf und die Hüfte zu verfolgen. Die Filter arbeiten korrekt und verfolgen die ihnen zugewiesenen Marker.



**Abbildung 4.7:** In diesem Fall wurden vier Partikelfilter gesetzt, um jeweils die Marker des Kopfes, der mittleren Hüfte und der Knie verfolgen. Der Partikelfilter für die Hüfte hat hierbei seinen ursprünglichen Marker verloren und ist zu einem anderen Marker gewechselt.



**Abbildung 4.8:** Die Optionen zum Import einer Textdatei. Hier kann eingestellt werden, wie Spalten voneinander getrennt sind und welche Zeilen der Textdatei relevante Informationen enthalten. Das Feld *Messungen pro Sekunde* beschreibt die Auflösung der Daten und entspricht der Frame-Rate. Bei den im Tracking-Programm ausgewerteten Kletteraufnahmen werden 25 Messungen pro Sekunde vorgenommen.





# Kapitel 5

## Experimente und Evaluation

Dieses Kapitel beschreibt die Evaluation der Markerdetektion und -verfolgung und vergleicht die Ergebnisse der Analyse in Simi Motion mit den erhaltenen Daten der Verfolgung durch Partikelfilter.

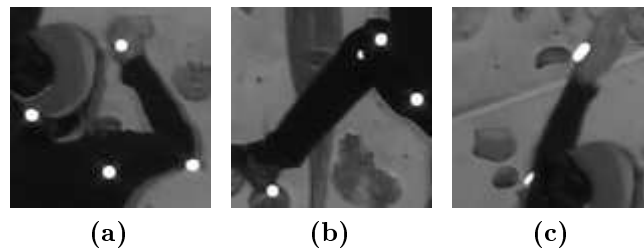
### 5.1 Evaluation der Markerdetektion

Zunächst soll die Zuverlässigkeit der Markererkennung der entwickelten Detektions-Software bewertet werden. Parameter zur Beeinflussung des Detektionsergebnisses sind:

- Bestimmung des Schwellwertes für Binarisierung
- Toleranz der Kompaktheit
- Abweichung vom Median über die Flächeninhalte der Konturen

Weiterhin spielt die Auflösung und Qualität der zu verarbeitenden Bilder eine Rolle. In dieser Arbeit wurden daher sämtliche Bilder auf eine Größe von 1024x576 Pixel skaliert.

Für die Evaluation der Markererkennung wurde das Programm zur Markerdetektion auf verschiedenen in Einzelbilder zerlegten Klettervideos ausgeführt. Gefundene Marker wurden durch rote Kreise gekennzeichnet und in das jeweilige Bild gezeichnet. Wie zuvor bereits erwähnt, ergeben sich beim Klettern zeitweise Verdeckungen von Gelenken. Dies resultiert neben verdeckten auch in teilverdeckten Markern. Durch die Frame-Rate der hier verwendeten Kamera ergeben sich zudem Bewegungsunschärfen, wenn ein dynamischer Zug vollzogen wird. In Abbildung 5.1 werden die verschiedenen Erscheinungsformen der Marker zusammenfassend dargestellt.



**Abbildung 5.1:** Unterschiedliche Erscheinungsformen eines Markers innerhalb eines Frames (a) Marker, die als vollständig sichtbar bewertet werden können, (b) Marker am linken Knie wird als teilverdeckt bezeichnet, (c) Bewegungsunschärfen, die sich durch eine geringe Frame-Rate der Kamera ergeben. Solche Marker werden ebenfalls der Klasse *teilverdeckt* zugeordnet.

Parameter	Wert
Schwellwert Ratio	99%
Kompaktheit	0,72
Median Abweichung	80%

**Tabelle 5.1:** Die zum Zeitpunkt der Evaluation verwendeten Parameterwerte

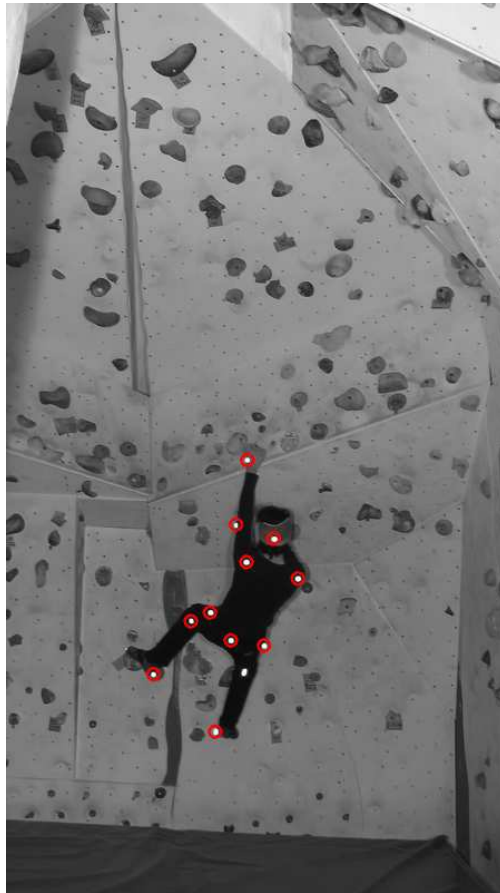
Im Anschluss wurden basierend auf diesen Beobachtungen 5 Klassen erstellt und die Detektionsergebnisse eines Bildes manuell jeweils einer der Klassen zugeordnet. Es wird unterschieden zwischen:

- erkannter, vollständig sichtbarer Marker
- erkannter, teilverdeckter oder verzerrter Marker
- nicht erkannter, vollständig sichtbarer Marker
- nicht erkannter, teilverdeckter oder verzerrter Marker
- falsch erkannter Marker

Durch frühere Auswertung einzelner Frames der aufgenommenen Videosequenzen, haben sich Parameter gefestigt, die für die meisten Situationen gut funktionieren. Die Parameterwerte, mit denen alle nachfolgenden Detektions-Tests durchgeführt wurden, sind in Tabelle 5.1 dargestellt.

### 5.1.1 Erster Test

Im ersten Evaluations-Test wurde das Video ausgewertet, aus dem in den früheren Kapiteln bereits Einzelbilder gezeigt wurden. Es wurden insgesamt 651 Frames



**Abbildung 5.2:** Das Ergebnis der Markerdetektion (vgl. Abbildung 4.2a). Der Marker des rechten Knies wurde nicht erkannt, da er verzerrt aufgenommen wurde.

untersucht, die alle Klettervorgänge erfassen - vom ersten Tritt bis zum Vorbereiten des Absprungs. Zur besseren Vorstellung zeigt Abbildung 5.2 das Ergebnis einer Detektion.

Bei Betrachtung der Tabelle 5.2 sowie des Diagramms in Abbildung 5.3 fällt auf, dass die Erfolgsrate bei den vollständig sichtbaren Markern wesentlich höher ist als bei teilverdeckten bzw. verzerrten Markern. Hierzu muss gesagt werden, dass das Programm zur Markerdetektion nicht darauf ausgelegt wurde, teilverdeckte Marker zu erkennen. Mit der Veränderung entsprechender Parameter, beispielsweise der Kompaktheit, wird die Akzeptanz größer, es steigt jedoch auch die Falsch-Positiv-Rate. Dies soll näher in der zweiten Evaluation gezeigt werden.

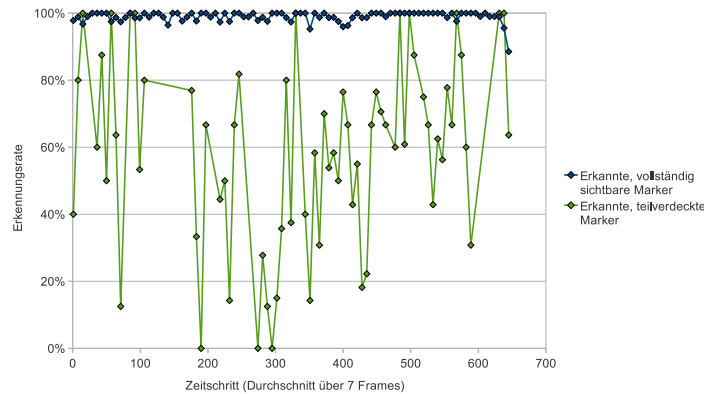
Summe Detektionen	8264
davon tatsächlich Marker	8261
<b>Wahrscheinlichkeit einen Marker zu finden</b>	<b>99,96%</b>
Summe ausgewertete Marker	8261
davon korrekt ausgewertete Marker	7880
<b>Erkennungswahrscheinlichkeit allgemein</b>	<b>95,4%</b>
Summe ausgewertete Marker (vollständig sichtbar)	7558
davon korrekt ausgewertete Marker	7479
<b>Erkennungswahrscheinlichkeit vollständig sichtbare Marker</b>	<b>99%</b>
Summe ausgewertete Marker (teilverdeckt)	703
davon korrekt ausgewertete Marker	401
<b>Erkennungswahrscheinlichkeit teilverdeckte Marker</b>	<b>57%</b>

Tabelle 5.2: Statistiken des ersten Tests

### 5.1.2 Zweiter Test

Im zweiten Test wurde ein Video ausgewertet, das veränderte Aufnahmebedingungen vorweist. Der Kletterer bewegt sich zum einen von rechts nach links an der Wand entlang, zum anderen konnte aus Platzgründen nur ein statt zwei Scheinwerfer zur Beleuchtung verwendet werden. Gleichzeitig waren Kamera und Scheinwerfer wesentlich näher an der Wand und am Kletterer, als beim Video aus dem ersten Test, wodurch die Aufnahme insgesamt heller und an manchen Stellen überbelichtet wurde. Es wurden 455 Frames des Videos ausgewertet. Ein beispielhaftes Detektionsergebnis eines Bildes ist in Abbildung 5.4 dargestellt.

Betrachtet man nun die Erkennungsraten von Markern im zweiten Video in Tabelle 5.3 und das Diagramm in Abbildung 5.5, so fällt auf, dass die Wahrscheinlichkeit der Erkennung vollständig sichtbarer Marker im Vergleich zum ersten Test wesentlich gesunken ist. Die Falsch-Positiv-Rate ist gestiegen, da in vielen hell reflektierenden Griffen fälschlicherweise Marker erkannt wurden (vgl. Abbildung 5.4). Da der jeweils berechnete Schwellwert auf den Bildern dieses Videos von Grauwert 118 auf 225 steigt, lässt sich hieraus vermuten, dass ein Schwellwert-Ratio größer als 99% möglicherweise bessere Ergebnisse liefert. Bei der Auswertung ist weiterhin aufgefallen, dass der Marker an der rechten Hand niemals erkannt wurde. Grund dafür ist, dass er mit weißem Kreppband fixiert war, und dieses einen ähnlichen Reflektionsgrad wie der Marker hatte. Daher waren im Vorverarbeitungsschritt die Konturen des Markers und des Tapes stets miteinander verbunden.



**Abbildung 5.3:** Die Erkennungsrate von Markern in Bildern des Videos aus Test 1. Es wurden 651 Frames ausgewertet und für das Diagramm der Durchschnittswert über 7 Frames berechnet. Unterschiedlich große Abstände zwischen den Datenpunkten ergeben sich, weil dort keine Messdaten verfügbar waren und der Markertyp in diesen Sequenzen nicht aufgetreten ist.

## 5.2 Tracking

### 5.2.1 Vergleich der Trackingergebnisse

Um die Daten beider Programme vergleichbar darstellen zu können, wurde eine Kalibrierung in Simi Motion vorgenommen. Der Grund ist, dass sich die Ursprungskordinate im Tracking-Programm oben links im Bild befindet und in Simi Motion unten links. Um Messdaten den Raumverhältnissen zuordnen zu können, muss bei der Kalibrierung in Simi Motion der Maßstab eines Meters im Bild markiert werden. Da es sich bei der Messeinheit des Tracking-Programms aber um Pixel handelt, wurde das Bild auf 1024 x 576 Meter kalibriert, was der verwendeten Bildauflösung entspricht.

Die Erkennung der Markerkoordinaten wurde in Simi Motion mit der Funktion zum *automatischen Erfassen* der Marker durchgeführt. Da die Erfassung bei kurzzeitiger Verdeckung von Markern jedoch fehlerhaft fortgeführt wurde, erfolgte die weitere Erfassung mit schrittweiser Prüfung jedes einzelnen Bildes, wobei die falsch gesetzten Marker korrigiert wurden.

Die Diagramme in Abbildung 5.6 stellen jeweils die X- und Y-Koordinaten der Marker dar, die aus der Markererfassung mit Simi Motion resultieren. Die Daten beziehen sich auf eine bisher nicht vorgestellte Aufnahme, die der in Abschnitt 5.1.1 evaluierten Aufnahme jedoch ähnlich ist. Bei näherer Betrachtung der Grafiken fällt auf, dass sich zwischen den Datenreihen vereinzelt Lücken befinden. Diese sind durch Verdeckung oder Verzerrung der Marker entstanden, wodurch die Da-



**Abbildung 5.4:** Das Ergebnis der Markerdetektion auf einem Bild des Videos aus Test 2. Durch die im Vergleich zum ersten Video heller beleuchtete Szene reflektieren vereinzelt Teile von Griffen und werden fälschlicherweise als Marker erkannt.

tenerfassung ausgesetzt wurde. Konstant sichtbar sind hingegen die Marker an Kopf, Schultern und Hüfte.

Zum Vergleich einzelner Datenreihen sind in Abbildung 5.7 die Markerdaten vom Kopf und mittlerer Hüfte aus Simi Motion zusammen mit den als Textdatei importierten Markerdaten des Tracking-Programms dargestellt. Dicke Linien entsprechen den Daten aus Simi Motion, dünne Linien den erfassten Daten aus dem Tracking-Programm. Das Partikelfilter-Tracking wurde in Abbildung 5.7a mit 150 Partikeln durchgeführt und in Abbildung 5.7b mit 200 Partikeln. Es sind auf den ersten Blick kaum Unterschiede festzustellen, da sowohl in Simi Motion als auch im Tracking-Programm ähnliche Positionen der Marker bestimmt wurden. Bei näherer Betrachtung von Abbildung 5.7b fallen jedoch kleine Fehler auf. Dort werden im Bereich des Schnittpunktes (12, 500) die Partikelfilter im übertragenen Sinne von ihrer korrekten Bahn abgelenkt, wenn sie in die Nähe eines anderen Markers kommen. Ähnlich ist der Fall im Schnittpunkt (16, 500). Weiterhin ist in Abbildung 5.7b zu bemerken, dass der Partikelfilter, der den Marker des Kopfes verfolgen soll, nach 19 Sekunden resultierend aus der Verdeckung seines Markers von einem anderen Marker abgelenkt wird. In diesem Fall war der zu schätzende Weg für den Partikelfilter zu lang.

Eine zuverlässige Verfolgung der Marker ist nicht mehr garantiert, wenn Spuren von häufig verdeckten Markern verglichen werden (vgl. Abbildung 5.8). Das Markertracking durch Partikelfilter wurde hier mit 150 Partikeln durchgeführt. Es ist zu erkennen, dass die Partikelfilter sehr unsicher agieren und beim Zusammentreffen mehrerer Marker einige Partikelfilter ihren zu verfolgenden Marker verlieren.

Summe Detektionen	7133
davon tatsächlich Marker	5348
<b>Wahrscheinlichkeit einen Marker zu finden</b>	<b>74,98%</b>
Summe ausgewertete Marker	5348
davon korrekt ausgewertete Marker	4197
<b>Erkennungswahrscheinlichkeit allgemein</b>	<b>78,5%</b>
Summe ausgewertete Marker (vollständig sichtbar)	4267
davon korrekt ausgewertete Marker	3545
<b>Erkennungswahrscheinlichkeit vollständig sichtbare Marker</b>	<b>83,1%</b>
Summe ausgewertete Marker (teilverdeckt)	1081
davon korrekt ausgewertete Marker	652
<b>Erkennungswahrscheinlichkeit teilverdeckte Marker</b>	<b>60,3%</b>

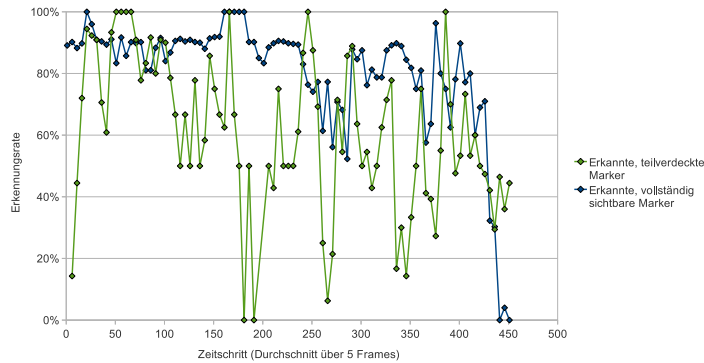
Tabelle 5.3: Statistiken des zweiten Tests

Es ist jedoch zu berücksichtigen, dass manche Marker über einen sehr langen Zeitraum verdeckt werden, der schwer abschätzbar ist.

In einer letzten Untersuchung soll das Tracking auf der Aufnahme einer Hochgeschwindigkeitskamera durchgeführt werden, die im Rahmen von [Han08] entstand. Darin wird ein mit 3 Markern versehener Unterkörper bei einem Tritt des rechten Beins von der Seite aufgenommen, wobei der Marker des Sprunggelenks am stärksten beschleunigt wird. Durch die hohe Frame-Rate der Kamera und einer daraus resultierenden großen Datenmenge, ergeben sich weiche Bewegungen, die von 3 Partikelfiltern à 200 Partikeln gut erfasst werden. Zu Testzwecken wurde der Marker am Knie in 20 Frames manuell entfernt, um eine Verdeckung zu simulieren. Das Ergebnis ist in Abbildung 5.9 dargestellt. Hier ist zu erkennen, dass die Partikelfilter in Momenten verdeckter Marker kurz von ihrer vorgesehenen Linie abweichen, um den Marker bei erneutem Auftauchen weiter zu verfolgen.

### 5.2.2 Bewertung

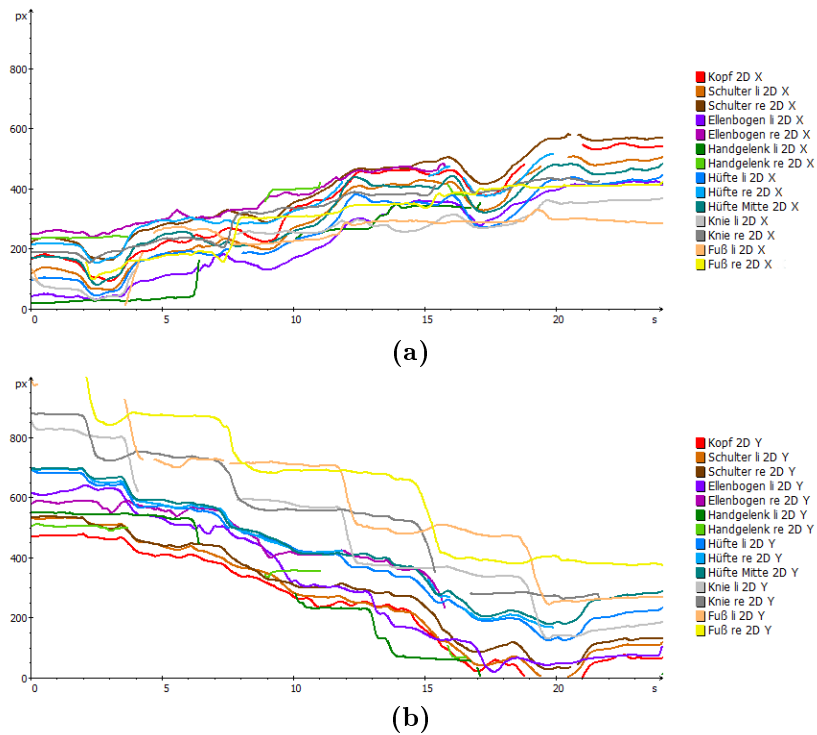
Die im letzten Abschnitt durchgeführten Untersuchungen haben gezeigt, dass die Ergebnisse der Markerverfolgung mit Partikelfiltern von der jeweiligen Partikelanzahl abhängen sowie von der Dauer, über die ein temporär nicht sichtbarer Marker zu verfolgen ist. Filter mit einer größeren Zahl an Partikeln neigen durch die Mehrzahl an aufgestellten Hypothesen zu Ablenkungen und Sprüngen, aber können ihren Marker, sofern sich in unmittelbarer Nähe keine weiteren Marker befinden, sicher verfolgen (vgl. Abbildung 5.7). Partikelfilter mit weniger Partikeln laufen Gefahr ihren zu trackenden Marker zu verlieren, aber neigen dafür nicht so sehr zu Ablenkungen durch andere Marker.



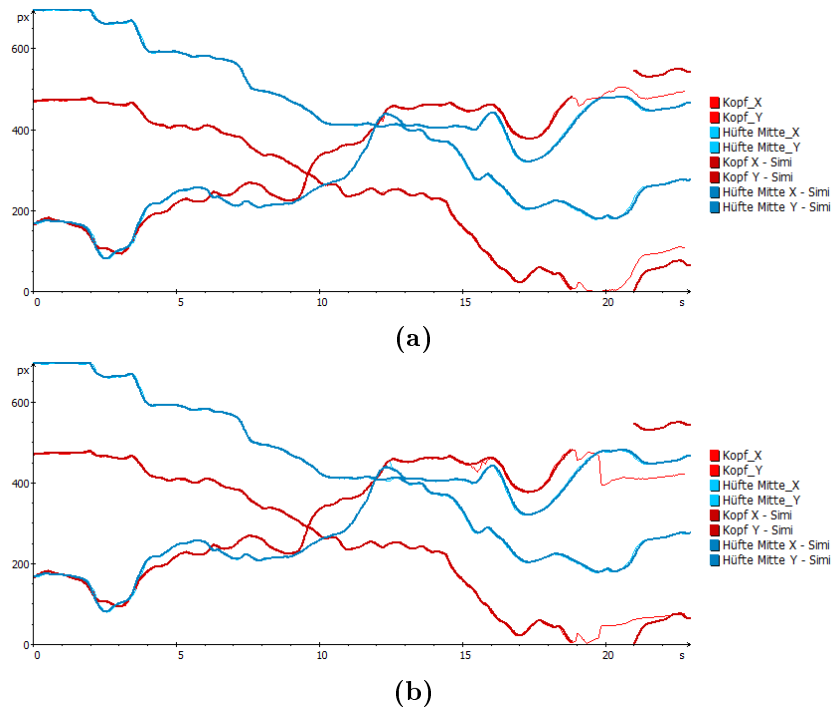
**Abbildung 5.5:** Die Erkennungsrate von Markern in Bildern des Videos aus Test 2. Es wurden 455 Frames ausgewertet und für das Diagramm der Durchschnittswert über 5 Frames berechnet.

Wird ein Marker über einen kurzen Zeitraum verdeckt, kann der Partikelfilter ihn in der Regel wiederfinden (vgl. Abbildung 5.9). Ist der Zeitraum der Verdeckung jedoch größer, gelingt dies nicht mehr, da der Partikelfilter die Bewegungsrichtung des Markers nicht berücksichtigt, sondern nur eine Streuung der Markerposition stattfindet (vgl. Abbildung 5.8). In dem Fall wechseln die Filter auf sichtbare Marker in unmittelbarer Nähe oder “schwärmen aus” wie in Abbildung 4.7 angedeutet wird. Wenn zu trackende Marker nah beieinander liegen, kann weiterhin die Streuung verringert werden, wodurch jedoch schnelle Bewegungen nicht sicher erfasst werden.

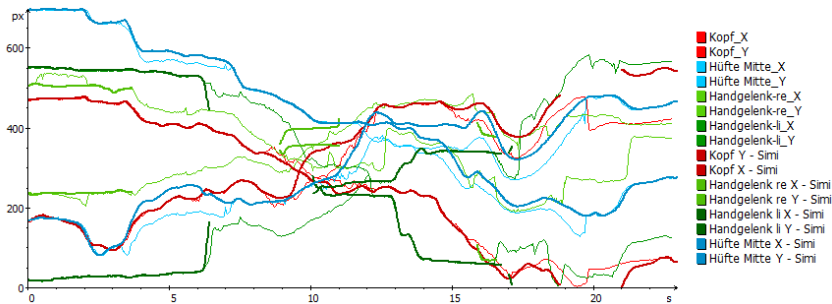




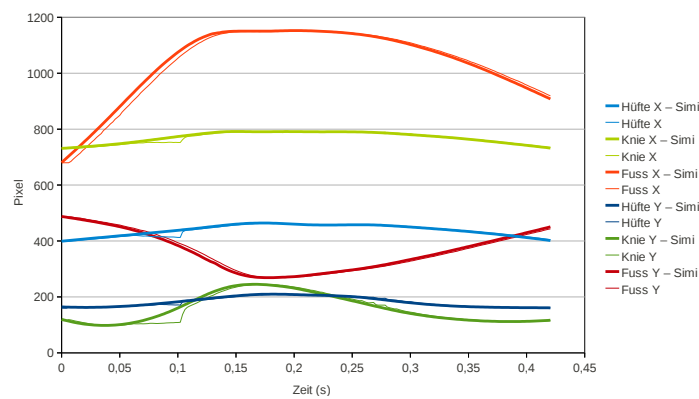
**Abbildung 5.6:** Diese Diagramme stellen die Koordinaten aller in Simi Motion detektierten Marker über die Zeit dar. Hier ist zu beachten, dass sich aufgrund der Kalibrierung der Koordinatenursprung oben links an der Decke der Kletterwand befindet. Die Lücken in den Datenreihen sind durch Verdeckung bzw. Verzerrung der Marker entstanden.



**Abbildung 5.7:** Diese Diagramme stellen die Markerpositionen einiger in Simi Motion detektierten Marker zusammen mit den Tracking-Ergebnissen über die Zeit dar. Dicke Linien beschreiben Simi-Daten, dünne Linien beschreiben Daten aus dem Tracking-Programm. In (a) wurde das Tracking mit 150 Partikeln durchgeführt und in (b) mit 200 Partikeln. Die großen Lücken sind durch Verdeckung der Marker entstanden.



**Abbildung 5.8:** Gemeinsame Darstellung mehrerer erfasster Markerdaten aus Simi-Motion mit Ergebnissen aus dem Tracking-Programm. Dicke Linien beschreiben Simi-Daten, dünne Linien beschreiben Tracking-Ergebnisse. Die Partikelfilter agieren aufgrund langer Verdeckungen von Markern sehr unsicher und kommen häufig von ihrem vorgesehenem Weg ab.



**Abbildung 5.9:** Gemeinsame Darstellung erfasster Markerdaten aus Simi-Motion mit Ergebnissen aus dem Tracking-Programm, basierend auf einer Hochgeschwindigkeitsaufnahme. In der Aufnahme wird ein Tritt nach vorne mit dem rechten Bein gezeigt, wobei sich die Marker an Hüfte, Knie und Fuss befinden. Vor dem Tracking wurde der Marker am Knie in 20 Frames manuell entfernt, um eine Verdeckung zu simulieren. Die Partikelfilter weichen in Momenten kurzer Verdeckungen kurz von ihren vorgesehenem Weg ab.



# Kapitel 6

## Zusammenfassung

In dieser Arbeit wurde ein Verfahren zur markerbasierten Erkennung und Verfolgung von Bewegungen unter Verwendung von Partikelfiltern vorgestellt. Am Beispiel der Bewegungen eines Kletterers konnte gezeigt werden, dass mit diesem Verfahren unter Nutzung einer handelsüblichen HD-Kamera und reflektierenden Markern gute Detektions- und Trackingergebnisse erzielbar sind. Der Export der Trackingdaten in das Bewegungsanalyse-Programm Simi Motion konnte ebenfalls demonstriert werden.

Bei der Analyse der Bewegungsabläufe kommt der zuverlässigen Markerdetektion eine besondere Bedeutung zu, da teilverdeckte Marker einer deutlich geringeren Erkennungswahrscheinlichkeit unterliegen als vollständig sichtbare Marker. Zur Demonstration der Markerdetektion wurden mehrere Versuchsreihen durchgeführt. Dabei konnte weiterhin gezeigt werden, wie Lichtverhältnisse und Kamerapositionierung die Markerdetektion beeinflussen.

Weiterhin wurden die Ergebnisse des Trackings mit Partikelfiltern mit den Ergebnissen aus der Markerdetektion in Simi Motion verglichen. Hierbei hat sich gezeigt, dass die Partikelfilter in der Lage sind, die Positionsänderungen kurzzeitig verdeckter Marker zu schätzen und beim erneuten Auftauchen wieder zu finden.

### 6.1 Grenzen der Verfahren

Das Markererkennungs- und Trackingverfahren liefert also auch mit vergleichsweise geringem technischen Aufwand gute Ergebnisse. Dennoch wurden einige Grenzen der Verfahren erkennbar:

1. **Markererkennung:** Die Erkennung basiert im wesentlichen auf der Unterscheidung der Helligkeitswerte und der Konturinformation der Marker. Beide Informationen sind anfällig für Verdeckung bzw. Teilverdeckung, Beleuchtungsprobleme sowie Herstellung und Befestigung der Marker am Körper.

Daher ist es bei den Untersuchungen wiederholt zu Fehlern bei der Erkennung gekommen.

2. **Vertauschung und Verlust von Markern:** Verfahrensbedingt werden bei der Zuordnung der erkannten Marker in jedem Frame die Informationen des vorherigen Frames genutzt. Auf diese Weise pflanzen sich Fehler bei der Detektion eines Markers in den folgenden Frames fort. Wenn ein Marker in einem Frame nicht erkannt wird, kann es vorkommen, dass dieser Marker in den folgenden Frames nicht exakt zugeordnet werden kann bzw. mit einem anderen in der Nähe detektierten Marker vertauscht wird.
3. **Zweidimensionale Erfassung:** Da nur eine Kamera bei den Kletteraufnahmen eingesetzt wurde, kann auch nur eine zweidimensionale Erfassung und Visualisierung der Daten erfolgen und die Gefahr der Verdeckung von Markern ist hoch.

Um diese Probleme zu lösen, werden im nächsten Abschnitt verschiedene Vorschläge gemacht.

## 6.2 Ausblick

Es bestehen verschiedene Möglichkeiten zur Fortführung der Arbeit, die im Folgenden vorgestellt werden sollen.

**Implementierung eines Bewegungsmodells.** Durch die Implementierung eines Bewegungsmodells wäre es möglich, exaktere Hypothesen beim Tracking mit Partikelfiltern anzustellen. Durch die Berechnung der Geschwindigkeit mit der sich ein zu verfolgender Marker bewegt, könnte zusammen mit der Bewegungsrichtung die Position eines Markers vorhergesagt werden.

**Mehrere Kameras zur 3D-Bewegungserfassung.** Durch die Erfassung der Bewegungen mit mehreren synchron geschalteten Kameras aus verschiedenen Blickrichtungen könnten Verdeckungen von Markern vermieden werden.

**Aufnahmen mit Hochgeschwindigkeitskameras.** Eine erneute Aufnahme mit mehreren Hochgeschwindigkeitskameras wäre sinnvoll, um exaktere Daten zu erhalten und Beschleunigungen genau berechnen zu können. Ein weiterer Vorteil wäre, dass so die Erfassung bewegungsunscharfer Marker vermieden werden kann, was zu einer besseren Markerdetektion führt.

**Markerloses Tracking.** Die temporäre Verdeckung von Markern erschwert ein zuverlässiges Tracking. Weiterhin zwingt sich der mit Markern versehene Proband unbewusst zur Vorsicht im Umgang mit den Markern, damit diese nicht abfallen oder kaputt gehen, wodurch ein natürliches Bewegungsverhalten verhindert wird. Durch ein markerloses Tracking unter Verwendung eines Menschmodells können möglicherweise bessere Tracking-Ergebnisse erzielt werden [WHT02]. Zugleich würde jederzeit ein Computermodell des Probanden zur Verfügung stehen, an dem Analysen vorgenommen werden könnten.

**Kraftsensoren.** In ferner Zukunft könnte die Bewegungserfassung an einer Kletterwand mit installierten Kraftsensoren an Griffen und Tritten durchgeführt werden. Die Messung der Kraftverteilung könnte dazu benutzt werden, den Körperschwerpunkt des Kletterers zu berechnen und zu visualisieren. Gleichzeitig könnte eine vorgeschlagene Griff- und Tritt-Kombination anzeigen, wie die Route mit minimalem Kraftaufwand geklettert werden kann.





# Anhang A

## Herstellung von Reflektionsmarkern

Hochwertige Kugelmarder sind aus folgendem Material günstig selbst herstellbar:

- Wattekugeln mit 2,5cm Durchmesser
- selbstklebende Filz pads in Größe der Wattekugeln
- hochreflektierende, selbstklebende Folie von *3MScotchLite*

Die Wattekugeln werden im Orangenschalen-Prinzip mit schmalen Streifen der Folie beklebt. Ein kleiner Bereich wird ausgelassen, um dort ein Loch zu bohren (falls nicht bereits vorhanden) und mit einer Schraube das Filz pad mit der Wattekugel zu fixieren. Die Schutzfolie auf den Filz pads wird für die Aufnahmen entfernt.



**Abbildung A.1:** Verschiedene Marker mit unterschiedlicher Herstellungstechnik. Die zwei Marker mit der helleren Folie (oberen Bildhälfte) konnten keine so guten Reflektionseigenschaften vorweisen und wurden bei den Aufnahmen nicht verwendet. Der im Bild mittig liegende Marker mit der glatten Struktur wurde im Handel erworben.

# Anhang B

## Programmstrukturen und Benutzungshinweise

In diesem Anhang werden die Systemstruktur und ihre Schnittstellen beschrieben. Die Entwicklung erfolgte in Ubuntu 10.10 in C++, unter Verwendung der OpenCV. Zum besseren Verständnis erfolgt zunächst eine kurze Beschreibung des Systemaufbaus.

### B.1 Struktur

Das entwickelte System zum Erkennen und Verfolgen der Bewegungen von Kletternden wurde in zwei unabhängige Programme aufgeteilt, die untereinander Daten austauschen (vgl. Abbildung 4.1). Die Projektverzeichnisse sind wie folgt aufgebaut:

```
MarkerDetection
+- include
+- media
| imagefiles.txt
+- output
+- src
```

```
MarkerTracking
+- lib
+- Tracker
| +- include
```

```
| +- media
|   markerfileList.txt
| +- src
```

Das Programm `MarkerDetection` zur Erkennung von Markern arbeitet auf einer Menge von Einzelbildern. Der Ordner `media` ist zum Ablegen der zu verarbeitenden Bildern vorgesehen. Das Programm speichert die Koordinaten der gefundenen Marker in Textdateien und legt sie im Ordner `output` ab. Die Bildpfade werden aus der Textdatei `imagefiles.txt` gelesen und automatisch verarbeitet.

Das Tracking-Programm `MarkerTracking` visualisiert die Markerkoordinaten in einer Nutzeroberfläche. Der Ordner `Tracker` enthält dabei den Code für die Nutzeroberfläche und `lib` den Code für die Realisierung des Condensation Algorithmus. Der Nutzer kann in dem Programm Partikelfilter setzen und damit die Bewegungen der Marker verfolgen lassen. Ihre Spuren werden als Koordinaten in einer Textdatei abgespeichert und exportiert. Im Projektverzeichnis von `MarkerTracking` befindet sich der Ordner `media`, aus dem die Koordinatendateien für die Visualisierung gelesen werden. Auch dieses Einlesen erfolgt automatisch über die Textdatei `markerfileList.txt`, die mit den Pfaden zu den Textdateien gefüllt wird.

## B.2 Vorbereitung der Bilddaten

Dieses Kapitel beschreibt die notwendigen Schritte, um das Programm zur Markerdetektion mit Bilddateien zu füllen und um es richtig zu benutzen.

### Video in Einzelbilder zerlegen

Mit Hilfe des Tools `mplayer` können Videos wie folgt in Einzelbilder zerlegt werden:

```
$ mplayer -vo png:z=5 /inputfile
```

Die Bilder werden hierbei im png-Format leicht komprimiert abgespeichert (1 = niedrigste, 9 = beste Qualität).

### Bilder rotieren und skalieren

Um gegebenenfalls die Einzelbilder automatisiert gegen den Uhrzeigersinn zu rotieren und zu skalieren, ist folgender Befehl im Bilderverzeichnis hilfreich:

```
$ for file in *.png;
> do convert -rotate "-90>" -resize 1024x1024 $file ../$file
> done
```

## B.3 Schnittstellen

Die zu untersuchenden Bilder werden in den Ordner `media` von `MarkerDetection` gelegt, um die Markererkennung durchzuführen. Zur automatischen Verarbeitung mehrerer Bilder ist die Textdatei `imagefiles.txt` mit den Dateipfaden zu den Bildern (aus Sicht des Codeverzeichnisses) zu füllen. Die Auflistung der Dateipfade ist über einen Terminal-Befehl möglich:

```
$ ls --format single-column media/*.png > media/imagefiles.txt
```

Die Pfade werden hier direkt in die Textdatei geschrieben. Es ist darauf zu achten, dass die Textdatei keine leeren Zeilen enthält.

### Datenübergabe

Um die Ergebnisse der Markerdetektion beim Tracking verwenden zu können, werden die Textdateien mit den Markerkoordinaten in den Ordner `media` der Tracking-Software kopiert. Zur Verwaltung der Visualisierung befindet sich hier eine Textdatei `markerfileList.txt`, die die Pfade zu den Ergebnisdateien enthalten soll. Die Generierung der Liste an Dateipfaden erfolgt ähnlich wie bereits zuvor:

```
$ ls --format single-column media/*.txt > media/markerfileList.txt
```

Auch hier ist darauf zu achten, dass in der Textdatei keine leeren Zeilen und keine falschen Dateipfade stehen.

## B.4 Bedienung der Tracking-Software

Das folgende Kapitel beschreibt die Funktionen der Tracking-Software. Für die Erklärung der Nutzeroberfläche wird Abbildung 4.5 betrachtet:

- **Visualisierungsbereich:** Hier werden die Markerkoordinaten als kleine Quadrate dargestellt und animiert. Durch einen Klick in den Bereich kann ein Partikelfilter gesetzt werden.
- **Filter Control:** Zum einfachen Starten bzw. Pausieren der Animation wird zwischen den Knöpfen *Loop Mode* und *Single Step Mode* gewechselt. Nach einmaligem Durchlauf der Animation wechselt das Programm automatisch in den Einzel-schritt-Modus.
- **Single Step Optionen:** Befindet sich das Programm im Einzelschritt-Modus, können die Schritte des Condensation Algorithmus einzeln durchgeführt werden.

- **Export Control:** Für die Aufzeichnung der Markerverfolgung wird die Schaltfläche *Record Tracking* betätigt, um das Tracking direkt zu starten. Ist die Animation beendet, kann anschließend die Schaltfläche *Export* zum Export der Daten für Simi Motion gedrückt und ein Ort zum Abspeichern der Textdatei bestimmt werden.

# Literaturverzeichnis

- [AMGC02] ARULAMPALAM, S. ; MASKELL, S. ; GORDON, N. ; CLAPP, T.: A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. In: *IEEE Transactions on Signal Processing* 50 (2002), Nr. 2, S. 174–188
- [DFG01] DOUCET, Arnaud (Hrsg.) ; FREITAS, Nando d. (Hrsg.) ; GORDON, Neil (Hrsg.): *Sequential Monte Carlo Methods in Practice*. LLC, 175 Fifth Avenue, New York, NY 10010, USA : Springer-Verlag, 2001
- [FLB03] FIGUEROA, Pascual J. ; LEITE, Neucimar J. ; BARROS, Ricardo L.: A flexible software for tracking of markers used in human motion analysis. In: *Computer Methods and Programs in Biomedicine* Bd. 72, Elsevier Science, 2003, S. 155–165
- [Han08] HANDELS, Jochen: *Vergleich von 2d- und 3d-Bewegungsmodellen für die Vorhersage der Bewegung eines Sportlers beim Hochgeschwindigkeitstracking*, Universität Koblenz-Landau, Campus Koblenz, Diplomarbeit, 2008
- [Hom07] HOMBERG, Ulrike: *Multi-Objektverfolgung in Farbbildsequenzen auf der Basis von Partikelfiltern*, Otto-von-Guericke Universität Magdeburg, Diplomarbeit, 2007
- [IB98] ISARD, M. ; BLAKE, A.: CONDENSATION - conditional density propagation for visual tracking. In: *International Journal of Computer Vision* 29 (1998), Nr. 1, S. 5–28
- [Kan] *Kandi-Turm Kletterhalle*. Stadionstr. 77, 56626 Andernach, . – <http://www.kanditurm.de>
- [Koc00] KOCH, Karl-Rudolf: *Einführung in die Bayes-Statistik*. Springer, Berlin, Heidelberg, New York, 2000

- [Pap02] PAPOULIS, Athanasios: *Probability, Random Variables and Stochastic Processes*. 4. McGraw Hill, 2002
- [SA85] SUZUKI, Satoshi ; ABE, Keiichi: Topological structural analysis of digitized binary images by border following. In: *Computer vision, graphics, and image processing* 30 (1985), Nr. 1, S. 32–46
- [Sch05] SCHWARZENBERG, Gregor: *Objektverfolgung mit Partikel-Filtern*, Universität Karlsruhe, Fakultät für Informatik, Institut für interaktive Echtzeitsysteme, Diplomarbeit, 2005
- [Sim] *Simi Reality Motion Systems GmbH*. <http://www.simi.com>,
- [Ste08] STEINMÜLLER, Johannes: *Bildanalyse: Von der Bildverarbeitung zur räumlichen Interpretation von Bildern*. 1. exam.press. Springer-Verlag, Berlin, 2008
- [WHT02] WANG, Liang ; HU, Weiming ; TAN, Tienuiu: *Recent Developments in Human Motion Analysis*. 2002
- [Wir07] WIRTH, Stephan: *Visual Underwater Cable/Pipeline Tracking*, Universität Koblenz-Landau, Campus Koblenz, Fachbereich 4 Informatik, Institut für Computervisualistik, Diplomarbeit, 2007