Personen-Tracking in Umgebungen mit verdeckenden Objekten basierend auf 3D-Rekonstruktionsdaten

Von der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

von

Antje Ober-Gecks

aus Bad Frankenhausen

1. Gutachter: Prof. Dr. Dominik Henrich

2. Gutachter: Prof. Dr. Michael Guthe

Tag der Einreichung: 12.04.2022

Tag des Kolloquiums: 18.11.2022

Zusammenfassung

Das Verfolgen (Tracking) von Personen im Arbeitsraum von Robotern ermöglicht die Vorhersage menschlicher Bewegungen, die Abschätzung von Intentionen und kann die 3D-Lokalisierung der Personen und somit die Sicherheit der Mensch-Roboter-Kooperation verbessern. Deshalb ist das Tracking von großer Bedeutung für diesen Anwendungsbereich. Durch die Präsenz statischer Objekte im Arbeitsraum können Verdeckungsvolumina entstehen, die mit optischer Sensorik nicht einsehbar und zugleich leer sind. Begeben sich dynamische Objekte wie Personen in solche Volumina, so werden sie ganz oder teilweise verdeckt, was die Objektdetektion und -segmentierung in den Sensordaten einschränkt. Treten stärkere Verdeckungen über mehrere Frames auf, so führt dies häufig zu einem Tracking-Verlust der betroffenen Objekte. Diese Problematik ist Untersuchungsgegenstand der vorliegenden Dissertation, in welcher eine Möglichkeit aufgezeigt wird, erfolgreiches Tracking auch in solchen Situationen durchzuführen. Der gewählte Lösungsansatz besteht darin, zusätzliches Wissen zu den vorliegenden statischen Objekten und den durch sie erzeugten Verdeckungsvolumina in ein Verfahren des Personen-Trackings zu integrieren.

Als Sensorik wird ein Multi-View-Kamerasystem verwendet, das Farbbilder des Arbeitsraums aus unterschiedlichen Perspektiven aufzeichnet. Basierend auf den Kamerabildern wird zur Objektdetektion ein Background Subtraction durchgeführt. Mit den daraus resultierenden Silhouettenbildern aller Kameras wird zu jedem betrachteten Zeitpunkt die Visuelle Hülle erkannter Personen rekonstruiert. Dabei wird Wissen zu den gegebenen statischen Objekten in den Rekonstruktionsprozess einbezogen, um Fehler, die durch Verdeckungen der Personen entstehen, zu vermeiden. Die Visuelle Hülle wird als Voxeldatenstruktur abgelegt und dient als Eingabe für ein Personen-Tracking mittels Partikelfilter im Zustandsraum. Als Objektmodell wurde ein Ellipsoid gewählt, was die Dimensionalität des Zustandsraums beschränkt und so vorteilhaft für die Anwendung des Partikelfilters ist. Jede Person wird mit einer eigenen Partikelfilterinstanz getrackt, wobei für den Umgang mit Datenassoziationsproblemen, die typischerweise beim Mehrpersonen-Tracking entstehen, eine Blocking-Methode eingesetzt wird.

Der zentrale Beitrag dieser Dissertation ist der Entwurf einer Likelihood-Funktion für den Partikelfilter, in der verschiedene Voxelzustände berücksichtigt werden, welche Wissen zu den Belegungs- sowie Verdeckungsinformationen des Arbeitsraums kodieren. Die Voxelzustände werden innerhalb der Likelihood-Funktion des Partikelfilters unterschiedlich gewichtet, wobei verschiedene Varianten von Gewichtungsfunktionen anhand des resultierenden Filterverhaltens untersucht und gegenübergestellt werden. Es wird

gezeigt, dass mit einer geeigneten Gewichtung der Voxelzustände das Tracking von Personen durch Verdeckungsvolumina ermöglicht wird, ohne eine Terminierung des Filters (Tracking-Abbruch) hervorzurufen. Dabei können partielle ebenso wie vollständige Verdeckungen der Personen vorliegen, deren Dauer prinzipiell uneingeschränkt sein darf. Zur Vermeidung unerwünschten Diffundierens der Filter durch statische Objekte hindurch wird ein spezieller Kollisionstest entworfen, der in verschiedenen Situationen einen Vorteil erbringen kann.

Abstract

Person tracking in robot work spaces enables predicting humans' movements, estimating or infering their intentions, and increases interaction safety through improved 3D person localization. This is of great importance for human-robot cooperation. However, the presence of static objects in the work space can lead to occluded volumes that are empty and not observable to optical sensor systems at all. Dynamic objects such as persons entering these volumes may become partially or completely occluded and thus, detection and segmention in the sensor data is affected. Stronger occlusions of the objects of interest that persist for several sensor frames usually result in tracking loss. This is the object of investigation in the presented dissertation, which proposes an approach to enable successful tracking in the described scenario. To achieve this, prior knowledge of the static objects and their occluded volumes is integrated into the tracking method.

A multi-view camera system is used to capture color images from different perspectives. Based on these images a background subtraction is applied for object detection, which provides silhouette images for all cameras of the humans present. Those are then used to compute a visual hull at each time step. Knowledge of static objects is integrated in this process to avoid errors in the reconstruction results that arise due to occlusions of the persons. The visual hull is stored as voxel data and serves as input for person tracking, accomplished in state space via particle filters. An ellipsoid is used as the object model for tracking, limiting state space dimensionality, which is advantageous for the particle filter. Each person is tracked with a separate particle filter instance. Data association problems typically occuring in multiple-object tracking scenarios are solved with a blocking method.

The main contribution of this dissertation is the design of a likelihood function for the particle filter that considers different voxel states encoding knowledge about occupation and occlusions of the working space. The voxel states are weighted differently in the likelihood function. Various weighting functions are investigated and compared by evaluating the resulting particle filter behaviors. It is shown that an appropriate weighting of the voxel states enables person tracking through occluded volumes without causing filter termination or tracking loss. Tracked persons may be partially or completely occluded, with occlusion duration being unlimited in principle. Furthermore, to avoid unwanted (physically impossible) particle filter diffusion through static objects, a special collision test is presented that may be helpful in specific situations.

Danksagung

Das Vorhaben Promotion war eine längere Phase meines Lebens. Daher möchte ich die Chance ergreifen, den Menschen, die mich auf meinem Weg begleitet haben, meinen persönlichen Dank zum Ausdruck zu bringen.

An erster Stelle möchte ich meinem Doktorvater Prof. Dr. Dominik Henrich danken für die jahrelange Unterstützung, sein Verständnis für verschiedene Lebensumstände sowie für seine anhaltende Zuversicht bezüglich der Fertigstellung meiner Dissertation. Der Firma Elan Schaltelemente GmbH gilt mein Dank für die finanzielle Förderung meines dreijährigen Drittmittelprojekts „Safety Vision“, in dem ich eine Vielfalt an Erfahrungen sammeln konnte. Der Stabsabteilung Chancengleichheit unter Leitung von Miriam Bauch danke ich vielmals für mein Stipendium und finanzierte HiWi-Tätigkeiten sowie die hilfreichen Angebote, die ich für meine Weiterentwicklung nutzen konnte.

Ein ganz besonderes Dankeschön für das sorgfältige Lesen von Teilen meiner Arbeit sowie die wertvollen und konstruktiven Anregungen gilt Constanze Ober, Jan Schilbach, Thorsten Gecks, Kathrin Juhart und Philipp Stolka.

Meinen ehemaligen Kollegen und Kolleginnen der Universität Bayreuth danke ich sehr für den fachlichen Austausch, die schönen gemeinsamen Zeiten und so manche tiefergehende Freundschaft, insbesondere: Thorsten Gecks, Stefan Kuhn, Tobias Werner, Philipp Stolka, Marc Schütz, Simon Heckl, Maria Hänel, Christian Groth, Michael Spangenberg, Markus Fischer, Jan Deiterding, Katharina Barth, Michael Gradmann, Maximilian Sand, Anke Paul, Susanne Süß sowie Marvin Ferber, Berit Leo und Lydia Bodner.

Für das engagierte Bearbeiten der Aufgabenstellungen und die fachlichen Beiträge bedanke ich mich bei „meinen“ Studenten: Malte Munder, Florian Bäuerlein, Marius Zwicker, Holger Kastner, Christian Groth, Philipp Struntz, Rolf Fickentscher, Johannes Völkel, Lars Ackermann, Kim Wölfel, Matthias Höger, Fabian Lorenz, Vladislav Stoychev und allen anderen.

Meinem Mann Thorsten Gecks gilt ein Dank aus tiefstem Herzen für seine Geduld, das intensive Betreuen unserer Kinder und die Zurückstellung seiner Bedürfnisse und Ziele zugunsten der Weiterverfolgung meiner Promotion. Meinen Eltern danke ich ebenso herzlich für die bedingungslose Unterstützung all meiner bisherigen Vorhaben. Meiner Mutter gilt ein besonderer Dank dafür, dass sie mir einen zügigen Wiedereinstieg in die Arbeitswelt nach dem ersten Kind ermöglicht hat, indem sie wöchentlich zur Kinderbetreuung angereist kam. Meiner Schwester Constanze Ober danke ich aus

vollstem Herzen für ihr Zuhören, ihre große Anteilnahme an meinen „Projekten“ sowie ihre garantierte Unterstützung in allen Lebenslagen. Meinem Bruder Cornelius Ober danke ich für seine moralische Bestärkung. Bei meinen Kindern bedanke ich mich zutiefst für ihre Geduld mit ihrer Mama und den Verzicht, den sie aufgrund der Promotion an verschiedenen Stellen üben mussten.

Bedanken möchte ich mich weiterhin bei allen Freunden und Verwandten, die mich ermutigt und Verständnis für meine oft eingeschränkte Verfügbarkeit aufgebracht haben, insbesondere bei: Anja Hoffmann, Erika Behnsen, Rike Brecht, Kathrin Juhart, Nadja Schuler, Simone Hülper und Franz Grosse. Ich danke euch sehr für Eure Treue, Eure Verbundenheit und Euren guten Rat.

Für ihren sehr engagierten und wertvollen ehrenamtlichen Einsatz für eine nachhaltige und gerechtere Welt bedanke ich mich bei meinen Mitstreitern und Mitstreiterinnen: Franz Grosse, Heidemarie Karch, Alexander Clauß, Sia Bauer, Jasmin Oldag, Sandra Szostak, Philipp Haslach, Carina Bezold, Thorsten Gecks (schon wieder) sowie vielen weiteren.

Widmen möchte ich meine Dissertation meinen lieben Freundinnen Kathrin Juhart und Rike Brecht.

Inhaltsverzeichnis

Abbildungsverzeichnis

Algorithmenverzeichnis

Kapitel 1

Einleitung

Zu Beginn dieser Dissertation wird in Abschnitt 1.1 ein historischer und aktueller Überblick zur Vision von Mensch-Roboter-Kooperationen gegeben. In diesem Zusammenhang wird der Einsatz von Tracking-Verfahren in Abschnitt 1.2 motiviert und auf damit verbundene Herausforderungen eingegangen. Die Arbeitsweise von Tracking-Verfahren wird detaillierter in Abschnitt 1.3 beschrieben und im allgemeinen Kontext der Sensordatenverarbeitung eingeordnet, um verwandte Verfahrensschritte davon abzugrenzen. Da es verschiedenste Anwendungen für Tracking-Verfahren gibt, erfolgt in Abschnitt 1.4 eine Beschreibung des betrachteten Anwendungsszenarios, für welches die Aufgabe des Personen-Trackings untersucht werden soll. Die Zielstellung dieser Dissertation wird in Abschnitt 1.5 konkretisiert, bevor in Abschnitt 1.6 eine Kapitelübersicht gegeben wird.

1.1 Vision von Mensch-Roboter-Kooperationen

Mit der Entwicklung elektrischer Antriebssysteme entstanden im Zuge der industriellen Revolution viele neue Maschinen, die ganze Arbeitsprozesse bei der Herstellung von Produkten übernehmen konnten. Seitdem hat sich das Leben in den Industrieländern deutlich vereinfacht und Maschinen sind alltäglich geworden. Die Maschinen führen Tätigkeiten aus, die für den Menschen körperlich schwer zu bewältigen, einseitig belastend oder sehr zeitaufwändig sind, wie z. B. die Produktion von Lebensmitteln. Mittlerweile schreitet aufgrund der Fortschritte in der Computertechnik, auch Digitalisierung genannt, die Entwicklung von **Robotern** voran. Der Begriff Roboter wird für sehr unterschiedliche Arten von Maschinen verwendet, denen gemein ist, dass sie meist von einem komplexen Computerprogramm gesteuert werden. Roboter sollen multifunktional für diverse Aufgaben und flexibel in unterschiedlichen und variierenden Umgebungen einsetzbar sein, was bei bisherigen Maschinen nicht oder nur begrenzt gegeben ist. Betrachtet man die derzeit auf dem Markt verfügbaren Produkte, so lässt sich grob zwischen **Industrierobotern** und **Servicerobotern** unterscheiden, die für verschiedene Aufgabenbereiche konzipiert werden. Während Industrieroboter als Automatisierungstechnik in der Fertigung von Produkten eingesetzt werden, sollen

Serviceroboter vorwiegend Dienstleistungen für den Menschen in seiner natürlichen Umgebung erbringen und beispielsweise Tätigkeiten im Haushalt verrichten. Gegenwärtig häufig anzutreffende Serviceroboter sind kleine mobile Geräte, die selbstständig Spezialaufgaben erledigen, wie z. B. Rasenmähen oder Staubsaugen.

Eine Vision im Bereich der Servicerobotik ist die Entwicklung menschenähnlicher Roboter, genannt **Humanoide** (auch Androide). Ein Humanoide ist das, was den Begriff Roboter ursprünglich in Literatur und Fiktion geprägt hat, z. B. in [Capek, 1920] und [Asimov, 1950], bevor es zur realen Entwicklung von Robotern kam. Beim Humanoiden steht die **Interaktion** mit dem Menschen im Vordergrund. Er soll mit dem Menschen natürlichsprachlich kommunizieren, ihm als Partner zur Seite stehen und ihn im Alltag unterstützen. Dafür muss ein Humanoide viele Fähigkeiten besitzen. Er muss Menschen und Objekte seiner Umgebung erkennen, sich mobil in natürlichen Umgebungen bewegen und manipulatorisch darin agieren können. Er benötigt einen hohen Grad an Autonomie (Selbstständigkeit), muss adaptiv (anpassungsfähig) sein, ein gutes Gedächtnis besitzen und sich weiterentwickeln können. Nur durch diese Fähigkeiten kann ein Roboter angemessen und flexibel auf die veränderlichen Zustände seiner Umgebung und seine Interaktionspartner reagieren. Er sollte so mit dem Menschen interagieren, wie dieser es von seinen Mitmenschen her gewohnt ist und wie dieser es von einer intelligenten Kopie des Menschen erwartet.

Als Industrieroboter werden derzeit Roboterarme (auch Manipulatoren genannt) verstanden, die im Vergleich zu Servicerobotern schon länger im Einsatz sind. Ein Roboterarm ist ein programmierbares Mehrzweckhandhabungsgerät für das Bewegen von Objekten. Es besitzt mehrere Achsen, deren Bewegungen frei programmiert und gegebenenfalls sensorgestützt gesteuert werden können. Roboterarme können mit Werkzeugen, Greifern und sonstigen Mitteln versehen werden, sodass sie Handhabungs- oder Fertigungsaufgaben bewältigen können. Mit Hilfe von Roboterarmen lassen sich Prozesse optimieren, die vielfach ausgeführt werden müssen, das Bewegen schwerer Lasten beinhalten oder eine hohe Präzision bei ihrer Durchführung erfordern. Durch die möglichen hohen Bewegungsgeschwindigkeiten von Industrierobotern kann Zeit eingespart werden, was in der Regel mit einer Reduktion von Produktionskosten verbunden ist.

Der bisherige Einsatz von Roboterarmen ist vorwiegend auf die industrielle Fertigung von Produkten beschränkt, aufgrund des erheblichen Programmieraufwands, mangelnder Reaktionsfähigkeiten auf veränderliche Umgebungen sowie die erforderliche Absicherung des Menschen. Zukünftig sollen jedoch die Einsatzbereiche und das Aufgabenspektrum der Industrieroboter erweitert werden. So sollen Roboter ihre Tätigkeiten in Räumen verrichten können, in denen sich auch Menschen befinden, ohne dabei eine Gefahr für diese darzustellen, was als **Mensch-Roboter-Koexistenz** bezeichnet wird.

Weitergehend sollen Roboter auch in direkter Zusammenarbeit mit dem Menschen Aufgaben bewältigen können, genannt **Mensch-Roboter-Kooperation**. Hierbei werden Synergieeffekte aus der Kombination der Stärken von Mensch und Roboter erwartet, wodurch bestimmte Aufgaben, z. B. bei der Feinmontage, effizienter und für den Menschen entlastender gestaltet werden können. Zur Umsetzung dieser Ziele eröffnen sich insbesondere durch die handhabbaren Leichtbauroboter neue Möglichkeiten. So sollen kleine Industrieroboter als leicht bedienbares und intuitiv programmierbares Werkzeug alltagstauglich gemacht werden und damit flexibel in Werkstätten oder im Haushalt, z. B. als Küchenmaschine, eingesetzt werden können. Dies ist insbesondere auch für kleinere Unternehmen sehr attraktiv, bei denen Produkte in nur kleinen Losgrößen hergestellt werden.

Zusammenfassend ist die Vision in der Robotik – sowohl in der Servicerobotik als auch in der Industrierobotik – die Umsetzung von mehr Interaktivität und Zusammenarbeit zwischen Mensch und Roboter.

1.2 Motivation zum Personen-Tracking

Für viele der vom Menschen gewünschten Interaktionen benötigt ein Roboter kognitive Fähigkeiten, die es ihm ermöglichen, Signale aus der Umwelt wahrzunehmen und weiterzuverarbeiten. Nur damit lässt sich auch ein reaktives und adaptives Verhalten des Roboters bei der Interaktion mit dem Menschen realisieren. Hierfür werden Sensoren zur Aufzeichnung von Umgebungsdaten eingesetzt, beispielsweise Kameras, Lasersensoren oder Sonarsensoren. Die aufgezeichneten Daten werden verarbeitet, um die benötigten Informationen aus den Messungen zu extrahieren, genannt **Sensordatenverarbeitung**.

Eine grundlegende Funktionalität, welche die Auswertung von Umgebungsdaten erfordert, ist die (intelligente) **sensorgestützte Bahnplanung** von Roboterarmen. Bislang sind schwere Roboterarme in der Industrie nur mit wenigen oder gar keinen Sensoren und Wahrnehmungsfähigkeiten ausgestattet. Störungen im Bewegungsablauf werden verhindert, indem diese in definierten Umgebungen mit bekannten Zuständen eingesetzt werden. Bei veränderlichen Umgebungen mit dynamischen Objekten wie Menschen wird jedoch eine online durchgeführte, sensorgestützte Bahnplanung benötigt. Die unbekannten Objekte in der Umgebung des Roboters werden dabei ständig sensorisch lokalisiert, um daraus zu jedem Zeitpunkt die Raumbelegung und den Freiraum abzuleiten. Der Roboter darf bei seiner Bahnplanung nur Freiraum berücksichtigen, um kollisionsfreie Eigenbewegungen ausführen zu können. **Verdeckungen**, welche durch dynamische und statische Objekte entstehen und die Sicht des Roboters einschränken, stellen dabei eine Herausforderung dar. Einerseits kann dadurch die Bestimmung des Freiraums erschwert

werden. Andererseits können diese bei der Bahnplanung eine besondere Behandlung erfordern. Beispielsweise könnte sich eine Person unerkannt aus einer Verdeckung auf den Roboter zubewegen, was ein hohes Verletzungsrisiko mit sich bringt. Die Sicherheit des Menschen muss bei der sensorgestützten Bahnplanung – auch im Vorhandensein von Verdeckungen – gewährleistet werden.

Für die zukünftige Mensch-Roboter-Kooperation ist allerdings nicht nur die Kollisionsfreiheit und Sicherheit des Menschen bei den Roboterbewegungen von Bedeutung, sondern auch die Feinabstimmung zwischen Bewegungen des Menschen und des Roboters. Bei einer aktiven Zusammenarbeit, wie der gemeinsamen Bewältigung manipulatorischer Aufgaben, soll für den menschlichen Arbeiter das Gefühl eines natürlichen Miteinanders entstehen. Die Bewegungen sollen flüssig, ohne lange Wartezeiten, abruptes Stehenbleiben oder aufwändige und für den Menschen möglicherweise unangenehme Richtungswechsel zur Kollisionsvermeidung ablaufen. Es wird eine vorausschauenende kollisionsfreie Bahnplanung des Roboters benötigt, die wiederum eine **Bewegungsvorhersage** und **Intentionsschätzung** des Menschen erfordert.

Verschiedene Informationen können dabei aus den menschlichen Bewegungen selbst gewonnen werden, denn diese enthalten Informationen mit deren Hilfe auf (verborgene) Zustände und Intentionen der Personen geschlossen werden kann. Der Mensch macht sich dies ständig durch die Beobachtung seiner Mitmenschen zunutze. Bei der Kommunikation erkennt und interpretiert er Feinbewegungen der Mimik und Gestik. Bei der Planung seiner Eigenbewegungen durch den Raum nimmt er **Prädiktionen** (Vorhersagen) zu den Bewegungszielen anderer Personen in seinem direkten Umfeld vor. Damit schätzt der Mensch implizit zukünftige Raumbelegungen ab, was es ihm ermöglicht, Kollisionen zu vermeiden und einen eigenen (energie-)optimierten Weg durch den vorhergesagten Freiraum zu planen. Menschen können die Bewegungen, Ziele und Intentionen anderer Menschen durch die Analyse ihrer Bewegungen vorhersagen. Ein Roboter sollte solche Vorhersagen zumindest soweit treffen können, dass er intelligente adaptive Bewegungen ausführen kann, die insbesondere bei einer stark interaktiven Mensch-Roboter-Kooperation benötigt werden.

Als Voraussetzung für die Vorhersage von Personen- und Objektbewegungen kann die Bewegungsverfolgung – das sogenannte **Tracking** – betrachtet werden. Beim Tracking wird zu jedem Zeitpunkt anhand der vorangegangenen Bewegungshistorie sowie einer kurzfristigen Bewegungsvorhersage der aktuelle Zustand einer Person oder eines Objekts geschätzt. Diese Dissertation befasst sich mit dem Tracking von Personen in einer Roboterarbeitszelle. Das primäre Ziel besteht darin, die Personen permanent lokalisieren zu können. Der Fokus liegt auf dem herausfordernden Umgang mit sensorischen **Verdeckungen**. Ebenso wie bei der sensorgestützten Bahnplanung von Robotern leidet auch

die Bewegungsverfolgung und Bewegungsvorhersage von Personen unter Verdeckungen. Sind Personen teilweise oder vollständig verdeckt, so wird die Bewegungsvorhersage erschwert. Sie kann aufgrund von Mehrdeutigkeiten zu falschen Ergebnissen führen oder auch vollständig ausfallen. Auch die Klassifikation von Bewegungen kann erschwert werden oder falsch sein, wenn Teile der Bewegungshistorie fehlen.

Der Umgang mit Verdeckungen wird in dieser Dissertation folgendermaßen adressiert: Als globale Sensorik wird ein **Multi-View-Kamerasystem** zur Reduktion von Verdeckungen eingesetzt, welches Farbbilder der Arbeitszelle aus unterschiedlichen Perspektiven liefert. Aus diesen Daten wird eine 3D-Rekonstruktion generiert, genannt **Visuelle Hülle**, die dem Tracking-Verfahren als Eingabe dient. Für das Tracking selbst wird ein Partikelfilter eingesetzt. Wissen zu gegebenen statischen Verdeckungen und Objekten wird sowohl in den Prozess der Rekonstruktion als auch in das Tracking-Verfahren integriert. Letzteres stellt dabei den Neuigkeitswert der durchgeführten Betrachtungen dar.

1.3 Einordnung von Tracking-Verfahren

Betrachtet man das Ziel der Vorhersage menschlicher Bewegungen, so spielen die in Abbildung 1.1 dargestellten Verfahrensklassen eine Rolle.

Ein häufig eingesetzter Verarbeitungsschritt wird als **Objektdetektion** bezeichnet. Hierbei werden die Messdaten identifiziert, die zu den gesuchten Objekten von Interesse gehören. Bei Kamerabildern müssen beispielsweise für eine Person die Pixel bestimmt werden, auf welche diese abbildet, um weitere Analysen durchführen zu können. Zur Lösung dieser Aufgabe kommen abhängig vom Schwierigkeitsgrad Verfahren der „Low- und Mid-Level-Verarbeitung“ wie die Segmentierung zum Einsatz, aber auch Verfahren mit semantischem Bezug wie eine Merkmalsextraktion oder Klassifikation. Die Aufgabe der Objektdetektion ist nicht immer einfach lösbar, sondern kann einige Herausforderungen mit sich bringen. So können die Messungen und extrahierten Merkmale zumindest zeitweise ungenügend sein, um Objekte ausreichend gut zu detektieren. Zudem kann es Mehrdeutigkeiten bei der Zuordnung von Messungen zu Objekten geben, beson-

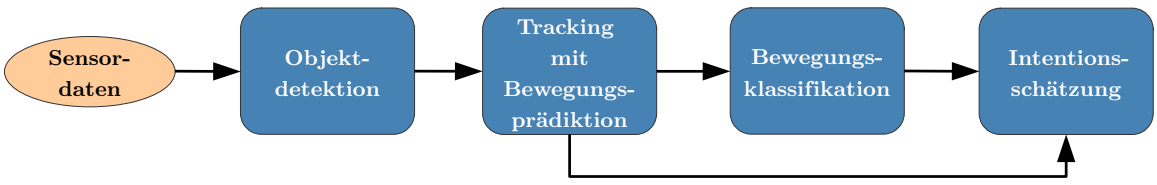


Abb. 1.1: Aufeinander aufbauende Verfahrensklassen der Sensordatenverarbeitung, die für eine Vorhersage menschlicher Bewegungen und eine Intentionsschätzung von Bedeutung sind.

ders dann, wenn diese nah beieinander liegen, sich aus Sicht der Kamera gegenseitig verdecken und ähnliche Eigenschaften, wie z. B. die gleiche Farbe, besitzen. Dieses Zuordnungsproblem ist auch unter dem Begriff **Datenassoziationsproblem** bekannt.

Zur Lösung von Problemen der Objektdetektion und Datenassoziation kann häufig ein **Tracking-Verfahren** beitragen. Unter dem Begriff Tracking wird eine Verfolgung von Objekten über zeitliche Sequenzen von Messdaten, z. B. Videosequenzen, verstanden. Dabei werden Korrelationen aufeinanderfolgender Messungen genutzt, um mehr Robustheit in die Auswertung der Messdaten zu bringen. Die Historie vorangegangener Messungen wird in die Analyse und Bewertung der jeweils aktuellen Messung einbezogen. Beim Tracking werden Informationen zur Geometrie und zum Aussehen der Objekte, genannt **Appearance**, verwendet, nach denen in den Messdaten gesucht wird. Der Suchbereich wird dazu durch einen Mechanismus der Bewegungsvorhersage (Prädiktion) eingeschränkt. In einfachen Fällen kann dieser auf Annahmen an die Bewegungen, wie z. B. eine begrenzte Bewegungsgeschwindigkeit, Trägheit und Stetigkeit, beruhen. Häufig werden beim Tracking nicht nur statische Eigenschaften von Objekten bzw. Personen verwendet, sondern parametrisierbare (geometrische) **Objektmodelle**, die an den Tracking-Verlauf angepasst werden können. Die Schätzung der nicht konstanten Parameter eines Objektmodells erfolgt mit jeder verfügbaren Messung. Dies wird ganz allgemein auch als **Zustandsschätzung** bezeichnet, wenn das Tracking im **Zustandsraum** erfolgt, zu dem typischerweise die Parameter des Objektmodells gehören. Durch die Verwendung von Zustandsschätzungen erhalten die verrauschten Messdaten weniger Gewicht, wodurch die Objektdetektion oft implizit verbessert werden kann. Zudem können Bewegungen als Aneinanderreihung von Zuständen im Zustandsraum beschrieben werden, was Vorteile mit sich bringen kann (z. B. die kompakte Repräsentation der Merkmalsvektoren für eine nachfolgende Klassifikation).

Durch die Anwendung eines Tracking-Verfahrens werden Bewegungen in Form von **Trajektorien** erfasst, welche die Wege der Objekte im Mess- oder Zustandsraum mit zeitlichen Informationen beschreiben. Durch die Analyse aufgezeichneter **Trajektorien** können semantische Informationen zu den Bewegungen gewonnen und Bewegungen erkannt werden. Hierfür kommen Methoden der Mustererkennung zum Einsatz, die ähnliche Trajektorien gruppieren und Bewegungsklassen bilden, die sich semantisch unterscheiden, was auch als **Bewegungsklassifikation** bezeichnet wird. Diese Aufgabe ist allerdings oft nicht trivial, da Trajektorien in Bezug auf ihre geometrischen Eigenschaften als auch ihre zeitlichen Verläufe sehr stark variieren können [Ober, 2007]. Dadurch können mitunter nicht genügend Unterscheidungsmerkmale extrahiert werden, die eine Zuordnung zu den gewünschten semantischen Klassen ermöglichen.

Zur Umsetzung vorausschauender sensorgestützter Bahnplanungen von Robotern

für die Ermöglichung kontinuierlicher Roboterbewegungen bei der Mensch-Roboter-Kooperation ist sowohl das Tracking als auch die Bewegungsklassifikation von Bedeutung. Beim Tracking wird durch eine Prädiktionskomponente eine kurzfristige Bewegungsvorhersage des nächsten Bewegungsabschnitts erzielt. Die Klassifikation ganzer Trajektorien oder Teilen davon bietet hingegen das Potential, auch übergeordnete Zusammenhänge von Bewegungen wie ganze Bewegungsfolgen sowie Zustände von Tätigkeiten vorherzusagen, die häufig mit lokalen Zielen sowie definierten Arbeitsabläufen zusammenhängen. Die Zuordnung von Trajektorien zu vorab gelernten Bewegungsklassen sollte dabei möglichst früh während ihrer Entstehung (online) erfolgen. Durch die Kombination von Verfahren des Trackings und der Bewegungsklassifikation sollte ein Roboter in der Lage sein, sowohl auf kurzfristige Bewegungsänderungen des Menschen zu reagieren, ohne Kollisionen zu verursachen, als auch seine Bahnplanung global auf die Bewegungsziele des Menschen anzupassen. In diesem Zusammenhang kann bei der Schätzung übergeordneter Ziele im Arbeitsverlauf auch von **Intentionsschätzung** gesprochen werden.

1.4 Anwendungsszenario

An dieser Stelle soll das Anwendungsszenario genauer spezifiziert werden. Die Betrachtungen zu dem Personen-Tracking in dieser Dissertation lassen sich dem Bereich der Online-Überwachung zuordnen (engl. Surveillance). Da Personen prinzipiell in beliebigen Umgebungen überwacht und getrackt werden können, gehen die Charakteristika der Umgebungen mitunter deutlich auseinander. Diese Charakteristika führen zu unterschiedlichen Ausprägungen typischer Probleme bei der Objektdetektion und beim Tracking, wie z. B. Störungen durch Beleuchtungsänderungen, Strukturierungen der Umgebung oder sensorische Verdeckungen. Die Eigenschaften von Umgebung und Anwendung wirken sich deshalb sowohl auf die Wahl der Art der Sensorik und Hardware als auch auf die Wahl des Tracking-Verfahrens aus.

Im Fokus steht ein Personen-Tracking in Innenräumen mit Ausmaßen von typischerweise wenigen Metern. Dazu kann auch eine Roboterarbeitszelle gezählt werden. Solch ein Überwachungsraum enthält oft Gegenstände und wird häufig durch Wände begrenzt, die zu eingeschränkten Zugangsbereichen führen. Die betrachteten Umgebungen sind überwiegend statisch. Das heißt, die Räumlichkeiten ändern sich oft nur gering und selten in größerem Maßstab. Alltagsgegenstände oder Arbeitswerkzeuge, die typischerweise in solchen Räumen bewegt werden und zu Änderungen der Szene führen, weisen oft eher kleinere Ausmaße auf. In dieser Dissertation findet das Tracking in der SIMERO-Roboterarbeitszelle statt, die in Abb. 1.2 dargestellt ist (Auf das SIMERO-Projekt

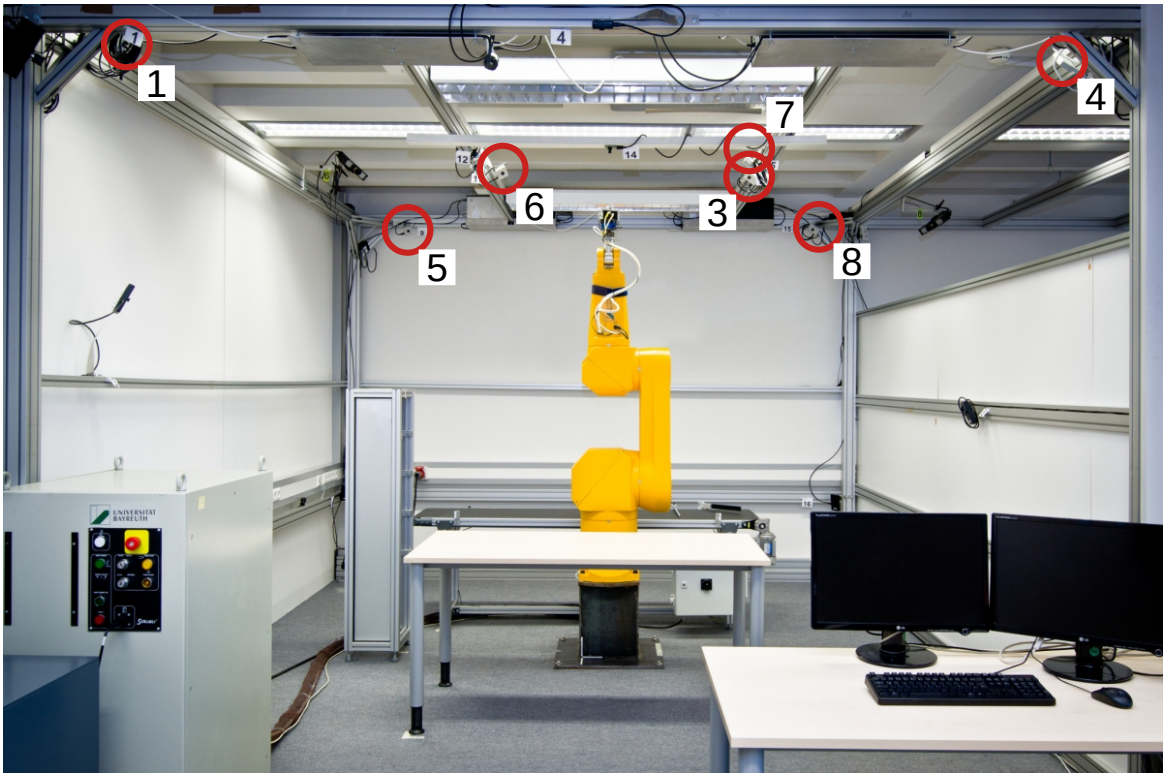


Abb. 1.2: Die SIMERO-Roboterarbeitszelle des Lehrstuhls für Angewandte Informatik III der Universität Bayreuth, in welcher die Videosequenzen für das Tracking dieser Dissertation aufgezeichnet wurden. Die verwendeten Kameras sind mit Kreisen markiert (rot) und nummeriert. Bild ohne Markierungen entnommen aus [Haenel, 2015, S. 128].

wird in Abschnitt 2.3 näher eingegangen.). Die Zelle besitzt eine Größe von etwa $4{,}4\,\mathrm{m} \times 4\,\mathrm{m} \times 2{,}5\,\mathrm{m}$.

Blockiert ein Objekt den Messbereich eines Sensors, so kann dieser den Raum hinter dem Objekt nicht messen (Verdeckung). Sind die Verdeckungen statisch, wie für eine statische Kamera bei einem statischen Objekt, so besteht die Gefahr, dass Personen auch über einen größeren Zeitraum teilweise oder vollständig verdeckt sind, was für ein Tracking problematisch sein kann. Beispielsweise könnte sich in der Roboterarbeitszelle aus Abb. 1.2 eine Person für längere Zeit unter den Tisch begeben und wäre in dieser Zeit für die Kameras darüber nicht sichtbar. Durch den Einsatz mehrerer Kameras an verschiedenen Stellen im Überwachungsraum können die Verdeckungen der statischen Objekte teilweise kompensiert werden. Zu diesem Zweck wird das in Abb. 1.2 zu sehende Multi-View-Kamerasystem, bestehend aus in den Raum gerichteten Farbkameras, für das Tracking verwendet. Konkret werden die sieben im Bild mit roten Kreisen markierten Kameras eingesetzt, denen die Nummern 1 bis 8 (mit Ausnahme der 2) zugeordnet sind. Eine ausführlichere Begründung für die Verwendung eines Multi-View-Kamerasystems im Zusammenhang mit Verdeckungen wird in Abschnitt 2.1 gegeben. Informationen zu den dynamischen Verdeckungen des Roboters könnten durch Wissen

von der Robotersteuerung in das Gesamtsystem integriert werden, so wie es bereits in Vorarbeiten umgesetzt wurde (vgl. Abschnitt 2.3). Allerdings ist dieser Aspekt nicht Bestandteil der Untersuchungen. Der Roboter wird als statisches Objekt betrachtet, da der Fokus dieser Arbeit zunächst auf der Handhabung statischer Verdeckungen beim Tracking liegen soll. Konkrete Annahmen zu den betrachteten Objekten werden in Abschnitt 2.4 beschrieben.

Menschliche Bewegungen können sehr vielfältig sein, da sich Personen prinzipiell auf jede Art und Weise bewegen können, die ihre Physis zulässt. Verschiedene Richtungs- und Tempowechsel können ebenso wie lange Ruhezeiten zwischen den Bewegungen auftreten. Einschränkungen können allerdings durch die Objekte im Raum vorgegeben werden. Dahingehend kann die durchschnittliche Bewegungsgeschwindigkeit bei kleinen und engen Räumen auch als eher gering angenommen werden. Abrupte Bewegungs- und Richtungswechsel sind häufiger erwartbar, weil die Objekte bei niedrigen Geschwindigkeiten nur eine geringe Trägheit besitzen. Es ist weiterhin davon auszugehen, dass bei dem gegebenen Überwachungsszenario wiederholt die selben Personen getrackt werden. Dies bietet die Möglichkeit, Wissen zu den Modellparametern, wie z. B. die Größe der Person, vorauszusetzen.

Es ist vorteilhaft, die Charakteristik der Tracking-Umgebung zu kennen und beim Entwurf des Tracking-Verfahrens zu berücksichtigen, da dies in der Regel zu besseren Tracking-Ergebnissen führt.

1.5 Aufgabenstellung

Die Zielstellung dieser Dissertation besteht in der Umsetzung eines Tracking-Verfahrens für die 3D-Lokalisierung von Personen im Überwachungsraum unter Berücksichtigung statischer Verdeckungen. Dies kann als eine grundlegende und notwendige Aufgabe zur Erfassung und Klassifikation von Bewegungsabläufen betrachtet werden im Hinblick auf Bewegungsvorhersagen und Intentionsschätzungen für die Mensch-Roboter-Kooperation.

Gegeben seien

- ein ortsfestes Multi-View-Kamerasystem, das zeitlich synchronisierte Farbbilder aus unterschiedlichen Perspektiven liefert,
- Kalibrierdaten zu allen Kameras (intrinsische und extrinsische Parameter),
- ein Background-Subtraction-Verfahren für die Objektdetektion in den Bildern,
- eine konservative voxelbasierte 3D-Rekonstruktion der Visuellen Hülle von den

dynamischen Objekten im Überwachungsraum nach [Kuhn, 2012], [Kuhn und Henrich, 2009],

- 3D-Modelle aller statischen Objekte im Überwachungsraum (in Weltkoordinaten).

Gesucht ist

- eine tracking-basierte Zustandsschätzung für die 3D-Lokalisierung von Personen zu jedem Zeitpunkt k, welche statische Verdeckungen berücksichtigt.

Das Personen-Tracking erfolgt mit den gegebenen Eingabedaten und wird hinsichtlich seiner Varianten und Parametrisierungen auf der Basis experimenteller Daten untersucht. Die Einbeziehung statischer Objekte mit ihren Verdeckungen in das Verfahren ist hierbei der Hauptuntersuchungsgegenstand und die Neuheit. Experimentell soll ermittelt werden, inwieweit die Schätzungen von Tracking-Verfahren mit Wissen zu Verdeckungen verbessert werden können. Die Grenzen des entwickelten Ansatzes sollen dabei aufgezeigt werden.

1.6 Kapitelübersicht

Dieser Dissertation unterliegt die folgende Kapitelstruktur: In Kapitel 2 wird die Verwendung eines Multi-View-Kamerasystems ausführlich motiviert, bevor das gesamte System mit seinen Softwarekomponenten beschrieben wird. Es erfolgt eine Abgrenzung zu Vorarbeiten sowie eine Beschreibung der getroffenen Annahmen an Objekte. In Kapitel 3 werden neue formale Definitionen und Begrifflichkeiten zu Verdeckungen eingeführt, da der Begriff „Verdeckung“ in der Literatur uneinheitlich und mitunter recht unspezifisch verwendet wird. Der aktuelle Stand der Forschung ist Fokus von Kapitel 4. Zu den Forschungsbereichen des Human Tracking und der Pose Estimation wird ein Überblick gegeben. Der Rechercheschwerpunkt des Kapitels liegt auf Tracking-Verfahren, die Rekonstruktionsdaten als Eingabe verarbeiten. Eine wissenschaftliche Lücke wird entsprechend aufgezeigt. Thema des Kapitels 5 ist die voxelbasierte 3D-Rekonstruktion des Überwachungsraums mit dynamischen und statischen Objekten. Dazu wird ein Überblick über Rekonstruktionsprinzipien gegeben und das Konzept der Visuellen Hülle erläutert. Der Voxelraum sowie der verwendete konservative Rekonstruktionsalgorithmus werden beschrieben und die Bestandteile einer resultierenden 3D-Rekonstruktion besprochen. Anschließend wird eine Modifikation des Rekonstruktionsalgorithmus aufgezeigt, mit der sich verschiedene Voxelzustände erzeugen lassen, die das vorliegende Wissen zu den statischen Objekten und ihren Verdeckungen kodieren.

Diese Voxelzustände werden für das Tracking benötigt. In Kapitel 6 wird zu Beginn der vorgeschlagene Tracking-Ansatz sowie der theoretische Hintergrund des Partikelfilters beschrieben. Weiterhin werden die einzelnen Tracking-Komponenten detailliert beleuchtet, wobei auf den Zustandsraum, das Bewegungsmodell, das Mehrpersonen-Tracking und Datenassoziationsproblem eingegangen wird sowie auf die Track-Initialisierung und Track-Terminierung. Die Integration des Wissens zu Verdeckungen in das Tracking-Verfahren stellt den Hauptuntersuchungsaspekt dieser Dissertation dar und ist Teil der Likelihood-Funktion sowie des Kollisionstests, der bei der Partikelprädiktion eingesetzt werden kann. Die betrachtete Vorgehensweise wird theoretisch analysiert und in Kapitel 7 experimentell untersucht. In Kapitel 8 werden Schlussfolgerungen aus den Untersuchungen dieser Dissertation gezogen, was eine Zusammenfassung der Arbeit, eine Diskussion der betrachteten Vorgehensweise sowie einen daraus resultierenden Ausblick auf mögliche zukünftige Arbeiten beinhaltet.

Kapitel 2

Gesamtsystem

In diesem Kapitel wird das Gesamtsystem erläutert, das zur Realisierung des Tracking-Verfahrens eingesetzt wird. Eine zentrale Komponente davon stellt ein Multi-View-Kamerasystem dar, das aus mehreren Farbkameras besteht. In Abschnitt 2.1 wird begründet, warum die Wahl auf solch eine Sensorik für die Aufgabe des Trackings gefallen ist. Hierbei wird auch auf die 3D-Rekonstruktion der Visuellen Hülle eingegangen und die Problematik von Verdeckungen ausführlicher erläutert. In Abschnitt 2.2 wird anschließend ein Überblick über das gesamte Raumüberwachungssystem mit seinen einzelnen Softwarekomponenten gegeben. Im Zuge dessen werden auch verschiedene Grundlagen vermittelt, die für das Verständnis der vorliegenden Dissertation benötigt werden. In Abschnitt 2.3 wird Bezug auf Vorarbeiten genommen, welche die Grundlage für Teile des Gesamtsystems bilden. Darin erfolgt ebenso eine Abgrenzung dieser Dissertation zu den Vorarbeiten. Als letztes werden in Abschnitt 2.4 Annahmen an die betrachteten Objekte beschrieben.

2.1 3D-Rekonstruktion mit einem Multi-View-Kamerasystem

Zur Gewinnung von Umgebungsdaten eines Roboters ist es möglich, Sensoren direkt an dem Roboter zu platzieren oder an anderen Stellen im Raum. Für Sensoren, die sich am Roboter befinden, wird häufig der Begriff **lokale Sensorik** verwendet, der zum Ausdruck bringt, dass damit allein nur ein begrenzter Ausschnitt der Umgebung erfasst werden kann. Mit einer **globalen Sensorik**, die nicht (alleinig) am Roboter angebracht ist, kann hingegen ein größerer Teil der Umgebung, z. B. ein gesamter Überwachungsraum, eingesehen werden und der Roboter selbst auch als Teil davon. Bei mobilen Plattformen wie Servicerobotern kommen häufig mehrere lokale Sensoren (auch unterschiedlicher Technologie) zum Einsatz, um den Sichtbereich zu vergrößern. Diese liegen sehr nah beieinander und haben zumindest annähernd ein gemeinsames Projektionszentrum. Die Fusion der Sensordaten davon kann auch als ein einzelner, virtueller Sensor mit weitem Rundumblick aufgefasst werden. Im Vergleich dazu werden die einzel-

nen Sensoren einer globalen Sensorik dezentral und in größerem Abstand voneinander installiert. Damit sind Perspektiven wählbar, die eine Betrachtung der Gegenstände und Personen im Raum sowie des Roboters von unterschiedlichen Seiten zulässt. Dies kann im Umgang mit Verdeckungen vorteilhaft sein. Verdeckungen werden verursacht durch nicht transparente Objekte aller Art. Die Transparenz hängt dabei vom Sensorprinzip ab oder einem Mix aus Prinzipien, im Falle es werden unterschiedliche Technologien und deren Fusion betrachtet. Die heutzutage für die Raumüberwachung einsetzbaren Sensoren können bis zur nächsten nicht transparenten Objektoberfläche messen. Dahinterliegende Bereiche sind sensorisch abgeschattet und damit nicht erfassbar. Aktuell verfügbare Sensoren, die materialdurchdringend sind, wie z. B. Röntgenstrahlen oder Radar, sind für die dauerhafte Überwachung derzeit ungeeignet, beispielsweise aufgrund von Gesundheitsgefahren oder deren beschränkter Auflösung.

Die Nutzung einer globalen Sensorik mit dezentral verteilten Einzelsensoren unterschiedlicher Perspektiven bietet das Potential, verdeckte Volumina für das gesamte Kamerasystem zu minimieren. Dadurch ist im besten Fall lediglich das Innere gegebener Objekte (Gegenstände und Personen) sensorisch unzugänglich. Die verdeckte Sicht eines Sensors kann demnach durch die freie Sicht anderer Sensoren kompensiert werden, was in dieser Dissertation als **sensorische Kompensation** bezeichnet wird. Die Positionsoptimierung eines Multi-View-Kamerasystems mit dem Ziel der Minimierung verdeckter Volumina für gegebene statische Objekte im Überwachungsraum war Gegenstand einer vorangegangenen Forschungsarbeit [Haenel, 2015].

Warum sind Verdeckungen überhaupt ein Problem? Für die Vision einer vorausschauenden Bahnplanung des Roboters sind Verdeckungen einschränkend, weil man nur Freiraum bei der Planung berücksichtigen darf, um Kollisionen zu vermeiden. Der Freiraum muss jedoch bei dynamischen, a priori unbekannten Objekten wie Personen ständig sensorisch neu bestimmt werden, sofern kein weiteres Wissen zu den Dynamiken der Umgebung zur Verfügung steht. Je weitgehender man die Roboterbewegungen also optimieren möchte, desto besser muss der Roboter auch den Freiraum erfassen und vorhersagen können. Blockieren Objekte jedoch die Sicht eines Roboters, so dürfen die entstehenden verdeckten Volumina ohne Zusatzwissen nicht in die Bahnplanung einbezogen werden. Andernfalls könnte sich beispielsweise eine Person unerkannt aus einer Verdeckung auf den Roboter zubewegen, was ein hohes Verletzungsrisiko mit sich bringt. Aber nicht nur die Bewegungsplanung von Robotern sondern auch die Bewegungsverfolgung und Bewegungsvorhersage von Personen leidet unter Verdeckungen. Sind Personen teilweise oder vollständig verdeckt, so wird die Zustandsschätzung beim Tracking erschwert. Sie kann aufgrund von Mehrdeutigkeiten zu falschen Ergebnissen führen oder auch vollständig ausfallen, wenn nicht genügend geeignete Merkmale aus der Messung extrahiert werden können. Zusammenfassend ist sowohl für die Bewe-

gungsplanung von Robotern als auch für die Bewegungsverfolgung und -vorhersage von Personen eine globale Sensorik wünschenswert, um Verdeckungen reduzieren zu können.

Zur Raumüberwachung können Sensoren unterschiedlicher Eigenschaften eingesetzt werden, beispielsweise Tiefensensoren wie Laserscanner, Ultraschallsensoren oder Tiefenkameras. Diese führen eine Distanzmessung zu den nächsten Objektoberflächen durch. Tiefenkameras können als 2,5D-Sensoren kategorisiert werden und bestehen ebenso wie andere Tiefensensoren häufig aus einer Sender-Empfänger-Einheit, weshalb sie dann auch als **aktive Sensoren** bezeichnet werden. Mit ihnen kann ein Signal ausgesendet und die Laufzeit des Eintreffens auf den Empfänger nach erfolgter Reflektion im Raum gemessen werden. Alternativ kann über das Aussenden von strukturiertem Licht, z. B. mit der originalen Kinect [Zhang, 2012], mit Hilfe der Methode der Triangulierung die Tiefe einer betrachteten Oberfläche bestimmt werden. Die Triangulierung erfordert einen räumlichen Versatz zwischen Sender und Empfänger. **Passive Sensoren** messen Signale, die von der Umgebung abgegeben werden, wie z. B. Strahlung, die von Objektoberflächen reflektiert oder emittiert wird. Das Ergebnis sind Farb- oder Intensitätssignale im Sensor. Durch das gegebene physikalische Aufnahmeprinzip lässt sich mit einem passiven Sensor ohne eine Weiterverarbeitung der Daten nur ein zweidimensionales Abbild der Umgebung erstellen, da die Tiefe zum Objekt ohne weiteres Wissen nicht aus dem gemessenen Signal folgt. Standardkameras werden demzufolge als 2D-Sensoren kategorisiert.

Zur Realisierung einer globalen Sensorik bestehend aus Einzelsensoren unterschiedlicher Perspektive auf die Szene würde man bei idealisierter Betrachtungsweise bevorzugt 2,5D-Sensoren (Tiefenkameras) einsetzen, um bei ausreichender Anzahl eine fast vollständige 3D-Raumbelegung rekonstruieren zu können. Bei aktiven Tiefenkameras gibt es jedoch verschiedene Nachteile, welche insbesondere die gleichzeitige Verwendung mehrerer Sensoren für eine globale Sensorik erschweren. So können sich die aktiv ausgesendeten Signale gegenseitig beeinflussen (Interferenz) und auch von den falschen Empfängern verarbeitet werden. Zudem sind aktive Sensoren bei bestimmten Oberflächen, z. B. metallischer Art, störanfällig. Weitere einschränkende Kriterien sind eine begrenzte Auflösung und fehlende Weitwinkeligkeit, ein erhöhter Energieverbrauch und nicht zuletzt auch die mitunter sehr hohen Kosten, z. B. von 2,5D-Laserscannern. Aus diesem Grund wurde die Verwendung von Tiefensensoren zur Realisierung einer globalen Sensorik für die gewählte Aufgabenstellung des Personen-Trackings mit Verdeckungen als noch ungeeignet eingestuft.

Bei der Verwendung von passiven Kameras entfällt eine Reihe der negativen Eigenschaften aktiver Sensoren. Vorteilhaft ist, dass sich die Kameras nicht gegenseitig beeinflussen, sie können einander lediglich die Sicht verdecken, wenn sie ungünstig platziert werden.

Es gibt die Möglichkeit, Tiefendaten mit Standardkameras zu berechnen, indem die Daten mehrerer Kameras, die Objekte aus unterschiedlichen Blickwinkeln sehen, miteinander fusioniert werden. Bei **Stereokameras** sind zwei passive Kameras in einem geringen definierten Abstand zueinander angeordnet ($< 20\,\text{cm}$). Mit dem Prinzip der **Korrespondenzanalyse** werden gemeinsame Punkte oder andere Merkmale in den Bildern gesucht, um daraus Tiefeninformationen berechnen zu können. Korrespondierende Punkte oder Merkmale, wie z. B. SIFT-Features, werden über die Methode der Triangulierung in den Raum rückprojiziert, was Strahlen im Raum ergibt. Im Anschluss wird der 3D-Punkt bestimmt, der den quadratischen Fehler (Abstand) zu beiden Strahlen minimiert und daher auf einer Objektoberfläche liegen sollte. Zur Verbesserung der Suche nach Korrespondenzen in den Bildern wird die Epipolargeometrie der Kameras verwendet, was eine entsprechende Kalibrierung erfordert.

Die Einsatzmöglichkeit von Stereokameras hat ihre Grenzen. Liegen homogene Bildinhalte aufgrund einer fehlenden Texturierung oder Strukturierung der Umgebung vor, wie z. B. bei einfarbiger Kleidung und einfarbigem Hintergrund, so kann dies zu fehlerhaften oder mangelnden Korrespondenzen führen. Dann werden zu viele Punkte mit ähnlichen Eigenschaften gefunden, wodurch Mehrdeutigkeiten entstehen. Weiterhin gibt es auch Probleme mit Verdeckungen, besonders an Objekträndern, wobei aufgrund des geringen Versatzes der Kameras oftmals trotzdem noch ausreichend viele Korrespondenzen gefunden werden.

Anders sieht es mit Verdeckungen bei Kameras aus, die einen Versatz zueinander haben, der mitunter deutlich größer ist als bei Stereokameras. In solchen Fällen spricht man auch von **Wide-Baseline-Kameras**. Die Verdeckungsproblematik vergößert sich hierbei, wodurch nicht mehr garantiert werden kann, dass selbst bei guter Texturierung der Objekte ausreichend viele korrespondierende Punkte gesehen werden können. Deshalb ist das Verfahren der Triangulierung hierbei nicht ohne Weiteres einsetzbar. Bei solchen Kameraanordnungen wird häufig ein anderes Rekonstruktionsprinzip eingesetzt, welches auch für das Ziel dieser Dissertation vielversprechend erscheint, namentlich die Rekonstruktion einer **Visuellen Hülle** durch ein **Shape-from-Silhouette**-Verfahren.

Die Rekonstruktion einer Visuellen Hülle ermöglicht eine Fusion der Daten von mehr als zwei Kameras, welche beliebig im Raum platziert werden können, aber annähernd dasselbe Volumen betrachten sollten. Damit ist eine globale Sensorik mit sehr unterschiedlichen Perspektiven realisierbar. Als Ergebnis erhält man eine volumenbasierte Approximation der Raumgeometrien. In [Kuhn und Henrich, 2009] wird die Rekonstruktion einer Visuellen Hülle bei gegebenen Verdeckungen im Raum näher betrachtet und ein Algorithmus entworfen, welcher Wissen zu statischen Objekten im Raum integriert (3D-Modelle der statischen Objekte), wodurch eine obere konservative Abschätzung der

Raumbelegung vorgenommen werden kann. Hieraus entwickelte sich die Frage, ob sich 3D-Rekonstruktionen (in Form von Voxeldaten) als Eingabe für eine tracking-basierte Zustandsschätzung von Personen eignen, um eine gute Personenlokalisierung zu erzielen. Aus oben genannten Gründen der Störanfälligkeit von Stereokameras sowie den bereits existierenden Algorithmen für die Visuelle Hülle bei Verdeckungen, werden in dieser Dissertation C Farbkameras in einer Wide-Baseline-Anordnung verwendet (ein $C\times$2D-Farbkamerasystem) und die Berechnung einer Visuellen Hülle vorgenommen.

Als alternatives Rekonstruktionsprinzip zum Shape-from-Silhouette-Verfahren wird in Abschnitt 9.2 ein Verfahren zur Rekonstruktion einer **Photohülle** mit 2D-Kameras beschrieben, das für die Verwendung in dieser Dissertation untersucht wurde und eine spezielle Art der Korrespondenzanalyse vornimmt. Eine Photohülle wird als einschließende Hülle auf der Basis von Farbkonsistenzen in den Kamerabildern erzeugt, was eine definierte Abarbeitung des Voxelraums mit Sichtbarkeitstests erfordert, genannt **Voxel Carving** oder **Space Carving**. Die Visuelle Hülle dient bei den durchgeführten Betrachtungen als Eingabe zur vorgeschlagenen Rekonstruktion der Photohülle, die ebenfalls als Voxeldatenstruktur abgelegt wird. Vorteile einer Photohülle gegenüber einer Visuellen Hülle ist die im besten Fall fotorealistische Einfärbung der rekonstruierten Volumina sowie eine verbesserte Approximation der Objektgeometrien, wenn eine ausreichende Anzahl an Kameras zum Einsatz kommt. Im Idealfall werden die Pixelfarben der Kamerabilder annähernd an den (virtuellen) 3D-Orten der Rekonstruktion fusioniert, an denen sich in der realen Welt zum Zeitpunkt der Bilderaufnahme auch die zugehörigen Oberflächenpunkte befanden, die zu der Farbentstehung in den Pixeln geführt haben. Damit könnten qualitative 3D-Farbmerkmale für das Tracking gewonnen werden, die sich durch ein nachträgliches automatisiertes Einfärben einer Visuellen Hülle (ohne Sichtbarkeitstests) bisher nicht generieren lassen.

Die Anwendbarkeit von Voxel-Carving-Algorithmen für Überwachungsszenarien erfordert jedoch einen hohen Rechenaufwand, insbesondere aufgrund der notwendigen iterativen Vorgehensweise. Dafür wurden in [Zwicker, 2013] GPU-Algorithmen entworfen, auch für eine konservative Visuelle Hülle wie sie in dieser Dissertation betrachtet wird. Die Ergebnisse wurden auch in [Ober-Gecks et al., 2014b] und [Ober-Gecks et al., 2016] veröffentlicht. In Gänze erwies sich die Photohülle als zu rechenaufwändig im Verhältnis zu dem geringen Nutzen hinsichtlich der Verbesserung der Objektapproximation. Auch war die resultierende Kolorierung durch die im gegebenen Anwendungsszenario vorliegenden Verdeckungen unvollständig, was bei einem Einsatz im Tracking-Verfahren entsprechend behandelt werden muss. Aus genannten Gründen wurden für die Untersuchungen dieser Dissertation Rekonstruktionsdaten mit einem Shape-from-Silhouette-Verfahren erzeugt.

2.2 Softwarekomponenten

Im vorangegangenen Abschnitt wurde für die Aufgabe des Personen-Trackings die Verwendung eines Multi-View-Kamerasystems, bestehend aus $C \times$2D-Farbkameras, sowie die Rekonstruktion einer Visuellen Hülle motiviert. An dieser Stelle soll ein Überblick über das zum Einsatz kommende Gesamtsystem mit seinen Softwarekomponenten gegeben werden, die in Abb. 2.1 blau hinterlegt sind. Daten, die von den Komponenten erzeugt und zwischen ihnen ausgetauscht werden, sind orange eingefärbt. Zu unterscheiden ist zwischen Offline- und Online-Softwarekomponenten. Unter „offline“ ist hier der Einrichtbetrieb zu verstehen, der einen teilweise manuellen Aufwand erfordert. Die Ausgaben der Offline-Softwarekomponenten müssen zu Beginn des Systemstarts vorliegen, da sie als Eingabe für die Online-Softwarekomponenten benötigt werden, welche zyklisch während des Systembetriebs ausgeführt werden. Jeder Zyklus beginnt mit dem Erhalt neuer Szenenbilder und endet mit der Zustandsschätzung des Personen-Trackings (vgl. Abb. 2.1).

Die Durchführung einer intrinsischen und extrinsischen **Kamerakalibrierung** für alle Kameras des Multi-View-Kamerasystems ist unabdingbar. Bei einer intrinsischen Kalibrierung werden die spezifischen Abbildungsparameter bestimmt, welche die realen Abbildungseigenschaften des optischen Systems und mögliche Verzerrungen beschreiben. Diese Parameter werden benötigt, um die Projektion von Raumpunkten auf die Sensorpunkte berechnen zu können. Die Entzerrung der Bilder dient zur Reduktion der Abbildungsberechnungen auf die Transformationsgleichungen eines Lochkameramodells (vgl. Kapitel 3). Die extrinsischen Kameraparameter umfassen die Position und Ausrichtung des optischen Zentrums der Kameras und ermöglichen die Fusion der Bilddaten mehrerer Kameras, so wie es im Schritt der 3D-Rekonstruktion erforderlich ist. Voraussetzung hierbei ist, dass die Kameras statisch montiert sind, sodass insbesondere die extrinsischen Kameraparameter konstant bleiben.

In dieser Dissertation wird das globale Optimierungsverfahren von [Svoboda et al., 2005] zur Kamerakalibrierung eingesetzt, das in Form eines MATLAB-Pakets frei verfügbar ist (vgl. [Svoboda, 2011]). Für die Eingabe des Verfahrens werden Leuchtpunkte an unterschiedlichen Orten innerhalb des abgedunkelten Überwachungsraums jeweils von allen Kameras synchron aufgezeichnet. Mit diesen Daten wird ein lineares überbestimmtes Gleichungssystem aufgestellt. Eine approximative Lösung des Gleichungssystems wird durch einen iterativen Fehlerminimierungsprozess erzeugt. Abschließend wird eine Abbildung (Mapping) der Kalibrierdaten auf ein definiertes Weltkoordinatensystem des Überwachungsraums durchgeführt. Dafür kommt ein Registrierungsobjekt mit bekannten Punkten im Weltkoordinatensystem zum Einsatz. Mit diesem Schritt werden auch

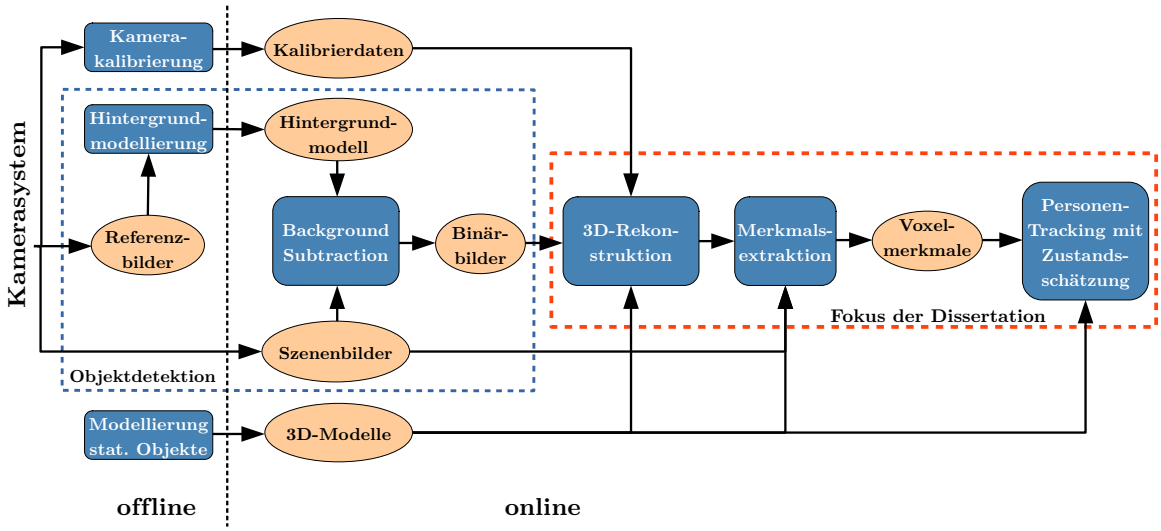
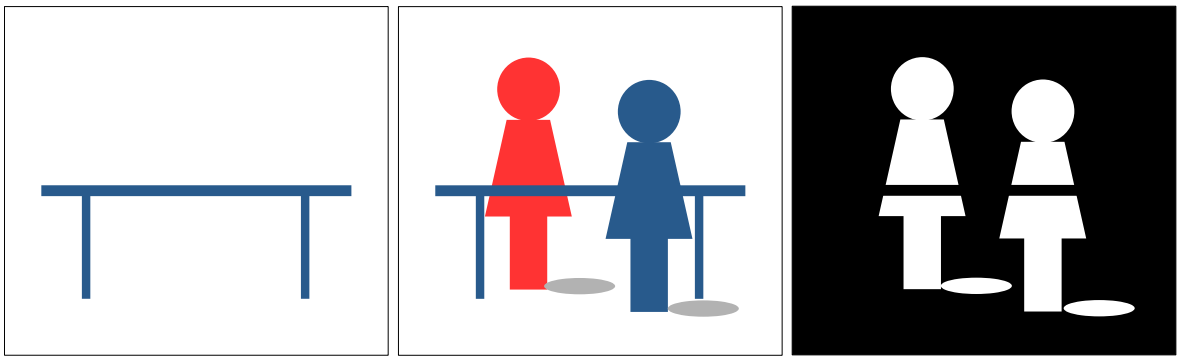


Abb. 2.1: Das Gesamtsystem mit seinen Softwarekomponenten (blau) und Daten (orange).

korrekte Skalierungsparameter für die Kamerapositionen generiert. Die Implementierung oder Verbesserung eines Kalibrierverfahrens ist nicht Teil dieser Dissertation.

Als **Hintergrundmodellierung** wird die nächste betrachtete Offline-Softwarekomponente bezeichnet. Sie wird für die automatische Segmentierung von Personen und anderen dynamischen Objekten in den Kamerabildern benötigt (Objektdetektion). In diesem Schritt wird für jede Kamera ein Hintergrundmodell (engl. Background Model) generiert, welches Wissen zum erwarteten Aussehen der statischen Szene, ohne die Präsenz dynamischer Objekte, speichert. Mit Hintergrund ist demnach die statische Szene ohne Objekte von Interesse gemeint. Zur Erzeugung eines Hintergrundmodells werden üblicherweise ein oder mehrere Referenzbilder von der statischen Szene aufgenommen. Das Hintergrundmodell selbst kann z. B. ein Einzelbild, Mittelwertbild oder ein komplexeres Modell sein, das aus den Referenzbildern generiert wird. Letzteres liegt beispielsweise vor, wenn Künstliche Neuronale Netze eingesetzt werden [Werner et al., 2017]. Auch die bekannten Mixture of Gaussians [Bouwmans et al., 2008] gehören zu den komplexeren Hintergrundmodellen. Sie bieten den Vorteil der Speicherfähigkeit mehrerer Hintergründe, wodurch die Objektdetektion im Online-Betrieb verbessert werden kann, beispielsweise wenn es zu einem Wechsel zwischen erwarteten Beleuchtungssituationen kommt, die im Modell gespeichert sind.

Das Hintergrundmodell wird im Systembetrieb von einem sogenannten **Background Subtraction** (BS) für die Objektdetektion verwendet (synonym auch als „Change Detection“ bezeichnet [Radke et al., 2005]). Das Prinzip dieser Verfahren kann allgemein als die Suche nach Bildbereichen aufgefasst werden, die sich signifikant gegenüber einem oder mehrerer erwarteter Zustände verändert haben. In einem naiven Ansatz werden zwei Bilder einer Kamera, die zu unterschiedlichen Zeitpunkten aufgenommen

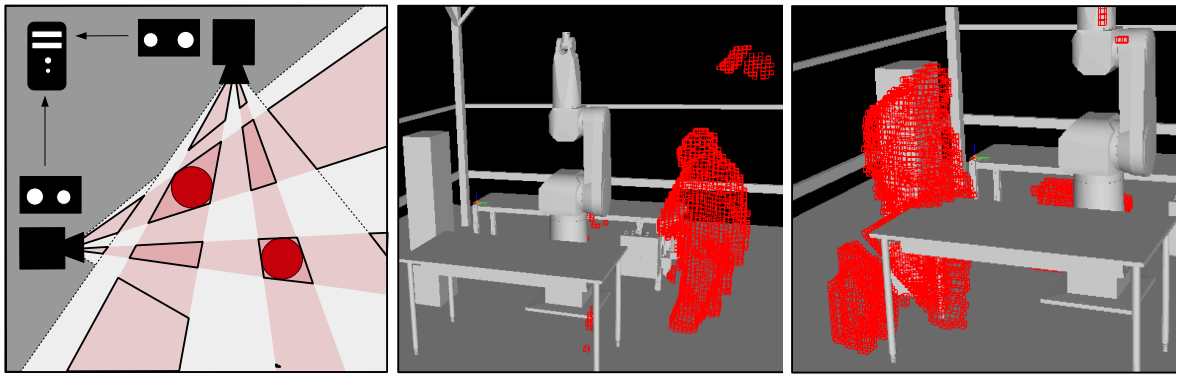


(a) (b) (c)

Abb. 2.2: Background Subtraction. Referenzbild von dem statischen Hintergrund (a). Szenenbild mit zwei Personen, die detektiert werden sollen (b). Binärbild bzw. Silhouettenbild nach der Objektdetektion (c), auch Vordergrund genannt. Neben den Personen wurden auch deren Schatten detektiert, die nicht von Interesse sind (falsch-positiv klassifizierte Pixel). Teile der Personen fehlen im Binärbild (falsch-negativ klassifizierte Pixel).

wurden (hier im Offline- und Online-Betrieb) miteinander verglichen, dargestellt in Abb. 2.2. Dabei wird die pixelweise Differenz beider Bilder berechnet. Anschließend erfolgt eine Binarisierung des resultierenden Differenzbildes: Liegt für ein Pixel der Differenzbetrag über einem definierten Schwellenwert, so wird davon ausgegangen, dass sich der Zustand des Raumbereichs, den dieses Pixel erfasst, signifikant geändert hat. Das Pixel wird als verändert klassifiziert und erhält einen entsprechenden Eintrag im Binärbild (z. B. 1), gezeigt in Abb. 2.2(c) durch die weißen Bereiche. Liegt der Pixelwert hingegen unterhalb des Schwellenwerts, so wird das Pixel als unverändert klassifiziert und bekommt einen anderen Eintrag im Binärbild (z. B. 0). Das binäre Ergebnisbild wird auch Maske („Change Mask“ in [Radke et al., 2005]) oder Silhouettenbild genannt, da es die Silhouetten bzw. Konturen der detektierten dynamischen Objekte zum Vorschein bringt. Zu detektierende Objekte werden auch als Vordergrund (engl. Foreground) bezeichnet.

BS-Verfahren werden aufgrund ihres einfachen Detektionsprinzips sehr häufig eingesetzt, um auf die Anwesenheit unbekannter Objekte im Überwachungsraum zu schließen. Allerdings sind die Verfahren recht fehleranfällig. Beispielsweise können Beleuchtungsänderungen auftreten, die nicht im Hintergrundmodell kodiert sind und dadurch als Vordergrund detektiert werden, sogenannte **falsch-positiv** klassifizierte Pixel. Auch Schatteneffekte gehören zu dieser Kategorie (vgl. Abb. 2.2(c)). Ein „Entfernen“ solch detektierter Schattensegmente kann aber bereits Bestandteil des Background-Subtraction-Verfahrens sein. Auch andere Beleuchtungsänderungen können teilweise kompensiert werden. Ein sicherheitsrelevantes und damit größeres Problem liegt jedoch dann vor, wenn die Annahme einer optischen Unterscheidbarkeit von Vordergrund und Hintergrund nicht zutrifft. Vordergrundobjekte sehen dann (teilweise) so aus wie der Hin-



(a) (b) (c)

Abb. 2.3: Shape-from-Silhouette-Prinzip zur Rekonstruktion einer Visuellen Hülle. Die von segmentierten Pixeln stammenden rückprojizierten Sichtkegel werden im 3D-Raum miteinander verschnitten, wodurch Volumina entstehen, welche die Objekte einschließen (a). In (b) und (c) sind Rekonstruktionsergebnisse in Form von Voxeldaten dargestellt (rot). Die statischen Objekte (grau) sind modelliert und nicht Teil der Rekonstruktion.

tergrund. In Abb. 2.2 hat die blaue Person die gleiche Farbe wie der Tisch, deshalb kann ein Teil der Person nicht detektiert werden. Dies resultiert in **falsch-negativ** klassifizierten Pixeln, wie in Abb. 2.2(c) zu sehen ist. Solche Fälle stellen eine Herausforderung dar. Bei dem Einsatz von BS-Verfahren in sicherheitskritischen Anwendungsszenarien, z. B. in der Umgebung schneller und schwerer Industrieroboter, muss sichergestellt werden, dass ein Auftreten falsch-negativer Pixel verhindert oder kompensiert wird. Dies kann beispielsweise durch eine Festlegung des Hintergrunds, der Beleuchtung sowie des Aussehens der zu detektierenden Objekte erreicht werden. Es gibt inzwischen unzählig viele verschiedene BS-Verfahren. Bekannte Übersichtspublikationen dazu sind die Arbeiten von [Radke et al., 2005], [Elhabian et al., 2008] und [Bouwmans, 2014]. Der Fokus dieser Dissertation liegt nicht auf der Entwicklung eines robusten BS-Verfahrens. In den Experimenten zum Personen-Tracking wird ein Codebook-Algorithmus aus der OpenCV-Bibliothek verwendet [Kim et al., 2004], [Pavlenko, 2012].

Nach jedem Online-Schritt der Binärbildberechnung für alle Kameras wird mit einem Verfahren der **3D-Rekonstruktion** der Überwachungsraum rekonstruiert. Hierfür werden auch die Daten der Kamerakalibrierung benötigt. Bei dem Rekonstruktionsprinzip der **Visuellen Hülle** werden die detektierten Silhouetten der dynamischen Objekte in den 3D-Raum rückprojiziert und anschließend miteinander verschnitten. Dieses Prinzip ist in Abb. 2.3(a) dargestellt. Durch die unterschiedlichen Perspektiven des Multi-View-Kamerassystems, aus denen die Objekte und Personen betrachtet werden, entsteht als Ergebnis des Volumenverschnitts eine geometrische Approximation der detektierten Objekte. Eine Visuelle Hülle, die in dieser Dissertation in Form von Voxeldaten gespeichert wird, umschließt die Objekte (vgl. Abb. 2.3(b) und (c)). Das Umschließen ist jedoch

dann unvollständig, wenn die Objektsilhouetten in den Binärbildern unvollständig sind. Dies kann neben dem oben dargestellten Fall falsch-negativ klassifizierter Pixel auch dann auftreten, wenn sich Vordergrundobjekte aus Sicht der Kameras hinter die statischen Objekte der Szene begeben und dadurch teilweise oder vollständig für eine oder mehrere Kameras verdeckt sind – so wie ein Teil der roten Person in Abb. 2.2(b) verdeckt ist und dieser Teil nicht detektiert werden kann (vgl. Abb. 2.2(c)).

Zur Lösung dieses Problems wird der Ansatz von [Kuhn, 2012], [Kuhn und Henrich, 2009] angewandt. Dafür wird offline von allen statischen Objekten des Überwachungsraums in einem Schritt der **Modellierung statischer Objekte** ein 3D-Modell erzeugt. Basierend auf den 3D-Modellen können synthetische Tiefenbilder vom statischen Überwachungsraum generiert werden. Für jede Kamera ist damit bekannt wie weit sie ohne weitere unbekannte Objekte pro Pixel schauen kann. Die Integration dieser Tiefeninformationen in den Rekonstruktionsprozess ermöglicht die Berücksichtigung verdeckter Raumbereiche bei der Verschneidung der rückprojizierten Silhouetten. Dadurch kann eine konservative Visuelle Hülle erzeugt werden, die eine obere Abschätzung des von dynamischen Objekten belegten Raums darstellt. Voraussetzung dafür ist, dass die für die Kameras unverdeckten Objektteile auch richtig detektiert werden. Die Modellierung statischer Objekte kann auf unterschiedliche Art und Weise erfolgen, z. B. durch eine sensorbasierte 3D-Rekonstruktion mit einer Tiefenkamera [Sand, 2019]. In dieser Dissertation werden Dreiecksnetze von den statischen Objekten der SIMERO-Roboterarbeitszelle (vgl. Abb. 1.2) verwendet, die in der Softwareumgebung Blender erstellt wurden [Stoychev, 2013]. Das Robotermodell selbst wird vom Hersteller Stäubli in Form von CAD-Daten zur Verfügung gestellt.

Nach erfolgter 3D-Rekonstruktion wird im nächsten Online-Verarbeitungsschritt eine **Merkmalsextraktion** vorgenommen. Es werden Merkmale aus den Voxeldaten generiert, die in der Likelihood-Funktion des Tracking-Verfahrens ausgewertet werden. Prinzipiell können verschiedenste Merkmale aus den Bildern oder der Rekonstruktion gewonnen werden, die sich auf die Geometrien, das Aussehen wie Farbe und Textur oder sonstige Eigenschaften der Objekte beziehen. Wichtig ist, dass sich diese zur Auffindung der zu trackenden Objekte in den Daten eignen und im besten Falle auch eine Unterscheidung verschiedener Objekte voneinander ermöglichen, um eine gute Zustandsschätzung zu realisieren. In dieser Dissertation werden Voxelzustände verwendet, die sich auf die Raumbelegung oder die Verdeckungssituation beziehen [Ober-Gecks et al., 2014a]. Die Bestimmung der Voxelzustände basiert auf den modellierten statischen Objekten und wird in Abschnitt 5.7 beschrieben. Auch für die Vorhersage im Prädiktionsschritt des Trackings werden Modelle der statischen Objekte eingesetzt. Das **Tracking** ist der finale Schritt in der Verarbeitungskette zur Verfolgung von Personen im Überwachungsraum (vgl. Abb. 2.1), wofür ein Partikelfilteransatz Anwendung findet. Die Theorie zum Par-

tikelfilter und der implementierte Algorithmus werden in Abschnitt 6.2 erläutert. Die Beschreibung des gesamten umgesetzten Tracking-Verfahrens findet sich in Kapitel 6.

2.3 Vorarbeiten des SIMERO-Projekts

Das im vorangegangenen Abschnitt dargestellte Gesamtsystem zur Raumüberwachung basiert zum Teil auf Vorarbeiten, die im Rahmen des SIMERO-Projekts entstanden sind [Henrich, 2021]. Das SIMERO-Projekt zielt auf die Entwicklung von Sicherheitsstrategien für die Mensch-Roboter-Koexistenz und -Kooperation ab. Eine Kernfunktionalität ist dabei die Entwicklung einer Sicherheitsfunktion zur Kollisionsvermeidung, damit sich Personen in der Arbeitszelle eines Roboters frei bewegen können, ohne ein Verletzungsrisiko einzugehen. Hierfür müssen die Personen und auch andere unbekannte Objekte zu jedem Zeitpunkt in der Arbeitszelle sicher lokalisiert werden können. Für diese Aufgabe wird ein stationäres Multi-View-Kamerasystem untersucht, welches die dynamischen Objekte in den Farbbildern detektiert und Rückschlüsse auf deren 3D-Lokalisierung ermöglicht. Der Roboter kann basierend auf diesen Informationen seine Geschwindigkeit adaptiv regeln und online geplante Ausweichbewegungen bei der Verfolgung seiner eigenen Ziele vornehmen, ohne Kollisionen zu verursachen (sensorgestützte Bahnplanung).

Die ersten Berechnungen zur Kollisionsvermeidung im SIMERO-Projekt erfolgten in der 2D-Ebene der Kamerabilder. Bei dem Ansatz wird ein Robotermodell, das zur Laufzeit über die Robotersteuerung konfiguriert wird, in die Kamerabilder projiziert und auf Schnitt mit den detektierten Objektsilhouetten geprüft [Ebert und Henrich, 2002]. Ein weiterführender Ansatz besteht darin, eine Rückprojektion detektierter Silhouettenpixel in den 3D-Raum als Approximation der Raumbelegungen vorzunehmen, was auch als Sichtkegel bezeichnet wird [Kuhn et al., 2006]. Damit können Abstände im 3D-Raum zum konfigurierten Robotermodell berechnet werden, was implizit eine genauere Lokalisierung unbekannter Objekte im Raum ermöglicht als die rein bildbasierte Vorgehensweise. Eine weitere Verbesserung der Lokalisierung kann durch die Rekonstruktion einer Visuellen Hülle erreicht werden, wenn die rückprojizierten Silhouettenbilder aller Kameras im 3D-Raum miteinander verschnitten werden [Kuhn, 2012]. Die Approximation der Raumbelegungen ist dabei genauer. Parallel zu dieser Dissertation entstand ein Ansatz bei dem statische stückweise planare Objekte offline als Boundary-Repräsentation (B-rep) aus Sensordaten modelliert und mit einer Online-Rekonstruktion der dynamischen Objekte kombiniert werden, die in Form einer binären Octree-Datenstruktur erzeugt wird [Werner et al., 2018]. Die Rekonstruktionspräzision konnte in diesem Ansatz im Vergleich zu den genannten Vorarbeiten weiter verbessert

und die Ausführungszeiten der Kollisionstests des Bahnplaners reduziert werden. Die binäre Octree-Datenstruktur aus [Werner und Henrich, 2014] lässt sich überdies in eine Octree-Repräsentation überführen, die ein distanzbasiertes Kollisionsrisiko mit statischen und dynamischen Objekten der Arbeitszelle kodiert und sich für die Durchführung effizienter Kollisionstests einer Online-Bahnplanung eignet [Werner et al., 2019].

Bezüglich der Entwicklung von Algorithmen für die Bestimmung von Raumbelegungen zur Objektlokalisierung als Grundlage für die online durchgeführte Geschwindigkeitsregulierung und Bahnplanung standen und stehen insbesondere die folgenden Aspekte im Vordergrund:

- Umgang mit verdeckten Raumbereichen bei der Abschätzung von Raumbelegungen für die Sicherheit und Genauigkeit,
- Plausibilisierungen zur Entfernung von Fehldetektionen,
- Konservativität der Algorithmen für die Sicherheit,
- Laufzeitoptimierungen für die Online-Anwendbarkeit (Echtzeitfähigkeit).

Diese Aspekte werden im Folgenden mit Blick auf die Vorarbeiten genauer erläutert.

2.3.1 Verdeckte Raumbereiche

Zentraler Untersuchungsaspekt für das industrielle Anwendungsszenario des SIMERO-Projekts ist der Umgang mit sensorisch nicht einsehbaren Volumina, genannt Verdeckungen, die durch Objekte im Raum entstehen. Für das betrachtete Gesamtsystem stellen insbesondere Verdeckungen ein Problem dar, die von statischen Objekten verursacht werden, da sie nicht Teil der Messungen sind. Statische Objekte werden von einem Background-Subtraction-Verfahren nicht detektiert, wenn sie sich bereits zum Zeitpunkt der Referenzbilderstellung im Überwachungsraum befanden und Teil des Hintergrundmodells sind. Ausnahmen stellen Verletzungen der Annahmen an die Objektdetektion wie abrupte Beleuchtungsänderungen dar. Meistens wäre es zu aufwändig oder nicht möglich, den Überwachungsraum leer zu räumen, damit die darin enthaltenen statischen Objekte nicht zum Hintergrundmodell gehören und folglich beim Background Subtraction in den Silhouettenbildern enthalten sind. In den häufigsten Fällen sollen statische Objekte jedoch auch nicht detektiert werden, da sie für die Anwendung nicht von Interesse sind.

Raumbereiche, die von statischen Objekten verdeckt und sensorisch nicht erfasst werden, können allerdings ein Risiko darstellen, insbesondere wenn sie recht groß und initial,

d. h. zum Systemstart, leer sind. Fährt der Roboter in solch eine Verdeckung hinein, so besteht eine Kollisionsgefahr, wenn sich unerkannt ein Objekt darin befindet oder sich ein dynamisches Objekt plötzlich aus der Verdeckung auf den Roboter zubewegt, ohne dass dieser rechtzeitig darauf reagieren kann.

Aus diesem Grund werden Zusatzinformationen zu den statischen Objekten mit ihren Verdeckungen benötigt, die in das System integriert werden können. Bei den bildbasierten Verfahren zur Kollisionserkennung erfolgt dafür eine Segmentierung der verdeckten Bereiche durch manuelles Maskieren der verursachenden statischen Objekte in den Kamerabildern. Auch der dynamische Roboter wird maskiert, was zur Laufzeit durch die Projektion eines Robotermodells in die Kameras geschieht, welches durch die Informationen der Robotersteuerung zu jedem Zeitpunkt parametrisiert wird. Durch das Hinzufügen aller Masken der statischen Objekte und des Roboters zur Laufzeit zu den Silhouettenbildern des BS-Verfahrens werden diese Objekte und die Verdeckungen, die sie erzeugen in bildbasierten Kollisionstests korrekt berücksichtigt. Im Falle einer Abstandsberechnung im 3D-Raum können dazu sämtliche Raumbereiche, die sich aus der Rückprojektion der maskierten Pixel ergeben als belegt angenommen und ebenfalls zur Kollisionserkennung eingesetzt werden. Beide Vorgehensweisen sind jedoch recht limitierend, wenn alle Masken zu jedem Zeitpunkt vollständig einbezogen werden und damit eine stark konservative Abschätzung der Raumbelegungen vorgenommen wird. Im Effekt schränkt dies die Bewegungsmöglichkeiten des Roboters deutlich ein durch erkannte Kollisionen, die im 3D-Raum und in der Realität häufig nicht vorliegen.

Aus diesen Betrachtungen heraus wurde für die SIMERO-Anwendung der sogenannte „θ-Parameter“ eingeführt [Ebert und Henrich, 2001] und weiterentwickelt [Gecks, 2011]. Dieser gibt an, in wie vielen Kameras eine Person maximal vollständig verdeckt sein kann. Dahinter steckt umgekehrt die Annahme, dass zu jedem Zeitpunkt im gesamten Überwachungsraum ein jedes unbekanntes Objekt von mindestens $C - \theta$ Kameras, jeweils zumindest teilweise, detektiert wird. Die Einbeziehung der sich aus den Masken ergebenden verdeckten Raumbereiche in die Abstands- und Kollisionsberechnungen muss unter dieser Annahme nur dann erfolgen, wenn im Bild eine Berührung zwischen einem detektierten Objektsegment und einer manuell erstellten Maske auftritt. Solange dies nicht der Fall ist, kann der Roboter seine Bewegungen auch in verdeckten Volumina ausführen, sofern sie zugänglich sind. Er läuft dann keine Gefahr, eine Kollision zu verursachen. Im Rahmen des „Safety-Vision“-Projekts am Lehrstuhl AI III (2008–2011) wurde die Integration einer Redundanz bei den Abstandsberechnungen unter Verwendung des θ-Parameters untersucht, um bei dem Ausfall einzelner Kameras zur Laufzeit einem Sicherheitsverlust vorzubeugen [Ober und Henrich, 2010].

Bei der Erzeugung einer Visuellen Hülle können die manuell generierten Masken stati-

scher Objekte auf die gleiche Art im 3D-Raum miteinander verschnitten werden wie die Silhouettenbilder des BS-Verfahrens [Ladikos et al., 2008]. In diesem Kontext wird auch von **Occlusion Masks** gesprochen. Die verdeckten Volumina sind damit allerdings Bestandteil der Rekonstruktion. In [Kuhn und Henrich, 2009] wird als Alternative dazu ein Verfahren beschrieben, welches keine Bildmasken verwendet, sondern synthetische Tiefenbilder von 3D-Modellen der statischen Objekte. Dadurch wird eine bessere Approximierung der verdeckten Raumbereiche ermöglicht. Diese Vorgehensweise wird auch in dieser Dissertation verfolgt und ist ausführlich in Abschnitt 5.3.3 beschrieben. Auch für Rekonstruktionen ist die Anwendung des θ-Parameters möglich, um verdeckte Volumina nur dann in den Kollisionstest einbeziehen zu müssen, wenn sich auch tatsächlich unbekannte Objekte darin befinden [Kuhn, 2012]. Dadurch erhält der Roboter mehr Bewegungsfreiheiten.

Nachteilig an der Verwendung des θ-Parameters ist jedoch, dass die Vereinigung aller bezüglich θ relevanten (teil-)verdeckten Volumina im Überwachungsraum zu keinem Zeitpunkt so groß sein darf, dass die zu detektierenden Objekte sensorisch vollständig darin „verschwinden“ können. In diesem Fall könnten sie von der Kollisionsdetektion übersehen werden. Diese implizite Annahme an ein begrenztes Volumen der Verdeckungen und ein Mindestvolumen der Objekte ist einschränkend und muss sichergestellt werden. Zur Anwendung des θ-Parameters ist weiterhin erforderlich, dass das BS-Verfahren möglichst keine falsch-negativ klassifizierten Pixel liefert, da andernfalls ein Sicherheitsproblem (eine Kollisionsgefahr) unerkannt entstehen kann, weil die Bedingung des θ-Parameters verletzt wird. In dieser Dissertation wird der θ-Parameter nicht betrachtet, da das Ziel des Trackings darin besteht, Personen im Überwachungsraum zu verfolgen, wodurch im besten Fall auch erkannt wird, wann Personen in sensorisch verdeckte Bereiche ein- und aus diesen wieder heraustreten. Damit sollte zumindest theoretisch eine Annahme an die Größe verdeckter Volumina im Überwachungsraum obsolet sein.

2.3.2 Plausibilisierungen

3D-Rekonstruktionen aus Kameradaten weisen bekanntermaßen Fehler auf. Bei der Visuellen Hülle ist aufgrund der praktisch begrenzten Anzahl an Kameras im Überwachungsraum von einer begrenzten geometrischen Approximationsgüte der Objekte in der Szene, insbesondere bei nicht konvexen Formen, auszugehen. Zusätzlich entstehen leere Volumina in der Rekonstruktion. Dieser Effekt kommt durch das Rekonstruktionsprinzip der Volumenverschneidung rückprojizierter Sichtkegel zustande.

In [Kuhn, 2012] werden die Rekonstruktionsdaten heuristisch durch sogenannte **Plausibilisierungen** bereinigt, mit dem Ziel die Lokalisierung dynamischer Objekte auf

die wahrscheinlichen Raumbereiche einzugrenzen und dadurch die Approximation der Raumbelegungen zu verbessern. Solche Plausibilisierungen bestehen beispielsweise aus Annahmen an die Mindestausmaße oder den Aufenthaltsort einer Person. Auch die Anwendung des θ-Parameters wurde als Plausibilisierung formuliert.

Die Plausibilisierungen bringen jedoch eine Reihe von Einschränkungen mit sich, da bestimmte Annahmen sichergestellt werden müssen. Die Plausibilisierungen könnten auch in dieser Dissertation angewendet werden und zu besseren Tracking-Ergebnissen führen. Das Tracking wird jedoch eher als alternative Vorgehensweise angesehen, weshalb unbereinigte Rekonstruktionsdaten verwendet werden.

2.3.3 Konservativität

Ein weiterer Fokus des SIMERO-Projekts liegt auf der Entwicklung konservativer Algorithmen, um den Sicherheitsansprüchen eines potentiellen industriellen Einsatzes gerecht zu werden. In [Kuhn, 2012], [Ober und Henrich, 2010] wurde Wert auf die konservative Verarbeitung der Daten gelegt, mit dem Ziel stets eine obere Abschätzung der Raumbelegungen zu erhalten. Die Betrachtungen zur 3D-Rekonstruktion und zu den berechneten Voxelzuständen für das Tracking (vgl. Kapitel 5) erfolgen fortsetzend ebenfalls konservativ.

2.3.4 Laufzeitoptimierung

Für die Entwicklung eines Demonstrators im SIMERO-Projekt war die Echtzeitfähigkeit des Gesamtsystems ein hartes Kriterium. Dabei entstand ein online-fähiger Demonstrator mit bildbasiertem Kollisionstest. Der Roboter konnte eine Geschwindigkeitsregulierung sowie die Planung von Ausweichbewegungen basierend auf den Kameradaten mit einer Geschwindigkeit von ca. 10 fps vornehmen [Gecks, 2011]. Die Anwendung laufzeitoptimierter 3D-Rekonstruktionsalgorithmen einer Visuellen Hülle wird derzeit weiter untersucht. Ein Beispiel ist der Octree-Ansatz, der hochauflösende Kameradaten verarbeitet [Werner und Henrich, 2014], [Werner et al., 2018]. Wie bereits erwähnt, wurde als Alternative zur Visuellen Hülle die Rekonstruktion einer Photohülle mittels Voxel-Carving-Algorithmen untersucht, was die Entwicklung eines beschleunigten Algorithmus erforderte, um die Anwendbarkeit einer Photohülle für ein Überwachungsszenario mit Verdeckungen zu ermöglichen, da Voxel-Carving-Algorithmen sehr rechenintensiv sind (vgl. [Zwicker, 2013] und [Ober-Gecks et al., 2016]). Die dabei entstandenen GPU-Algorithmen einer Photohülle und auch einer Visuellen Hülle werden in Abschnitt 9.2.3 ausführlich dargestellt.

Bei den Implementierungen zum Personen-Tracking wurde allerdings auf eine Laufzeitoptimierung verzichtet. Der zentrale Untersuchungsaspekt wurde auf die qualitativen Tracking-Ergebnisse bei bestehenden Verdeckungssituationen gelegt. Die in Abschnitt 7.1 erwähnten Laufzeitergebnisse sind in einem Rahmen, der mit möglichen Optimierungen Echtzeitfähigkeit erwarten lässt, wodurch auch eine Online-Anwendung umgesetzt werden könnte. In dieser Dissertation werden aufgezeichnete Videosequenzen des vorgestellten Multi-View-Kamerasystems offline für die experimentellen Untersuchungen zum Tracking eingesetzt.

2.4 Annahmen an Objekte

In der Verarbeitungskette des Gesamtsystems (vgl. Abb. 2.1) ist die Objektdetektion mittels Background Subtraction von fundamentaler Bedeutung für die nachfolgenden Ergebnisse der 3D-Rekonstruktion und dem sich anschließenden Tracking. Da der Fokus dieser Dissertation auf dem Umgang mit Verdeckungen liegt und eine gestörte Objektdetektion die Ergebnisse beeinflussen und Aussagen verfälschen könnte, werden Annahmen getroffen und Maßnahmen ergriffen, die zu weniger falsch-positiv und falsch-negativ klassifizierten Pixeln im Binärbild führen. Es werden daher Bildersequenzen von uniform und farbig gekleideten Personen verwendet, mit dem Ziel eine bessere Unterscheidbarkeit von Vordergrund und Hintergrund zu erreichen. Die Beleuchtungssituation ist für eine gesamte Bildersequenz nahezu konstant. Die statischen Objekte haben eine konstante Oberflächenfärbung. Sie sind also z. B. keine Monitore oder Signalleuchten. Dadurch sind die Anforderungen an das Background Subtraction weniger hoch und es können gute Segmentierungsergebnisse mit verschiedenen Verfahren erzielt werden wie mit dem eingesetzten Codebook-Verfahren der OpenCV [Pavlenko, 2012].

Damit die detektierten Segmente in den Silhouettenbildern weitestgehend zu Personen gehören, wird angenommen: Es existieren nur zwei Arten von Objekten, namentlich **statische Objekte** und **dynamische Objekte**, die die Menge aller **Objekte** bilden.

2.4.1 Statische Objekte

- Statische Objekte sind ausschließlich Gegenstände, die ihrer Definition nach nicht lebendig sind.
- Statische Objekte werden nicht bewegt. Sie sind konstant in ihrer Position, Lage und Orientierung.

- Die statischen Objekte befinden sich bereits vor dem Beginn des Trackings zu einem Zeitpunkt k_0 im Überwachungsraum.

2.4.2 Dynamische Objekte

- Dynamische Objekte sind ausschließlich Personen.
- Dynamische Objekte sind entsprechend der Eigenschaften von Personen frei beweglich, haben viele Freiheitsgrade und können deshalb komplexe Bewegungen ausführen.
- Dynamische Objekte dürfen erst nach dem Einrichtbetrieb ab einem Zeitpunkt $> k_0$ den Überwachungsraum betreten.

Synonym werden statische Objekte auch als **Hintergrundobjekte** und dynamische Objekte als **Vordergrundobjekte** bezeichnet, um den Bezug zum Background Subtraction herzustellen. Nur Vordergrundobjekte werden detektiert.

Bevor nachfolgend in Kapitel 4 zum aktuellen Stand der Forschung die Entscheidung für eine konkrete Tracking-Verfahrensklasse motiviert wird, erfolgt in Kapitel 3 die Einführung von Definitionen und Begrifflichkeiten zur Beschreibung von Verdeckungen. Dabei wird nur allgemein von Objekten gesprochen und nicht zwischen statischen und dynamischen Objekten unterschieden, da die Definitionen allgemein für unterschiedliche Anwendungen gültig sein sollen. Zu erwähnen ist weiterhin, dass für die Betrachtungen dieser Dissertation sämtliche Objekte im Inneren als vollständig belegt angenommen werden, um den Formalismus zu vereinfachen. Das Innere von abgeschlossenen Objekten wird sensorisch nicht erfasst.

2.5 Zusammenfassung

In diesem Kapitel wurden die Vorteile der Verwendung eines Multi-View-Kamerasystems dargelegt im Hinblick auf das Anwendungsszenario, welches die Präsenz statischer Objekte im Überwachungsraum mit sich bringt, die zu Verdeckungen der dynamischen Objekte führen können. Die Betrachtung des Überwachungsraums aus unterschiedlichen Perspektiven ermöglicht dabei eine Reduktion der Verdeckungen für das Gesamtsystem durch eine sensorische Kompensation. Die Fusion der Einzelbilder aller Kameras erfolgt zu jedem betrachteten Zeitschritt in Form einer rekonstruierten Visuellen Hülle. Dadurch können die Formen der dynamischen Objekte im Raum unter Berücksichtigung der statischen Verdeckungen geometrisch approximiert werden. Die

Softwarekomponenten des eingesetzten Gesamtsystems wurden erläutert, namentlich die Kamerakalibrierung, die Hintergrundmodellierung und das Background Subtraction sowie die 3D-Objektmodellierung der statischen Objekte. Weiterhin erfolgte eine Beschreibung der Komponenten für die 3D-Rekonstruktion der Visuellen Hülle, die Merkmalsextraktion sowie das Personen-Tracking. Anschließend wurde auf Vorarbeiten des SIMERO-Projekts eingegangen, in denen große Beiträge bezüglich der verwendeten Hardware und Software geleistet wurden. Eine Abgrenzung der Aufgabenstellung zu den Vorarbeiten bezüglich der Kriterien „Verdeckte Raumbereiche“, „Plausibilisierungen“, „Laufzeitoptimierung“ und „Konservativität“ war ebenfalls Bestandteil der Betrachtungen. Zuletzt wurden Annahmen an Objekte getroffen, nach denen bezugnehmend auf das Background Subtraction zwischen statischen Objekten (Hintergrundobjekte) sowie dynamischen Objekten (Vordergrundobjekte) unterschieden wird. Letztere sind mit den zu verfolgenden Personen gleichzusetzen.

Kapitel 3

Verdeckungen

Anwendungsabhängig wird der Begriff **Verdeckung** sehr unterschiedlich verwendet. Deshalb werden in den folgenden Abschnitten verschiedene Begriffe für das Verständnis von Verdeckungen definiert, die unterschiedliche Bedeutungen haben. Allgemein wird in dieser Arbeit als Verdeckung eine Menge von Raumpunkten bezeichnet, die sensorisch nicht erfassbar ist. Die für einen Sensor verdeckten Raumpunkte hängen von den intrinsischen und extrinsischen Sensorparametern ab, wie beispielsweise der Position und Orientierung im Raum. Die Verwendung einer globalen Sensorik bestehend aus mehreren Einzelsensoren ermöglicht es, Raumpunkte, die für einen Sensor verdeckt sind, durch andere Sensoren zu erfassen. Dies wird im Folgenden als das Prinzip der **sensorischen Kompensation** bezeichnet.

3.1 Überwachungsraum und Kamerasystem

Zur formalen Beschreibung von Verdeckungen wird zunächst von einem sensorisch zu überwachenden Raum $R \subset \mathbb{R}^3$ ausgegangen, dem **Überwachungsraum**, wie in Abb. 3.1 zu sehen. R sei gegeben im Weltkoordinatensystem (WKS) mit den Koordinatenachsen x, y, z. Innerhalb dieses Raums können sich Objekte befinden. Alle Raumpunkte, die Teil der Objekte sind, gelten als **belegt** und gehören zur Menge $R_{\text{filled}} \subset R$. Alle anderen Punkte sind leer, werden als **frei** bezeichnet und bilden die Menge $R_{\text{free}} := R \setminus R_{\text{filled}}$. Es gilt $R_{\text{filled}} \dot{\cup} R_{\text{free}} = R$. Sei O die Menge der Objekte o, dann ist o gegeben durch die Menge an Raumpunkten $R_{\text{o}} \subset R_{\text{filled}}$, wobei R_{o} zusammenhängend und abgeschlossen ist. Demnach ist es nicht möglich, ein Objekt durch zwei disjunkte, nichtleere, offene oder abgeschlossene Teilmengen zu repräsentieren. Jeder belegte Raumpunkt hat eine Objektzugehörigkeit und kann nur zu genau einem Objekt $o \in O$ gehören. Umgekehrt enthält jedes Objekt mindestens einen belegten Raumpunkt. Es gilt: $\dot{\bigcup}_{o \in O} R_{\text{o}} = R_{\text{filled}}$. Die Menge der belegten Raumpunkte lässt sich weiter unterteilen in eine Menge **innerer** Objektpunkte $R^{\circ}_{\text{filled}}$ sowie eine Menge **äußerer** Objektpunkte $\partial R_{\text{filled}}$. Hierfür gilt: $\dot{\bigcup}_{o \in O} \partial R_{\text{o}} = \partial R_{\text{filled}}$ sowie $\dot{\bigcup}_{o \in O} R^{\circ}_{\text{o}} = R^{\circ}_{\text{filled}}$.

Zur Überwachung des Raums wird ein **Kamerasystem** bestehend aus $C := \{c_1, \ldots, c_{|C|}\}$ einer endlichen Menge von $|C| \in \mathbb{N}$ kalibrierten und synchroni-

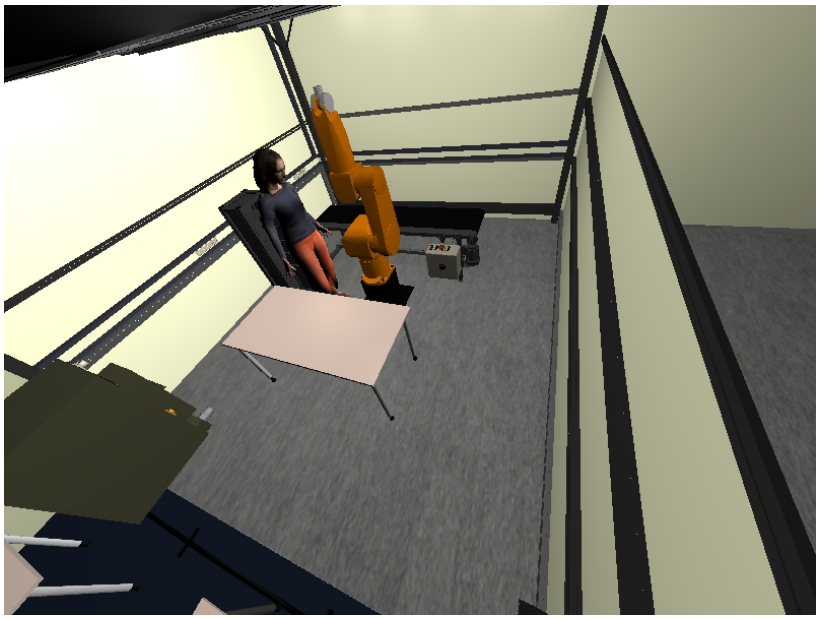


Abb. 3.1: Visualisierung eines Überwachungsraums (SIMERO-Roboterarbeitszelle)

sierten Kamerasensoren eingesetzt. Die Abbildung eines Raumpunkts $r \in R$ auf einen Punkt $\hat{r}_{c_j}$, der in der Bildebene einer Kamera c_j liegt, kann vereinfacht mit einem Lochkameramodell beschrieben werden (vgl. [Jähne, 2005] und [Wikipedia, 2021]). Das Abbildungsprinzip einer Lochkamera, Zentralprojektion genannt, wird in Abb. 3.2(a) dargestellt. Es wird davon ausgegangen, dass von jedem für die Kamera sichtbaren Raumpunkt r ein reflektierender Lichtstrahl durch ein kleines Loch, dem Projektionszentrum, auf einen Punkt der Bildebene trifft. Durch das Auftreffen einer Menge solcher Strahlen, die von unendlich vielen sichtbaren reflektierenden Raumpunkten ausgehen, entsteht ein zweidimensionales, gespiegeltes Abbild der Welt auf der Bildebene. Dieses Abbild wird umso schärfer, je kleiner das Loch der Kamera ist. In der Computergrafik wird beispielsweise für Kamerakalibrierungen oder Raycasting-Verfahren ein idealisiertes Kameramodell verwendet. Dieses besitzt ein unendlich kleines Loch und erzeugt damit keine Tiefenunschärfe, wodurch alle Objekte scharf abgebildet werden. Zudem befindet sich die Abbildung auf einer virtuellen Bildebene, die sich vor dem Projektionszentrum befindet, wodurch keine Spiegelung des Objekts erfolgt (vgl. Abb. 3.2(b)).

Zur formalen Definition der Abbildung eines Lochkameramodells sei das Projektionszentrum der Kamera gegeben durch den Zentralprojektionspunkt $e_{c_j} \in \mathbb{R}^3$ für den gilt: $e_{c_j} \notin R_{\text{filled}}$. In diesem befindet sich auch der Ursprung des lokalen Kamerakoordinatensystems (KKS) mit den Koordinatenachsen $\hat{x}_{c_j}, \hat{y}_{c_j}, \hat{z}_{c_j}$ (vgl. Abb. 3.2(c)). Die Bildebene ist in einem Abstand $\hat{f}_{c_j}$, der als Brennweite bezeichnet wird, parallel zur $\hat{x}_{c_j}, \hat{y}_{c_j}$-Ebene ausgerichtet. Die reale Sensorfläche sei gegeben durch die Punkte $\hat{r}_{c_j,\min}$, $\hat{r}_{c_j,\max} \in \mathbb{R}^3$ wie in Abb. 3.2(d) dargestellt. Für diese Punkte gilt ferner, da sie die Sensorfläche symmetrisch aufspannen: $\hat{r}^{(3)}_{c_j,\min} := \hat{r}^{(3)}_{c_j,\max} := \hat{f}_{c_j}$ sowie $\hat{r}^{(i)}_{c_j,\min} = -\hat{r}^{(i)}_{c_j,\max}$, mit $i \in \{1, 2\}$ und

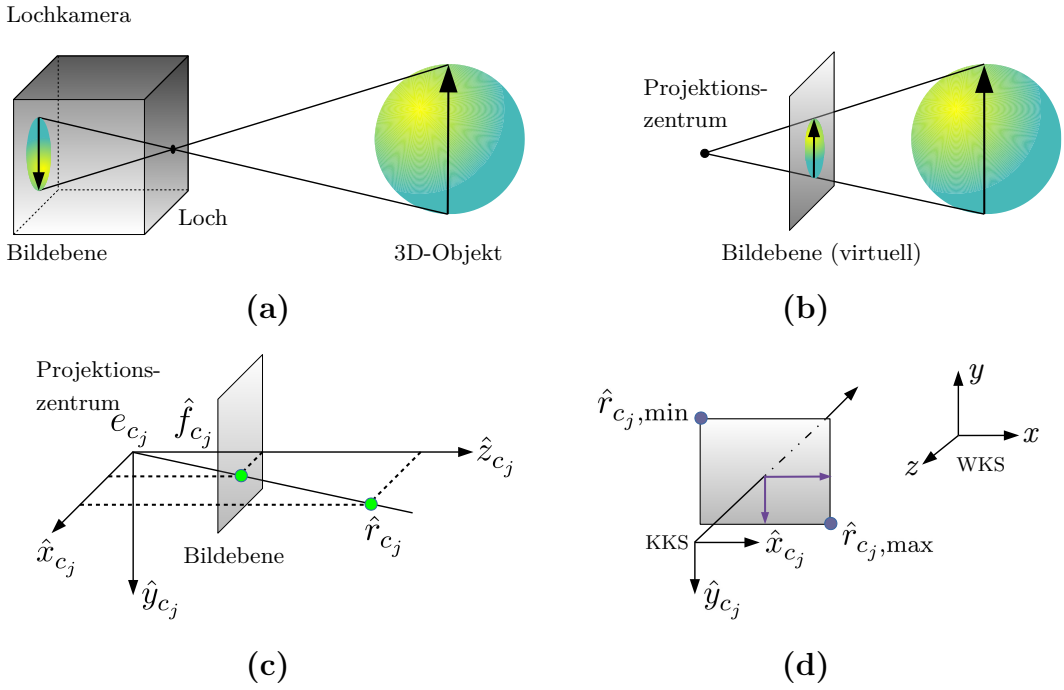


Abb. 3.2: Abbildungsprinzip einer realen Lochkamera (a), Modell einer Lochkamera mit virtueller Bildebene (b), Bestimmung der Koordinaten eines projizierten Punkts in der Bildebene mithilfe des Strahlensatzes (c), Lage einer realen Sensorfläche im lokalen Kamerakoordinatensystem (KKS) und Weltkoordinatensystem (WKS) (d).

$\underset{i}{max}(\hat{r}^{(i)}_{c_j,\min} - \hat{r}^{(i)}_{c_j,\max}) \leq 0$. Die Koordinaten des Kamerakoordinatensystems werden mit den geklammerten Hochindizes angegeben. Weiterhin sei die Menge der Sensorpunkte $\hat{P}_{c_j} \subset \mathbb{R}^3$ definiert durch $\hat{P}_{c_j} := [r^{(1)}_{c_j,\min}, r^{(1)}_{c_j,\max}] \times [r^{(2)}_{c_j,\min}, r^{(2)}_{c_j,\max}] \times \{\hat{f}_{c_j}\}$.

Für die Abbildung eines Raumpunkts r, gegeben im WKS, wird zunächst eine Transformation in das lokale KKS der Kamera c_j vorgenommen. Hierfür wird eine affine Abbildung nach bekanntem Schema entsprechend der Funktion $\text{CamCoords}_{c_j} : \mathbb{R}^3 \to \mathbb{R}^3$ in Formel (3.1) durchgeführt, die für jede Kamera c_j eine spezifische Rotationsmatrix $A_{\text{Rot},c_j} \in \mathbb{R}^{3\times3}$ sowie einen spezifischen Vektor $t_{c_j} \in \mathbb{R}^3$ benötigt.

$$\text{CamCoords}_{c_j}(r) := A_{\text{Rot},c_j} r + t_{c_j} \tag{3.1}$$

Für einen Punkt $\hat{r}_{c_j} := \text{CamCoords}_{c_j}(r)$ lässt sich dessen Projektion auf die Bildebene mithilfe des Strahlensatzes ermitteln. Die Funktion $\text{PinholeProjection}_{c_j} : \mathbb{R}^3 \to \mathbb{R}^3$ zeigt diese Anwendung in Formel (3.2). Die Projektion wird in Abb. 3.2(c) veranschaulicht.

$$\text{PinholeProjection}_{c_j}\left(\hat{r}_{c_j}\right) := \hat{f}_{c_j}\left(\frac{\hat{r}^{(1)}_{c_j}}{\hat{r}^{(3)}_{c_j}}, \quad \frac{\hat{r}^{(2)}_{c_j}}{\hat{r}^{(3)}_{c_j}}, \quad 1\right)^T \text{ mit } \quad \hat{r}^{(3)}_{c_j} \neq 0 \tag{3.2}$$

Eine Kamera hat für einen Sensorpunkt freie Sicht bis zum nächstgelegenen belegten Raumpunkt $r \in R_{\text{filled}}$, der zur Oberfläche eines Objekts gehört. Dementsprechend erfassen Kameras mit ihren Sensorpunkten die Eigenschaften solcher belegter Punkte. Bei Farbkameras handelt es sich dabei um das vom Oberflächenpunkt reflektierte und emittierte Lichtspektrum, welches in Form von Farbwerten gemessen wird. Bei Tiefenkameras wird ein Abstandswert zwischen dem Oberflächenpunkt und der Kamera erfasst. Die Betrachtung von Objekttransparenzen liegt außerhalb des Fokus der vorliegenden Dissertation. Deshalb wird von ausschließlich intransparenten Objekten ausgegangen. Es wird die Annahme getroffen, dass Objekte, die sich in der Szene befinden, für die von den Kamerasensoren messbaren Wellenlängen der elektromagnetischen Strahlung nicht durchdringbar sind und zudem keine spiegelnden Oberflächen besitzen. Dies gilt auch für die Raumbegrenzung des Überwachungsraums.

Reale optische Systeme verwenden Linsen, um eine größere Lichtintensität und eine Verstärkung des Signals zu erreichen. Dadurch wird eine Vielzahl gebündelter Lichtstrahlen auf jeden Punkt der Bildebene gebracht. Hierbei entstehen Beugungen der Lichtstrahlen und weitere Effekte, die zu einem verzerrten Bild führen. Diese Verzerrungen lassen sich nachträglich bei digitalen Systemen verringern, was als Entzerrung bezeichnet wird. Auch die für das Tracking aufgezeichneten Bilder werden entzerrt. Für die weiteren theoretischen Betrachtungen in diesem Kapitel wird jedoch nur das einfache Lochkameramodell verwendet, wodurch keine Linseneigenschaften berücksichtigt werden müssen. Zudem wird zur Vereinfachung vernachlässigt, dass Kameras zugleich Objekte sind und damit Raumpunkte belegen.

3.2 Verdeckungen im 3D-Raum

Nachfolgend werden verschiedene Verdeckungsarten definiert, beginnend bei der kleinsten Einheit, einer Punktverdeckung, bevor darauf aufbauend Verdeckungsvolumina, Objektverdeckungen sowie Objektprojektionsverdeckungen behandelt werden.

3.2.1 Punktverdeckung

Ein gegebener Raumpunkt $r \in R$ kann aus Sicht einer Kamera unter den getroffenen Annahmen in Abschnitt 3.1 entweder **verdeckt** (engl. occluded) oder **unverdeckt** (engl. visible) sein. Betrachtet man weitergehend die Ursache der Verdeckung, so lässt sich im Folgenden ein verdeckter Punkt als **messverdeckt** bezeichnen, wenn sich dieser außerhalb des möglichen Messbereichs des Sensors befindet. Andernfalls wird die Sicht auf ihn durch einen weiteren belegten Raumpunkt blockiert, was **sichtverdeckt** genannt

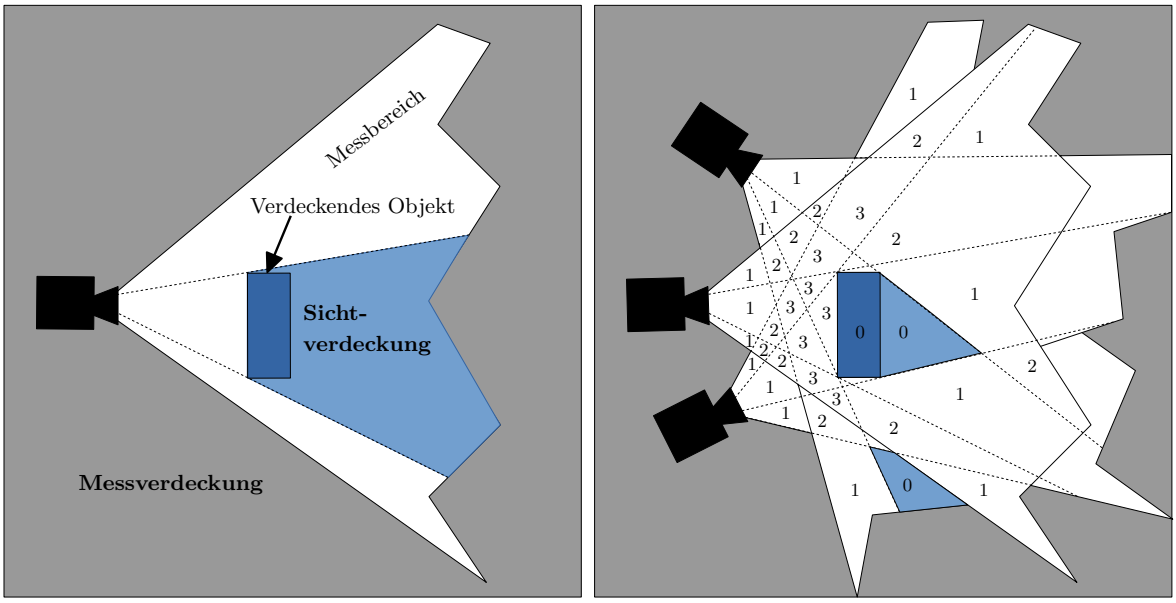


Abb. 3.3: Messbereich einer Kamera und damit verbundene Messverdeckung (a). Innerhalb des Messbereichs befindet sich ein verdeckendes Objekt (dunkelblau), das zu einer Sichtverdeckung führt (hellblau). Der Sichtbarkeitsgrad gibt an, wie viele Kameras freie Sicht auf einzelne Raumpunkte bzw. zusammenhängende Volumina haben (b). Ein nicht einsehbares Volumen mit dem Sichtbarkeitsgrad von 0 wird Verdeckungsvolumen genannt.

wird. In Abbildung 3.3(a) sind beide Verdeckungsursachen für eine Kamera dargestellt. Für die folgende Beschreibung wird davon ausgegangen, dass sich ein Kamerasensor unter der Annahme des Lochkameramodells als eine Menge von **Sichtstrahlen** repräsentieren lässt. Jeder Sichtstrahl hat seinen Anfangspunkt im Zentralprojektionspunkt e_{c_j} der Kamera und verläuft durch einen Sensorpunkt $\hat{p}_{c_j} \in \hat{P}_{c_j}$ der virtuellen Bildebene.

Zunächst soll der Begriff **Messverdeckung** genauer spezifiziert werden. Der reale Sensor einer Kamera c_j weist eine begrenzte Sensorfläche $\hat{P}_{c_j}$ auf, wie bereits in Abschnitt 3.1 definiert. Zudem besitzt jede Kamera eine Vorder- und Rückseite. Typischerweise ergibt sich ein Sichtbereich mit Pyramidenform, der die Sichtstrahlen umfasst. Damit ein Raumpunkt auf die Sensorfläche abgebildet werden kann, muss sich dieser auf einem Sichtstrahl befinden. Dies ist dann der Fall, wenn die Bedingungen der Funktion $\text{Measurable}_{c_j} : \mathbb{R}^3 \to \{0, 1\}$ in Formel (3.3) erfüllt sind. Hierbei wird vernachlässigt, dass sich die virtuelle Bildebene auf dem Sichtstrahl in einem Abstand > 0 vom Koordinatenursprung entfernt, befindet. Damit entspricht die in Formel (3.3) enthalte-

$$\text{Measurable}_{c_j}(r) := \begin{cases} 1 & \text{if} \quad \text{CamCoords}_{c_j}(r)^{(3)} > 0 \\ & \qquad \wedge \quad \text{PinholeProjection}_{c_j}(\text{CamCoords}_{c_j}(r)) \in \hat{P}_{c_j} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

ne Bedingung einer realen Kamera, auch wenn zur Vereinfachung mit der virtuellen Bildebene gerechnet wird. Werden die Bedingungen nicht erfüllt, dann liegt der Punkt außerhalb des Sichtbereichs der Kamera unter gegebener Position und Orientierung im Raum und ist demnach messverdeckt, so wie in Abb. 3.3(a) dargestellt.

Weitere reale Sensoreigenschaften, welche die Abbildungseigenschaften beeinflussen und ebenfalls dem Begriff Messverdeckung zugeordnet werden können, sind z. B. das Zusammenfassen von Sensorpunkten in Pixeln (Pixelquantisierung) und die damit verbundene Grenze des Auflösungsvermögens. Raumpunkte, die zu weit von der Sensorfläche entfernt liegen, können dadurch real keinen Beitrag mehr zu einem Sensorsignal liefern. Dies resultiert in einer Mindestgröße von Objekten und einem maximalen Abstand dieser zur Kamera, um eine Abbildung zu ermöglichen. Solche Effekte werden jedoch im Folgenden nicht betrachtet, da sie die Definitionen von Verdeckungen unnötig verkomplizieren würden.

Im Vergleich zur Messverdeckung tritt eine **Sichtverdeckung** (vgl. Abb. 3.3(a)) dann auf, wenn ein Raumpunkt r zwar auf einem Sichtstrahl liegt, jedoch die Sicht auf diesen durch einen anderen davorliegenden belegten Punkt $\tilde{r}$ aus der Menge R_{filled} physisch blockiert ist. Zur Bestimmung einer Sichtverdeckung wird der Abschnitt des entsprechenden Sichtstrahls zwischen dem betrachteten Raumpunkt r und dem Zentralprojektionspunkt e_{c_j} einer Kamera c_j herangezogen, wie in Formel (3.4) als Menge $M_{c_j}(r)$ angegeben. Handelt es sich bei dem betrachteten Strahl nicht um einen Sichtstrahl, so ergibt sich die leere Menge.

$$M_{c_j}(r) := \begin{cases} \{\tilde{r} \in R | \tilde{r} = r + a(e_{c_j} - r), a \in (0,1)\} & \text{if} \quad \text{Measurable}_{c_j}(r) = 1 \\ \emptyset & \text{otherwise} \end{cases} \tag{3.4}$$

In Formel (3.5) wird zur Bestimmung der Sichtverdeckung eines Punkts die Belegung des zugehörigen Sichtstrahls (sofern vorhanden) überprüft.

$$\text{RayOccupation}_{c_j}(r, \mathcal{T}) := \begin{cases} 1 & \text{if} \quad \exists \tilde{r} \in M_{c_j}(r) \quad \text{with} \quad \tilde{r} \in \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

Die Funktion $\text{RayOccupation}_{c_j} : \mathbb{R}^3 \times \mathcal{P}(\mathbb{R}^3) \to \{0, 1\}$ sei hierfür definiert. Dabei wird die Menge der belegten Punkte, auf die getestet wird, mit $\mathcal{T}$ bezeichnet. Es genügt, nur die äußeren Objektpunkte (Oberflächenpunkte) zu betrachten ($\mathcal{T} = \partial R_{\text{filled}}$) und die inneren Objektpunkte der Menge $R^{\circ}_{\text{filled}}$ im Weiteren zu vernachlässigen. Denn es gilt: wandert man für einen Punkt aus der Menge $R^{\circ}_{\text{filled}}$ entlang des zugehörigen Sichtstrahls in eine der beiden Richtungen, so wird man immer auf mindestens einen Punkt aus der Menge $\partial R_{\text{filled}}$ treffen. Zusammenfassend ist ein Raumpunkt r dann

verdeckt (occluded) und für einen Kamerasensor c_j nicht sichtbar, wenn dieser nicht auf einem Sichtstrahl des Sensors liegt (Punkt ist messverdeckt), oder wenn sich ein weiterer Punkt aus der Menge R_{filled} bzw. $\partial R_{\text{filled}}$ zwischen der Kamera und dem Raumpunkt auf dem Sichtstrahl befindet (Punkt ist sichtverdeckt). Durch die Funktion $\text{PointOcclusionCam}_{c_j} : \mathbb{R}^3 \rightarrow \{\textit{visible}, \textit{occluded}\}$ wird dies in Formel (3.6) definiert.

$$\text{PointOcclusionCam}_{c_j}(r) := \begin{cases} \textit{occluded} & \text{if} \quad \text{RayOccupation}_{c_j}(r, \mathcal{T} = \partial R_{\text{filled}}) = 1 \\ & \qquad \vee \quad \text{Measurable}_{c_j}(r) = 0 \\ \textit{visible} & \text{otherwise} \end{cases} \tag{3.6}$$

Betrachtet man alle Sensoren eines Kamerasystems C, so gilt bei der Umsetzung des Prinzips der sensorischen Kompensation ein Punkt dann als unverdeckt (visible), wenn mindestens eine Kamera c_j existiert, für die der Punkt unverdeckt ist. Dies wird zum Ausdruck gebracht durch die Funktion $\text{PointOcclusionSys}_C : \mathbb{R}^3 \rightarrow \{\textit{visible}, \textit{occluded}\}$, die in Formel (3.7) definiert wird.

$$\text{PointOcclusionSys}_C(r) := \begin{cases} \textit{visible} & \text{if} \quad \exists c_j \in C : \text{PointOcclusionCam}_{c_j}(r) = \textit{visible} \\ \textit{occluded} & \text{otherwise} \end{cases} \tag{3.7}$$

Mit dieser Definition geht auch einher, dass durch die Erweiterung des Kamerasystems um zusätzliche Kameras die Sichtbarkeit von Raumpunkten erhöht werden kann. Ob dies möglich ist, hängt jedoch auch von der konkreten Platzierung der Kameras und den Objekten im Überwachungsraum ab. Die Funktion gibt an, in wie vielen Kameras ein Raumpunkt unverdeckt ist, was im Folgenden als **Sichtbarkeitsgrad** bezeichnet und durch die Funktion $\text{VisibilityGrade}_C : \mathbb{R}^3 \rightarrow \mathbb{N}$ in Formel 3.8 definiert wird.

$$\text{VisibilityGrade}_C(r) := |\{c_j \in C | \text{PointOcclusionCam}_{c_j}(r) = \textit{visible}\}| \tag{3.8}$$

Ist ein Raumpunkt für das gesamte Kamerasystem C verdeckt, so weist er einen Sichtbarkeitsgrad von 0 auf, das heißt: kein einziger Sensor hat eine freie Sicht auf ihn. Alle sichtbaren Raumpunkte haben einen Sichtbarkeitsgrad von ≥ 1.

Ist ein Punkt sichtbar (weder messverdeckt noch sichtverdeckt) und gleichzeitig belegt, so handelt es sich dabei um einen zur Kamera nächstgelegenen Oberflächenpunkt eines Objekts, der zugleich der Punkt auf dem zugehörigen Sichtstrahl ist, der auch das Signal in dem Sensorpunkt erzeugt. Das heißt, der Raumpunkt r projiziert dann nicht nur auf einen Sensorpunkt $\hat{p} \in \hat{P}_{c_j}$, sondern er bildet auch tatsächlich darauf ab, indem er ein messbares Signal hervorruft. Die Bedingung hierfür geht aus der Funktion

$\text{SensorSignal}_{c_j} : \mathbb{R}^3 \to \{0,1\}$ in Formel 3.9 hervor.

$$\text{SensorSignal}_{c_j}(r) := \begin{cases} 1 & \text{if} \quad \text{PointOcclusionCam}_{c_j}(r) = \textit{visible} \\ & \quad \wedge \quad r \in R_{\text{filled}} \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Betrachtet man das gesamte Kamerasystem, so ist ein Raumpunkt dann ein Messpunkt, falls er in mindestens einer Kamera ein Signal erzeugt, wie in der Funktion $\text{SensorSignalSys}_C : \mathbb{R}^3 \to \{0,1\}$ in Formel 3.10 definiert. Auch hiermit wird noch einmal das Prinzip der sensorischen Kompensation verdeutlicht.

$$\text{SensorSignalSys}_C(r) := \begin{cases} 1 & \text{if} \quad \exists c_j \in C : \text{PointOcclusionCam}_{c_j}(r) = \textit{visible} \\ & \quad \wedge \quad r \in R_{\text{filled}} \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

3.2.2 Verdeckungsvolumina

Im vorherigen Abschnitt wurde definiert, wann ein Punkt für eine einzelne Kamera oder für ein Kamerasystem verdeckt oder sichtbar ist. Demnach lässt sich der Überwachungsraum unterteilen in eine Menge **sichtbarer** Raumpunkte R_{vis,c_j} sowie eine Menge **verdeckter** Raumpunkte R_{occ,c_j}, wobei $R_{\text{occ},c_j} := R \setminus R_{\text{vis},c_j}$ gilt. Für die Punkte dieser Mengen liegt jedoch im Raum keine beliebige Verteilung vor, sondern sie lassen sich jeweils in topologisch zusammenhängende Teilmengen unterteilen, woraus sich für die Menge der verdeckten Raumpunkte eine Anzahl an Verdeckungsvolumina ergibt. Dies wird im Folgenden präziser definiert. Hierfür wird zunächst eine Sichtstrahlunterteilung durchgeführt. Liegt auf einem Sichtstrahl ein belegter Raumpunkt, für den gilt: $\text{SensorSignal}_{c_j}(r) = 1$, so kann der zugehörige Sichtstrahl damit in einen sichtbaren und einen verdeckten Abschnitt unterteilt werden.

Für solch eine Unterteilung werden alle belegten Raumpunkte benötigt, die auf dem Sichtstrahl liegen, welcher durch den Zentralprojektionspunkt e_{c_j} der Kamera c_j und dem betrachteten Raumpunkt r definiert ist. Diese belegten Raumpunkte werden von der Menge $Q_{c_j}(r)$ repräsentiert, so wie in Formel 3.11 angegeben.

$$Q_{c_j}(r) := \{t \in R_{\text{filled}} | \exists a \geq 0 : t = e_{c_j} + a(r - e_{c_j})\} \tag{3.11}$$

Davon spielt nur der nächste belegte Punkt $\tilde{r}$ aus Sicht der Kamera eine Rolle, der sich vor dem betrachteten Raumpunkt r befindet und einen Messpunkt darstellt (sofern er überhaupt existiert). Die Menge sichtbarer Raumpunkte R_{vis,c_j} ergibt sich aus allen freien

Raumpunkten, die eine geringere Euklidische Distanz zum Zentralprojektionspunkt der Kamera aufweisen als der Messpunkt (sofern vorhanden) und auf demselben Sichtstrahl liegen (vgl. Formel (3.12)). Der Messpunkt selbst ($\text{SensorSignal}_{c_j}(r) = 1$) wird ebenfalls zu den sichtbaren Raumpunkten gezählt. Zusätzlich gehören auch alle Raumpunkte dazu, die auf einem Sichtstrahl liegen, der keinen Punkt aus der Menge R_{filled} enthält.

$$\begin{aligned} R_{\text{vis},c_j} := &\{r \in R | (Q_{c_j}(r) = \emptyset) \wedge (\text{Measureable}_{c_j}(r) = 1)\} \quad \cup \\ &\{r \in R | (Q_{c_j}(r) \neq \emptyset) \wedge (||r - e_{c_j}||^2 \leq ||\underset{\tilde{r} \in Q_{c_j}(r)}{\text{argmin}} - e_{c_j}||^2)\} \end{aligned} \tag{3.12}$$

Zur komplementären Menge verdeckter Raumpunkte R_{occ,c_j} gehören entsprechend alle Punkte auf allen Sichtstrahlen, die einen größeren Abstand als der zugehörige Messpunkt aufweisen unter der Voraussetzung, dass der jeweilige Sichtstrahl einen Messpunkt besitzt.

Betrachtet man das gesamte Kamerasystem, so lassen sich die Mengen der einzelnen Kameras folgendermaßen miteinander kombinieren: Für die Menge unverdeckter Punkte ergibt sich eine Vereinigung der Teilmengen R_{vis,c_j} über alle $|C|$ Sensoren (vgl. Formel 3.13).

$$R_{\text{vis},C} := \bigcup_{c_j \in C} R_{\text{vis},c_j} \tag{3.13}$$

Die Menge an verdeckten Punkten hingegen ergibt sich als Schnittmenge aus den Teilmengen R_{occ,c_j} der einzelnen Kameras (vgl. Formel 3.14).

$$R_{\text{occ},C} := \bigcap_{c_j \in C} R_{\text{occ},c_j} \tag{3.14}$$

Im Folgenden wird jede topologisch zusammenhängende Teilmenge aus $R_{\text{occ},C}$ als **Verdeckungsvolumen** bezeichnet. Verdeckungsvolumina sind sensorisch nicht einsehbar, unabhängig davon, ob sie belegt sind oder nicht. Sie besitzen einen Sichtbarkeitsgrad von 0. Die Kombination der verdeckten und sichtbaren Bereiche für ein Kamerasystem nach den Formeln (3.13) und (3.14) wird später in ähnlicher Weise auch algorithmisch relevant. In Abschnitt 5.3 wird das Verfahren der Visuellen Hülle genauer vorgestellt, welches eine Verschneidung approximierter verdeckter Volumina vornimmt, die durch Rückprojektion von Objektkonturen aus mehreren Kameras generiert werden. Dies wird in dem Zusammenhang auch als **Volumenverschnitt** bezeichnet.

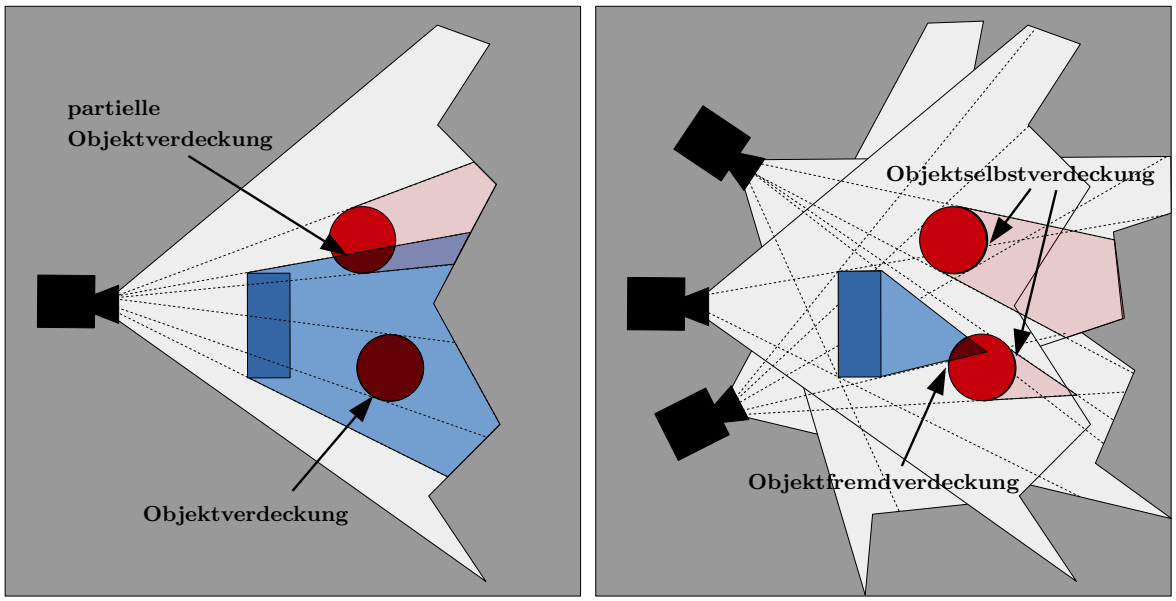


Abb. 3.4: Veranschaulichung von Objektverdeckungen. Ein blaues Objekt blockiert die Sicht auf die roten Kugeln abhängig von der Raumposition vollständig oder teilweise (a). Deshalb ist eine Kugel verdeckt und die andere partiell verdeckt. Durch die Verwendung mehrerer Kameras kann die Sichtbarkeit der Kugeln erhöht werden (b). Sie bleiben dennoch partiell verdeckt, da nicht ihre gesamte Objektoberfläche sensorisch erfasst wird. Die Kugeln verdecken sich selbst (Objektselbstverdeckung). Die untere Kugel wird zusätzlich noch von dem blauen Objekt verdeckt (Objektfremdverdeckung).

3.2.3 Objektverdeckung

Die beschriebenen Verdeckungsvolumina werden von verdeckenden Objekten erzeugt und können abhängig von ihrer Größe auch auf weitere Objekte wirken, wie in Abb. 3.4(a) dargestellt. Betrachtet man eine Verdeckungssituation aus der Sicht eines Objekts, so lässt sich unter dem Begriff **Objektverdeckung** verstehen, ob die Oberfläche des Objekts, oder ein Teil davon, verdeckt ist. Da die inneren Objektpunkte der Menge $R^{\circ}_{\text{filled}}$ nach getroffener Annahme immer verdeckt sind, werden diese im Folgenden nicht weiter betrachtet. Ist ein Objekt o mit R_{o} im Überwachungsraum gegeben, so werden in Formel (3.15) durch die Funktion $\text{ObjectOcclusionSys}_C : O \rightarrow \{\textit{visible}, \textit{partially occ.}, \textit{occluded}\}$ folgende Verdeckungsfälle definiert:

$$\text{ObjectOcclusionSys}_C\,(o) := \begin{cases} \textit{visible} & \text{if} \quad \forall r \in \partial R_{\text{o}} : \\ & \text{PointOcclusionSys}_C\,(r) = \textit{visible} \\ \textit{occluded} & \text{if} \quad \forall r \in \partial R_{\text{o}} : \\ & \text{PointOcclusionSys}_C\,(r) = \textit{occluded} \\ \textit{partially occ.} & \text{otherwise} \end{cases} \tag{3.15}$$

Betrachtet man eine einzelne Kamera, so gibt es lediglich die Fälle partiell verdeckt und verdeckt. Die Funktion $\text{ObjectOcclusionCam}_{c_j} : O \rightarrow \{\textit{partially occ.}, \textit{occluded}\}$ wird hierfür in Formel (3.16) definiert.

$$\text{ObjectOcclusionCam}_{c_j}(o) := \begin{cases} \textit{occluded} & \text{if} \quad \forall r \in \partial R_\text{o} : \\ & \text{PointOcclusionCam}_{c_j}(r) = \textit{occluded} \\ \textit{partially occ.} & \text{otherwise} \end{cases} \tag{3.16}$$

Mit Formel (3.16) wird zum Ausdruck gebracht, dass ein Objekt in einer einzelnen Kamera immer auch einen Teil seiner eigenen Oberfläche verdeckt, sofern das Objekt nicht aus einer unendlich dünnen Schicht oder einem Punkt besteht. Dies wird als **Objektselbstverdeckung** bezeichnet. In dieser Dissertation wird von solch einer Verdeckung dann gesprochen, wenn mindestens ein Punkt der Objektoberfläche existiert, der sichtverdeckt ist, wobei die Sichtverdeckung von einem oder mehreren Punkten verursacht wird, die topologisch gesehen zum selben Objekt gehören. Eine entsprechende Funktion $\text{ObjectSelfOcclusionCam}_{c_j} : O \rightarrow \{0, 1\}$ kann definiert werden als:

$$\text{ObjectSelfOcclusionCam}_{c_j}(o) := \begin{cases} 1 & \text{if} \quad \exists r \in \partial R_\text{o} : \\ & \text{RayOccupation}_{c_j}(r, \mathcal{T} = \partial R_\text{o}) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

Nach dem Prinzip der sensorischen Kompensation lässt sich auch für das gesamte Kamerasystem eine Objektselbstverdeckung beschreiben. Dafür sei die Funktion $\text{ObjectSelfOcclusionSys}_C : O \rightarrow \{0, 1\}$ definiert als:

$$\text{ObjectSelfOcclusionSys}_C(o) := \begin{cases} 1 & \text{if} \quad \exists r \in \partial R_\text{o} : \forall c_j \in C : \\ & \text{RayOccupation}_{c_j}(r, \mathcal{T} = \partial R_\text{o}) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

Wird die Sicht auf mindestens einen Punkt des Objekts durch ein anderes Objekt blockiert, so wird dies **Objektfremdverdeckung** genannt. In den Formeln (3.17) und (3.18) müsste die Bedingung wie folgt angepasst werden, um eine Objektfremdverdeckung formal auszudrücken:

$$\text{RayOccupation}_{c_j}(r \in R_\text{o}, \mathcal{T} = \partial R_\text{filled} \setminus R_\text{o}) = 1 \tag{3.19}$$

Ein Raumpunkt kann von mehreren Punkten unterschiedlicher Objektzugehörigkeit verdeckt werden. Das heißt: Objektselbstverdeckung und Objektfremdverdeckung können gleichzeitig auftreten und sich überlagern, wie in Abb. 3.4(b) dargestellt. In dieser Dissertation liegt der Fokus auf Objektfremdverdeckungen, da die Wirkung gegebener

Verdeckungsvolumina der statischen Objekte im Überwachungsraum auf die betrachteten dynamischen Objekte (Personen) beim Tracking beschrieben wird. Da häufig gleichzeitig auch Objektselbstverdeckungen auftreten, wird meist vereinfachend von Objektverdeckungen gesprochen.

3.3 Verdeckungen im 2D-Bild

Die bisherige Beschreibung von Verdeckungen bezog sich auf den 3D-Raum: auf verdeckte einzelne Punkte, auf verdeckte topologisch zusammenhängende Punkte, die Verdeckungsvolumina bilden sowie auf verdeckte Punkte, die eine Objektzugehörigkeit besitzen und demnach belegt sind. Die Betrachtungen erfolgten jeweils für einzelne und mehrere Sensoren eines Kamerasystems. Zur Beschreibung von Verdeckungen in der Bildebene bietet sich eine weitere Definition an, die der Vollständigkeit halber im Folgenden mit angegeben wird.

Objektprojektionsverdeckung

So wie man eine Verdeckungssituation im Bild intuitiv als Mensch beschreiben würde, wird ein Objekt dann als unverdeckt (visible) bezeichnet, wenn man eine freie Sicht auf das Objekt hat und die Sicht nicht durch ein anderes Objekt blockiert wird (Objektfremdverdeckung). Die Tatsache, dass nicht die gesamte Objektoberfläche für den Sensor sichtbar ist, sprich gleichzeitig Objektselbstverdeckungen vorliegen, wird dabei ignoriert (außer es wird Bezug aus bestimmte Körperteile, wie z. B. das Gesicht, genommen). Jedoch sollte das Objekt vollständig auf die Sensorfläche projizieren und nicht durch den Bildrand abgeschnitten sein (Messverdeckung), damit es als unverdeckt gilt. Das heißt, die Objektkontur muss vollständig im Bild enthalten sein. Andernfalls wird das Objekt als partiell verdeckt (partially occluded) bezeichnet, wenn ein Teil sichtbar ist oder als verdeckt (occluded), wenn kein Teil sichtbar ist.

$$\text{ObjectProjOcclusion}_{c_j}(o) := \begin{cases} \textit{visible} & \text{if} \quad \forall r \in R_\text{o}: \\ & \text{RayOccupation}_{c_j}(r, \mathcal{T} = \partial R_\text{filled} \backslash \partial R_\text{o}) = 0 \\ & \wedge \quad \text{Measurable}_{c_j}(r) = 1 \\ \textit{occluded} & \text{if} \quad \forall r \in R_\text{o}: \\ & \text{PointOcclusionCam}_{c_j}(r) = \textit{occluded} \\ \textit{partially} & \\ \textit{occluded} & \text{otherwise} \end{cases} \tag{3.20}$$

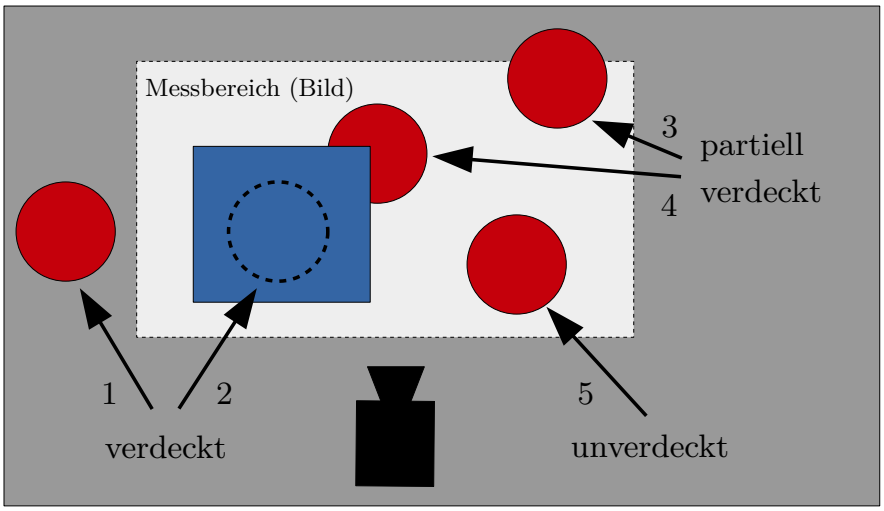


Abb. 3.5: Beschreibung von Objektprojektionsverdeckungen anhand einer roten Kugel an unterschiedlichen Positionen im Bildraum. Fall 1: Die Kugel ist verdeckt aufgrund einer Messverdeckung (1) oder einer Sichtverdeckung durch das blaue Objekt (2). Fall 2: Die Kugel ist partiell verdeckt aufgrund einer Messverdeckung (3) oder einer Sichtverdeckung durch das blaue Objekt (4). Fall 3: Es tritt keine Verdeckung auf, die Objektprojektion ist in Gänze sichtbar (5).

Die verschiedenen Verdeckungssituationen werden in Abb. 3.5 visualisiert. Für die Verdeckungsfälle der Projektion eines Objekts im Bild, genannt **Objektprojektionsverdeckung** sei eine Funktion $\text{ObjectProjOcclusion}_{c_j} : O \rightarrow \{visible, partially\ occ., occluded\}$ definiert, wie in Formel (3.20) angegeben.

3.4 Zusammenfassung

Ziel dieses Kapitels war die formale Beschreibung von Verdeckungen. Dafür wurden anfangs Definitionen zum Überwachungsraum und dem Multi-View-Kamerasystem eingeführt. Auf dieser Grundlage wurden anschließend Verdeckungsbetrachtungen im 3D-Raum durchgeführt, die für das gesamte Kamerasystem gelten und die sensorische Kompensation einzelner Kameras berücksichtigen. Beginnend bei der kleinsten Verdeckungseinheit, einer Punktverdeckung, wurden nachfolgend Verdeckungsvolumina definiert, die aus topologisch zusammenhängenden Punktverdeckungen resultieren. Die Wirkung gegebener Verdeckungsvolumina auf sich selbst oder andere Objekte wird als Objektverdeckung bezeichnet, wobei die Unterscheidung zwischen Objektselbstverdeckung und Objektfremdverdeckung getroffen wurde. Zusätzlich wurde auch der Begriff Sichtbarkeitsgrad definiert, welcher angibt für wie viele Kameras ein betrachtetes Volumen unverdeckt ist. Für die Beschreibung von Verdeckungen im 2D-Bild wurde der Vollständigkeit halber der Begriff Objektprojektionsverdeckung eingeführt, wobei die meisten Verdeckungsbetrachtungen dieser Dissertation im 3D-Raum durchgeführt

werden. Mit den definierten Begriffen lassen sich die nachfolgend dargestellten Verdeckungssituationen bei der 3D-Rekonstruktion und dem Personen-Tracking konkreter beschreiben.

Kapitel 4

Stand der Forschung

In dieser Dissertation wird die Zielstellung verfolgt, Personen anhand von Kamerabildern im 3D-Raum zu lokalisieren und über Bildersequenzen hinweg zu verfolgen, was als Personen-Tracking bezeichnet wird. Zu Beginn dieses Kapitels werden in Abschnitt 4.1 verwendete Begrifflichkeiten zu verschiedenen Forschungsbereichen erläutert, die mit dem Personen-Tracking in Verbindung stehen. Für die davon relevanten Forschungsbereiche des Human Trackings und der Human Pose Estimation wird anschließend in Abschnitt 4.2 eine sensorbasierte Unterteilung betrachtet, nach der sich Teilgebiete identifizieren lassen, die speziell für die Zielstellung dieser Dissertation von Interesse sind. In Abschnitt 4.3 wird nachfolgend eine Einführung in das Human Tracking gegeben und auf häufig eingesetzte Filtertechniken eingegangen. In dem sich anschließenden Abschnitt 4.4 stehen Verfahren der Human Pose Estimation im Vordergrund. Es werden klassische Verfahren betrachtet sowie jüngere Verfahren, die Techniken des Deep Learnings einsetzen. Anschließend wird in Abschnitt 4.5 eine Literaturübersicht zu speziellen Verfahren des Human Trackings und der Human Pose Estimation gegeben, die eine 3D-Rekonstruktion der Visuellen Hülle durchführen und die Schätzungen basierend auf diesen 3D-Daten vornehmen. Die betrachteten relevanten Forschungsbereiche werden in Abschnitt 4.6 bezüglich ihres potentiellen Umgangs mit Verdeckungen durch statische Objekte bewertet. Daraus resultierend wird die wissenschaftliche Lücke aufgezeigt, zu der die vorliegende Dissertation einen Beitrag liefern soll. Die Inhalte dieses Kapitels werden in Abschnitt 4.7 zusammengefasst.

4.1 Begrifflichkeiten

In der Literatur wird allgemein für das Tracking von Personen der Ausdruck **Human Tracking** verwendet [Poppe, 2007], in Abgrenzung zu Verfahren des **Object Trackings** oder allgemein des **Trackings**, bei denen auch andere Objekte als Personen von Interesse sein können. Beim Human Tracking werden Personen in Videosequenzen (von Frame zu Frame) verfolgt, um konsistente Pfade für diese zu erzeugen, die als **Trajektorien** bezeichnet werden. Die temporale Korrelation der Sensordaten aufeinander folgender Frames wird dabei ausgenutzt, um die Konsistenz herzustellen. Mit **Human Detection**

wird ein Suchmechanismus bezeichnet, der dazu dient, Personen in Einzelbildern zu lokalisieren. Dieser wird auch häufig für die Initialisierung eines Tracking-Verfahrens benötigt. In dieser Dissertation wird die synonyme Bezeichnung „Objektdetektion“ verwendet (vgl. Abschnitt 2.2).

Das Human Tracking ist von der **Human (Body) Pose Estimation** (Körperposenschätzung) zu unterscheiden, die auch als **Human Pose Recovery** bezeichnet wird. Im Folgenden wird synonym für beides vorwiegend der kürzere Begriff **Pose Estimation** (Posenschätzung) verwendet. In dem Übersichtsartikel von [Poppe, 2007] versteht man unter Pose Estimation den Prozess, in welchem die menschliche Pose möglichst präzise basierend auf sensorischen Daten geschätzt wird. Die Pose wird dabei häufig durch Gelenke und Gliedmaßen repräsentiert, für die die Werte verschiedener Parameter wie Gelenkwinkel bestimmt werden. Modellbasierte Ansätze verwenden dazu Modelle, welche die kinematischen Eigenschaften des Körpers repräsentieren, sprich die skelettartige bewegliche Struktur einer Person (Skelettmodell) oder die Form der Körperteile (volumetrische Modelle oder Oberflächenmodelle). Dabei werden insbesondere die großen Körperteile wie Rumpf, Kopf und Gliedmaßen für die Pose Estimation modelliert und weniger die Finger, Gesichtsmerkmale oder sonstige feingliedrige Strukturen.

Hängt bei der Pose Estimation eine Schätzung von den Schätzergebnissen der vorhergehenden Frames ab, so ist eine Tracking-Komponente integraler Bestandteil des Verfahrens. In diesem Zusammenhang wird mitunter der Begriff **Human Body Tracking** verwendet, der auch zum Ausdruck bringt, dass ein detailliertes Körpermodell (engl. Body Model) involviert ist.

Weitere Begriffe, die ebenfalls synonym für die Posenschätzung in Sequenzen von Frames verwendet werden, sind **Human Motion Analysis** oder **Motion Capture**. Motion Capture ist ein etablierter Begriff und bedeutet die Aufnahme und Analyse von (menschlichen) Bewegungen. Er wurde geprägt von den markerbasierten, kommerziellen Systemen des **Marker-based Motion Capture**, welche seit längerem in Studios unter geeigneten Bedingungen eingesetzt werden, um digitale Charaktere, insbesondere für die Computerspiel- und Filmindustrie, zu animieren. Dabei werden verschiedene Körperteile, wie beispielsweise Gelenke, mit Markern versehen. Digitale Charaktere werden damit erfolgreich zum Leben erweckt [Wikipedia, 2020], wie beispielsweise der Affe Caesar im Film „Planet der Affen“ [Perry, 2014]. Dem **Markerless Motion Capture** lassen sich neuere Systeme zuordnen, die zum Ziel haben, Bewegungen ohne Marker einzufangen. Hierfür sind inzwischen verschiedene Produkte erhältlich. Beispielsweise können mit Systemen der Firma „The Captury“ [Hasler et al., 2020] in Echtzeit bis zu drei Personen mit 6 bis 24 Kameras getrackt und deren Posen erfasst werden.

Ein neuerer Teilbereich der Pose Estimation, an dem geforscht wird und der vermutlich auch eine Rolle bei den kommerziellen Systemen des Markerless Motion Capturings spielt, wird als **Deep Human Pose Estimation** bezeichnet. Hierbei werden spezielle Algorithmen und Neuronale Netze des Deep Learnings eingesetzt, um Posenschätzungen aus Bildern und Videosequenzen zu generieren.

Pose Recognition (Posenerkennung) ist ein in der Literatur ebenfalls geläufiger Begriff und bezieht sich auf die Nachverarbeitung geschätzter Posen, auf deren Klassifikation [Poppe, 2007]. Die geschätzten Posen werden dabei auf Posenklassen abgebildet, beispielsweise die Klassen „Stehen“, „Sitzen“ und „Laufen“. Posen werden dadurch interpretiert; ihnen wird eine semantische Bedeutung zugewiesen. In der Literatur findet man anwendungsabhängig weitere Begriffe für Verfahrensklassen, die ein Human Tracking oder eine Pose Estimation voraussetzen und eine Klassifikationskomponente besitzen, wie beispielsweise **Behavior Analysis**, **Movement Recognition**, **Motion Recognition**, **Action and Activity Recognition**, **Event and Anomaly Detection** sowie **Understanding of Human Actions and Behavior**. Dabei kann die Erkennung verschiedenster Aktionen, Ereignisse, Aktivitäten und Verhaltensweisen von einer oder mehreren Personen in einem oder mehreren Frames das Ziel sein [Moeslund et al., 2006].

Anwendungsbereiche für ein Human Tracking, eine Pose Estimation und darauf aufbauende Funktionalitäten sind beispielsweise:

- Sicherheitsanwendungen zur Überwachung von Personen in (öffentlichen) Einrichtungen, um abnormales Verhalten zu erkennen, das zu Schäden wie z. B. Verletzungen anderer Personen führen kann,
- Motion Capturing zur Digitalisierung menschlicher Bewegungen für die Unterhaltungsindustrie,
- Erfassung menschlicher Bewegungen und weiterer Eigenschaften für die Interaktion mit (mobilen) Robotern,
- Analyse des Verhaltens von Personen und ihren Bewegungen zur Umsetzung nutzeradaptiver Komfortfunktionen in intelligenten Räumen und Smart Homes,
- Das SIMERO-Szenario: Gewährleistung der Sicherheit von Menschen bei manipulatorischen Eingriffen von Robotern in ihre Umgebung (Mensch-Roboter-Koexistenz und -Kooperation).

4.2 Relevante Forschungsgebiete

Die relevanten Forschungsgebiete, die im Folgenden betrachtet werden, sind die des Human Trackings und der (Human) Pose Estimation. Auf die Verwendung synonymer Begriffe (außer deren deutschen Übersetzungen) wird verzichtet. Betrachtet man diese Forschungsgebiete genauer, so zeigt sich eine Vielzahl unterschiedlichster Verfahren, die nach verschiedenen Kriterien weiter kategorisiert werden können. In Abb. 4.1 wird dazu ein bestimmtes Kriterium herausgegriffen: die Art des passiven Kamerasystem, das zum Einsatz kommt. Aktive Kamerasysteme werden nicht betrachtet (vgl. Abschnitt 2.1). Es gibt einen Forschungsbereich zu Verfahren des Human Trackings, die monokulare Bilder einer Einzelkamera verarbeiten. Abhängig von den Anwendungsszenarien, die sehr vielfältig sein können, werden dabei aus den Bildern unterschiedlichste Merkmale gewonnen. In Abschnitt 4.3 wird eine Übersicht über einzelne Verfahren dieser Kategorie gegeben und der Unterschied des Human Trackings im Bildraum und im Zustandsraum verdeutlicht. Für letzteres kommen häufig Filtertechniken wie beispielsweise ein Kalmanfilter zum Einsatz. Ein weiterer Forschungsbereich kann der Pose Estimation zugeschrieben werden, die basierend auf monokularen Bildern vorgenommen wird. Verfahrensabhängig kann dabei sowohl die Schätzung einer 2D-Pose als auch einer 3D-Pose das erklärte Ziel sein. Zu diesem Bereich wird in Abschnitt 4.4 eine Einführung und Übersicht gegeben.

Für ein Kamerasystem bestehend aus C Kameras werden andere oder ergänzende Verfahren eingesetzt als für die Verarbeitung monokularer Bilder. In Abb. 4.1 wird prinzipiell zwischen einer Multi-View-Kameratopologie und einer Topologie verteilter Kameras (engl. Distributed Cameras) unterschieden. Für eine Multi-View-Kameratopologie wie

	1 Kamera	C Kameras		
	Monokular	Multi-View-Kameras (überlappend)		Verteilte Kameras (nicht überlappend)
		Ohne 3D Rekonstruktion	Mit 3D Rekonstruktion	
Human Tracking	●	●	●	●
Human Pose Estimation	●	●	●	N/A

Abb. 4.1: Relevante Teilbereiche der Forschungsgebiete Human Tracking und Human Pose Estimation. Je größer der Kreis, desto größer wird nach subjektiver Abschätzung der Autorin der Forschungsbereich in Relation zu den anderen dargestellten Bereichen eingestuft (Anzahl Publikationen). Der Fokus dieser Arbeit wurde auf die schraffierten Forschungsgebiete gelegt.

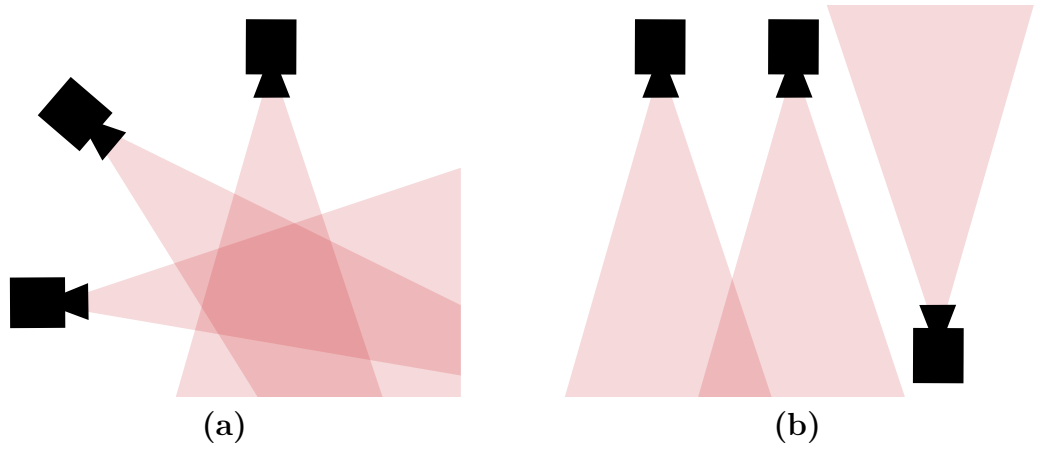


Abb. 4.2: Multi-View-Kameratopologie mit sich überlappenden Sichtbereichen (a) und Topologie verteilter Kameras mit Sichtbereichen, die sich häufig nicht überlappen (b).

in Abb. 4.2(a) ist charakteristisch, dass sich größere überlappende Kamerasichtbereiche ergeben und damit der Raum und die enthaltenen Personen und Objekte von mehreren Seiten betrachtet werden können. Solch eine Kameratopologie wird insbesondere eingesetzt, wenn das Ziel verfolgt wird, 3D-Informationen wie 3D-Volumina oder 3D-Posen zu erfassen.

Hingegen wird für verteilte Kameras wie in Abb. 4.2(b) im Kontext des Human Trackings angenommen, dass diese an beliebigen Orten platziert sein können. Damit ergeben sich nicht zwangsläufig gemeinsame Überwachungsvolumina für mehrere Kameras. Eine Überlappung kann zwar möglich sein, aber ist nicht zwingend erforderlich. Betrachtete Anwendungen zielen auf ein Verfolgen von Personen über größere Distanzen hinweg ab. Beispielsweise könnte das Ziel sein, Personen in einem Gebäude durch mehrere Räume zu verfolgen. Dazu müssen entsprechend geeignete Merkmale extrahiert werden, die eine Zuordnung der Teiltrajektorien verschiedener Kameras zueinander ermöglichen sowie die Wiedererkennung der Personen erlauben. Dieser Forschungsbereich liegt außerhalb des Fokus der vorliegenden Dissertation, ebenso wie die Human Pose Estimation für verteilte Kameras, die auch einen Forschungsbereich darstellen könnte, was aber nicht näher untersucht wird und deshalb mit dem Vermerk „N/A“ in Abb. 4.1 versehen ist.

Multi-View-Kamerasysteme werden sowohl für das Human Tracking als auch die Human Pose Estimation eingesetzt. Jeweils kann weiter darin unterschieden werden, ob die Durchführung einer 3D-Rekonstruktion Bestandteil der Verfahren ist oder nicht. Wird keine 3D-Rekonstruktion vorgenommen, so werden die Kamerabilder direkt herangezogen, um Betrachtungen im 3D-Raum zu bewerten. Dafür kann zunächst eine unabhängige Auswertung der Kamerabilder, wie bei einer 2D-Posenschätzung, vorgenommen werden, bevor eine Fusion von Merkmalen oder Schätzergebnissen für den 3D-Raum durchgeführt wird. Das Human Tracking mit einfachen Modellen (ohne Rekonstruk-

tion) stellt hierbei nach eigener Einschätzung einen eher kleineren Forschungsbereich dar. Für Betrachtungen im 3D-Raum bieten sich detailliertere Körpermodelle für eine Pose Estimation (Skelettmodelle und Volumenmodelle wie Oberflächenmodelle) stärker an, um Korrespondenzen ausfindig machen zu können. Dieser aktuelle und wachsende Forschungsbereich, verfolgt unter anderem das Ziel, 3D-Posen mehrerer Personen unter verschiedensten Gegebenheiten (inklusive Verdeckungen) zu schätzen. Dies erfordert die Auswertung mehrerer Ansichten, um Ambiguitäten, die durch gegenseitige Objektverdeckungen entstehen, auflösen zu können. In Abschnitt 4.4 wird auf dieses Forschungsgebiet eingegangen und ein Überblick zu bestehenden Kategorien klassischer Verfahren sowie Verfahren des Deep Learnings gegeben.

Ist eine 3D-Rekonstruktion (Visuelle Hülle) Bestandteil des Verfahrens, so kann ein Human Tracking oder eine Pose Estimation basierend auf 3D-Volumendaten (beispielsweise Voxeldaten) vorgenommen werden. Die rekonstruierten Volumina stellen dabei jedoch Approximationen der Geometrien dynamischer Objekte dar, wodurch sich auch Informationsverluste ergeben, insbesondere wenn wenige Kameraansichten für die Rekonstruktion verwendet werden. Das Human Tracking und die Pose Estimation basierend auf Rekonstruktionsdaten erhalten aktuell eher eine geringe Aufmerksamkeit, was möglicherweise auf die Erfolge der 2D Deep Human Pose Estimation zurückzuführen ist sowie auf den erhöhten Rechenaufwand, der typischerweise mit einer 3D-Rekonstruktion und der Auswertung von 3D-Daten einhergeht. Die rekonstruktionsbasierten Verfahren sind jedoch für das Ziel der 3D-Lokalisierung von Personen und die Generierung von 3D-Belegungsinformationen interessant und werden aus diesem Grund in Abschnitt 4.5 ausführlicher betrachtet.

Konkret rücken für die Zielstellung dieser Dissertation einer

- 3D-Lokalisierung (mittels Zustandsschätzung)
- von mehreren Personen
- in Anwesenheit statischer Objekte

die Teilbereiche in den Fokus, welche ein Multi-View-Kamerasystem verwenden (mit und ohne 3D-Rekonstruktion) und 3D-Informationen als Ausgabe generieren. Diese Bereiche sind in Abb. 4.1 schraffiert hervorgehoben und werden im weiteren Verlauf dieses Kapitels bezüglich der Aufgabenstellung bewertet.

4.3 Human Tracking

Das Ziel dieser Dissertation wurde als Personen-Tracking formuliert, was in der englischen Literatur als Human Tracking bezeichnet wird. Dafür können unterschiedlichste Verfahren eingesetzt werden, die einfache bis komplizierte Objektmodelle benutzen. Die meisten der Verfahren, die im Folgenden vorgestellt werden, sind nicht beschränkt auf das Tracking von Personen, sondern können auch für andere Objekte eingesetzt werden, was dann allgemein nur als „Tracking“ bezeichnet wird.

In dem Übersichtsartikel von [Yilmaz et al., 2006] wird Tracking in seiner einfachsten Form als das Problem der Schätzung der Trajektorie eines Objekts in der Bildebene beschrieben, während es sich im Raum bewegt. Die Position des Objekts wird in jedem Frame lokalisiert und mit einem konstanten Label versehen, sodass eine Korrespondenz zwischen den einzelnen Frames hergestellt werden kann. Neben der reinen Lokalisierung können vom Tracker zusätzliche Informationen über das zu verfolgende Objekt zur Verfügung gestellt werden, wie z. B. die Orientierung, Fläche und Form. Einfache Tracking-Algorithmen, welche direkt die Ergebnisse einer Objektdetektion und Merkmalsextraktion verarbeiten, werden in Abschnitt 4.3.1 als **Tracking im Bildraum** beschrieben.

Ein Tracking kann sehr herausfordernd sein, wodurch es nicht immer möglich ist, die zu den Objekten gehörenden Pixel und Regionen in den Bildern zu bestimmen. Probleme bereiten insbesondere komplexe Objektbewegungen, Objektverdeckungen durch andere zu trackende Objekte und Hintergrundobjekte sowie Veränderungen der Szenenbeleuchtung und Sensorrauschen [Yilmaz et al., 2006]. Aus diesem Grund wird oftmals im Zustandsraum getrackt, was insgesamt als robuster bewertet werden kann als das Tracking im Bildraum. Der Zustandsraum kann sich aus unterschiedlichen Parametern zusammensetzen, die z. B. zum Objektmodell oder zum Bewegungsmodell gehören. In Abschnitt 4.3.2 wird das **Tracking im Zustandsraum** näher erläutert.

Eng mit dem Tracking verbunden sind Verfahren der **Objektdetektion**. Bezieht sich diese auf den Menschen, so wird auch der englische Ausdruck Human Detection dafür verwendet, wie bereits beschrieben. In [Moeslund et al., 2006] wird die Objektdetektion als Bestandteil des Trackings selbst beschrieben. Die Objektdetektion dient dazu, die Objekte von Interesse von dem restlichen Bildinhalt zu separieren. Details dazu wurden bereits in Abschnitt 2.2 für Background-Subtraction-Verfahren erläutert. Die Objektdetektion in einem Frame ist erforderlich zur Track-Initialisierung sowie zur Initialisierung des gewählten Objektmodells [Yilmaz et al., 2006]. Schließlich sind die zu trackenden Objekte dem System bei Überwachungsszenarien oft a priori unbekannt, weshalb im ersten Schritt die zum Tracken notwendigen Informationen gewonnen werden

müssen. Sind die initialen Objektregionen im Bild gegeben, so ist es anschließend die Aufgabe des Trackers, die Objektkorrespondenzen zwischen den Frames herzustellen, um die Trajektorien zu erzeugen. In den beiden Übersichtsartikeln von [Moeslund et al., 2006] und [Yilmaz et al., 2006] findet man eine Reihe von Verfahren der Objektdetektion, die typischerweise beim Tracking zum Einsatz kommen, wie z. B. der Optische Fluss oder das Frame Differencing. Diese werten unterschiedliche Informationen der Objekte aus, wie Bewegungen, Aussehen, Formen oder sonstige Objekteigenschaften [Moeslund et al., 2006].

4.3.1 Tracking im Bildraum

Es gibt zahlreiche Anwendungen mit dem Ziel der Objektlokalisierung in Bildern. Diese finden sich beispielsweise in den Bereichen der Videokommunikation und -kompression, der Verkehrskontrolle und -überwachung, der medizinischen Bildgebung sowie der Videoeditierung. Derartige Tracking-Verfahren arbeiten meist mit einfachen Objektmodellen und Annahmen und werden auch als **Motion Tracking** bezeichnet [Yilmaz et al., 2006]. Oftmals genügen hierfür einfache Objektrepräsentationen mit nur wenigen Parametern. Dabei kann es sich beispielsweise um ein Punktmodell (Zentroid), ein geometrisches Primitiv wie ein Rechteck oder eine Ellipse, oder um Objektsilhouetten und -konturen handeln. Das Tracking im Bildraum basiert auf der Analyse der Pixeländerungen zwischen konsekutiven Frames und der Suche nach Objekteigenschaften in den Bildern.

Das Aussehen von Objekten, wie die Farbe und Textur, kann über Wahrscheinlichkeitsdichten, z. B. Histogramme, definiert werden. Diese lassen sich aus den Bildregionen gewinnen, die mit den gesuchten Objekten assoziiert sind. Die Auswahl der Objektrepräsentation (Objektmodell) hängt sehr stark von der Anwendungsdomäne ab, in welcher getrackt werden soll sowie von der Größe und Variation der Objektform und des Objektaussehens (engl. Appearance) in den Bildern. Die Modellierung entscheidet dabei mit über die Art der Bewegung und Deformierung, die aus den Trackingdaten erfasst oder auch durch ein Bewegungsmodell approximiert werden kann. So kann beispielsweise bei einer Punktrepräsentation nur eine Translation als Bewegung im Bild angewendet werden. Bei der Verwendung von Ellipsenmodellen hingegen können affine oder projektive Transformationen berücksichtigt werden [Yilmaz et al., 2006].

Zur Objektlokalisierung während des Trackings wird das Objektmodell mit dem Bild verglichen. Dafür werden sogenannte Merkmale (engl. Features oder Cues) extrahiert. Zum Beispiel wird Farbe als Merkmal in histogrammbasierten **Appearance-Modellen** verwendet, die das Aussehen von Objekten kodieren. Für konturbasierte Repräsentationen

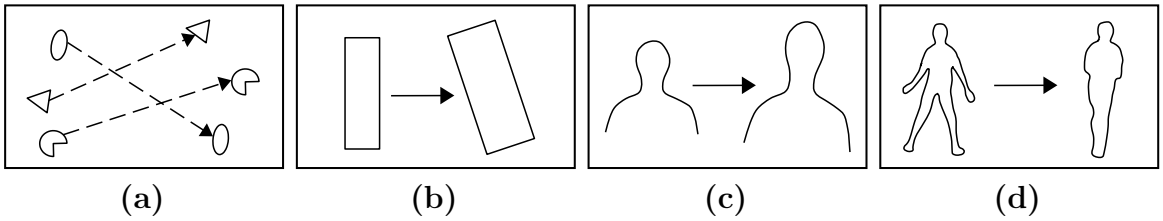


Abb. 4.3: Tracking-Ansätze für den Bildraum können verschiedenen Kategorien zugeordnet werden: Punkt-Tracking (a), Kernel-Tracking (b), silhouettenbasiertes Tracking (c) und (d). Abbildungen in Anlehnung an [Yilmaz et al., 2006, S. 17].

werden Objektkanten in den Bildern gesucht und ausgewertet. Die Wahl der Merkmale spielt für das Tracking eine zentrale Rolle und ist auch eng verbunden mit der Objektrepräsentation. Ziel der Auswahl ist die objektspezifische Eindeutigkeit, sodass Objekte leicht voneinander sowie vom Hintergrund differenziert werden können. In [Yilmaz et al., 2006] werden Objekt-Tracking-Verfahren kategorisiert nach **Punkt-Tracking**, **Kernel-Tracking** und **Silhouettenbasiertem Tracking** (engl. Contour Evolution) wie in Abb. 4.3 veranschaulicht.

Beim Punkt-Tracking werden markante Punkte in den Bildern detektiert und Korrespondenzen solcher Punkte in aufeinanderfolgenden Frames gesucht. Dies stellt oftmals ein komplexes Problem dar, aufgrund von Verdeckungen, Fehldetektionen, Beleuchtungsschwankungen, das Auftreten und Verschwinden von Objekten sowie anderen Störungen. Deterministische Methoden für die Punkt-Korrespondenz definieren eine Kostenfunktion für die Zuordnung eines Punkts in Frame $k-1$ zu einem Punkt in Frame k. Hierfür werden Bewegungsheuristiken, sogenannte **Motion Constraints**, eingesetzt. Die Minimierung der „Korrespondenzkosten“ wird als ein kombinatorisches Optimierungsproblem beschrieben. Eine Lösung wird durch **Optimal-Assignment-Methoden** gegeben, wie z. B. den populären Hungarian-Algorithmus oder Greedy-Suchmethoden (vgl. Abschnitt 6.3.2 für weitere Informationen). Dabei wird eine 1:1-Zuordnung angenommen, d. h. jedem Punkt in einem Frame wird maximal genau ein anderer Punkt im zeitlich benachbarten Frame zugeordnet.

Beim Kernel-Tracking wird ein Suchbereich, der durch das Objektmodell definiert ist, wie z. B. ein Rechteck, im Bild verschoben und gegebenenfalls parametrisch modifiziert. Die Farbwerte der Pixel innerhalb des Suchbereichs werden anschließend mit dem Appearance-Modell des Objekts verglichen. Die Parametrisierung des Suchbereichs, welche die maximale Übereinstimmung (Korrelation) liefert, bestimmt die Lokalisierung des Objekts im aktuellen Frame. Die Abtastung der Parametrisierung des Suchbereichs kann z. B. über einen Gradientenabstieg gesteuert werden. Als Ergebnis erhält man die Pixel im Bild, welche zum Objekt gehören.

Das Silhouettenbasierte Tracking wird verwendet, um kompliziertere Formen verfolgen zu können als mit dem Kernel-Tracking oder wenn vollständige Objektregionen für die Lokalisierung benötigt werden. Eine häufig eingesetzte Silhouettenrepräsentation ist die Binary-Indicator-Funktion, welche die Objektregionen mit Einsen und die anderen Regionen mit Nullen markiert. Detailliertere Beschreibungen zu verschiedenen Verfahren dieser Klasse wie dem **Shape Matching**, dem **Silhouette Matching** oder dem **Contour Tracking** können in [Yilmaz et al., 2006] nachgelesen werden.

An dieser Stelle sei erwähnt, dass die beschriebenen Verfahren aus den Übersichtsartikeln [Yilmaz et al., 2006] und [Moeslund et al., 2006] durch aktuellere Kategorisierungen ergänzt oder ersetzt werden könnten. Zudem wird häufig nicht mehr alleinig im Bildraum getrackt, sondern zusätzlich oder ausschließlich im Zustandsraum, was im nachfolgenden Abschnitt thematisiert wird. Als Einführung in das weite Gebiet des Trackings und zur Abgrenzung von der Pose Estimation, die besonders aus den anfänglichen Verfahren hervorgeht, wurde jedoch der Überblick zu den ausgewählten Verfahren gegeben, um unterschiedliche Vorgehensweisen zu verdeutlichen.

4.3.2 Tracking im Zustandsraum

Nachteilig am Tracking im Bildraum ist, dass man aus den geschätzten Bildpositionen und angepassten Modellparametern keine Informationen zur blickwinkelunabhängigen Bewegung und Pose der Objekte bzw. Personen im 3D-Raum erhält. So lassen sich Parameter wie z. B. die 3D-Position, die Bewegungsgeschwindigkeit und die Beschleunigung nicht oder nur grob daraus schätzen. Zudem führen Störungen wie Sensorrauschen, abrupte Bewegungsänderungen der Objekte, Verdeckungen und Korrespondenzprobleme oft zu einem Versagen von Schätzungen, die im Bildraum vorgenommen werden.

Zur Berücksichtigung von Mess- und Modellunsicherheiten sowie zur Bestimmung verborgener Variablen, die nicht direkt messbar sind, können Ansätze der **Zustandsschätzung** eingesetzt werden (engl. State Space Approaches). Solch ein Ansatz basiert darauf, dass ein zeitlich veränderlicher Prozess durch einen Vektor von Quantitäten beschrieben wird, wie z. B. einen Vektor aus Position, Geschwindigkeit und Beschleunigung. Diese Quantitäten gemeinsam repräsentieren den **Zustand** des Objekts im Bewegungsprozess (engl. State) und spannen den **Zustandsraum** (engl. State Space) auf. Die Historie der Entwicklung des Objektzustands über die Zeit wird als Tajektorie im Zustandsraum dargestellt. Die Zustandsschätzung ist die Suche nach der wahrscheinlichsten Parameterkombination für einen Frame. Die Bestimmung eines Trajektorienpunkts im Zustandsraum erfolgt durch den Abgleich mit den Beobachtungen aus dem Messraum, wofür häufig **Likelihood-Funktionen** zum Einsatz kommen.

Für die Umsetzung eines Trackings im Zustandsraum kann das Konzept des **Bayes-Filters** angewandt werden in Form seiner speziellen Implementierungen des **Kalman-Filters** oder des **Partikelfilters** [Arulampalam et al., 2002]. Der Bayes-Filter stellt einen probabilistischen Ansatz zur Schätzung einer unbekannten Wahrscheinlichkeitsdichtefunktion dar, die als A-Posteriori-Wahrscheinlichkeitsdichte bezeichnet wird. Die Schätzung erfolgt dabei rekursiv über die Zeit anhand der vorliegenden Messungen sowie einem mathematischen Prozessmodell. Für die Annahme, dass die Variablen linear und normalverteilt sind, also einem „Gauß'schen Rauschen" unterliegen, kann der Bayes-Filter in Form eines Kalman-Filters realisiert werden. Für nichtlineare und Nicht-Gauß'sche Prozesse lässt sich die gefilterte A-Posteriori-Dichtefunktion durch eine Menge gewichteter diskreter Punkte approximieren, die als Partikel bezeichnet werden. Isard und Blake haben hierzu den Partikelfilter, auch CONDENSATION genannt, für das visuelle Tracking eines dynamischen Systems eingeführt [Isard und Blake, 1998]. Dieser basiert auf der sequentiellen Monte-Carlo-Schätzung. Die Gewichtung der Partikel erfolgt mithilfe einer Likelihood-Funktion. Zudem werden die Partikel mit einem **Bewegungsmodell** propagiert, das häufig auch als Dynamikmodell bezeichnet wird. Die Abtastung der A-Posteriori-Dichte wird **Sampling** genannt.

Beim Kalman-Filter wird zu jedem Zeitpunkt eine unimodale Wahrscheinlichkeitsdichte geschätzt, ähnlich zu lokalen Optimierungsmethoden wie einem Gradientenabstieg oder Mean-Shift-Verfahren. Lokale Optimierer werden auch für das Tracking in der Bilddomäne, beispielsweise beim Kernel-Tracking, eingesetzt. Beim Vorliegen vieler Mehrdeutigkeiten, beispielsweise durch Objektselbstverdeckungen, ist es sehr wahrscheinlich, eine schlechte Schätzung zu erhalten. Als Resultat kann der Tracker von dem zu trackenden Objekt wegdriften und einen vollständigen Trackverlust erleiden.

Beim Partikelfilter wird ebenfalls zu jedem Zeitpunkt eine beste Schätzung bestimmt (z. B. gewichtetes Mittel aller Partikelzustände oder Zustand des Partikels mit maximalem Gewicht). Im Vergleich zum Kalman-Filter können mehrere Modalitäten der approximierten Dichtefunktion von den Partikeln repräsentiert und beim Tracking verfolgt werden. Sowohl der Kalman-Filter als auch der Partikelfilter sind populäre Ansätze mit verschiedenen Vor- und Nachteilen, für die zahlreiche Erweiterungen entwickelt wurden.

4.4 Human Pose Estimation

Für verschiedene Anwendungen ist ein Tracking im Bildraum oder in einem Zustandsraum niedriger Dimensionalität ausreichend, um einfache Bewegungsbeschreibungen zu erhalten und Objekte bzw. Personen im Bild oder im 3D-Raum weitestgehend

lokalisieren zu können. Für andere Anwendungen hingegen werden detaillierte Poseninformationen von Personen benötigt, was beispielsweise durch die Schätzung von 3D-Gelenkpositionen erreicht werden kann. Damit lassen sich mehr Details zu den Bewegungen von Personen gewinnen.

4.4.1 Körpermodell

Für die Posenschätzung können modellfreie Verfahren eingesetzt werden, die ohne Körpermodell menschliche Poseninformationen als Ausgabe erzeugen (z. B. bei der Deep Human Pose Estimation [Zheng et al., 2020]). Klassische Verfahren [Sarafianos et al., 2016] arbeiten jedoch häufig mit einem genauen und komplizierten Körpermodell.

Das Körpermodell (engl. Human Body Model) kann in Form eines kinematischen Baums vorliegen [Ikeuchi, 2014], genannt Skelettmodell (engl. Skeleton oder Stick Figure). Als Wurzelsegment (engl. Root) wird häufig das Becken verwendet, dessen Position und Rotation im Weltkoordinatensystem gegeben sind. Dazu wird je Körperteil eine Menge von relativen Winkeln angegeben, um die Orientierung des Körperteils bezüglich der jeweiligen Eltern entlang des Baums bestimmen zu können, wie z. B. die Lage des Oberschenkels bezüglich des Beckens. Kinematische Bäume können für 2D-, 2,5D- und 3D-Modelle verwendet werden (vgl. Abb. 4.4). Nach [Poppe, 2007] sind 2D-Modelle nur für Bewegungen parallel zur Bildebene geeignet, z. B. für Ganganalysen. Komplexere Bewegungen können damit nicht vollständig erfasst werden. Die 2,5D-Repräsentationen stellen eine Erweiterung der 2D-Modelle um Variablen dar, welche die relative Tiefe der Körperteile zueinander angeben. Dies wird auch als „Layering“ bezeichnet.

Die Repräsentation von Skelettmodellen resultiert typischerweise in einem hochdimensionalen Posenvektor mit einer Anzahl an Freiheitsgraden DOFs (engl. Degrees of Freedom) von zirka $\mathbb{R}^{30}$ bis $\mathbb{R}^{70}$ [Poppe, 2007]. Die Anzahl möglicher Posen ist damit ziemlich hoch (im Vergleich zu einer geringeren Anzahl an DOFs wie z. B. 10), selbst für ein Modell mit begrenzter Anzahl an DOFs je Gelenk sowie einer eher geringen Auflösung im diskreten Parameterraum. Die Anwendung von kinematischen Randbedingungen, genannt **Constraints**, ist ein effektiver Weg, um den Parameterraum, auch Konfigurationsraum (engl. Pose Space) genannt, zu „beschneiden“. Dadurch können unplausible Posen eliminiert werden. Typische Constraints sind die Grenzen der Gelenkwinkel sowie Grenzen der Winkelgeschwindigkeit und Winkelbeschleunigung [Poppe, 2007].

Zusätzlich zu den kinematischen Eigenschaften des Körpers wird auch die Form (engl. Shape) und das Aussehen (engl. Appearance) des Menschen modelliert. Segmente von 2D-Modellen werden häufig als rechteckige oder trapezförmige „Patches“ beschrieben. Für dreidimensionale Skelettrepräsentationen werden die zugehörigen Segmente entweder

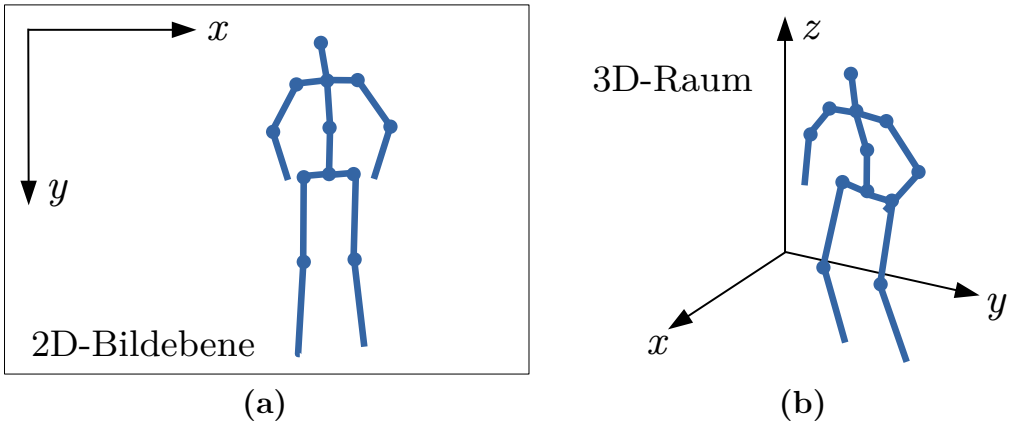


Abb. 4.4: Skelettale kinematische Bäume für die Menschrepräsentation bei der Pose Estimation im Zweidimensionalen (a) und im Dreidimensionalen (b).

volumetrisch modelliert, beispielsweise durch Kugeln, Zylinder und Superellipsoide, oder oberflächenbasiert, z. B. durch ein Punktgitter, das aus Polygonen besteht (engl. Mesh).

4.4.2 Klassische Verfahren

Eine Übersicht zu überwiegend klassischen Verfahren der Pose Estimation, die zwischen 2008 und 2015 veröffentlicht wurden, ist in [Sarafianos et al., 2016] zu finden. Es wird beschrieben, dass solche Verfahren typischerweise aus den in Abb. 4.5 dargestellten Einzelschritten bestehen, wobei nicht immer eine Vollständigkeit vorliegen muss. Die Einzelschritte aus Abb. 4.5 werden nun kurz erläutert. Als Eingabe für die Posenschätzung einer Kamera zu einem Zeitpunkt wird entweder ein Einzelbild oder aber eine Videosequenz verwendet (engl. Input Image/Video). Bestandteil sogenannter modellbasierter Verfahren ist ein Körpermodell wie ein Skelettmodell, das meist a priori erzeugt wird (engl. Body Modeling). Auf die Eingabebilder kann eine Vorverarbeitstechnik (engl. Preprocessing) wie beispielsweise ein Background Subtraction angewandt werden, um Regionen von Interesse zu detektieren (Objektdetektion). Weiterhin werden Methoden der Merkmalsextraktion und -selektion benötigt (engl. Feature Extraction), um Merkmale aus den Bildern gewinnen zu können, die signifikante Informationen für den Schätzalgorithmus liefern. In Abb. 4.7 werden einige Features gezeigt. In [Sarafianos et al., 2016] wird eine aktuellere Übersicht zu häufig gewählten Features für die Pose Estimation gegeben.

Basierend auf den Merkmalen kann zunächst eine 2D-Posenschätzung (engl. 2D Pose Estimation) generiert werden, mit deren Hilfe eine Initialisierung der gesuchten 3D-Pose vorgenommen wird (engl. Pose Initialization), mit dem Ziel, den Suchraum im Vorfeld einzuschränken. Basierend auf der initialen 3D-Pose wird die finale 3D-Pose

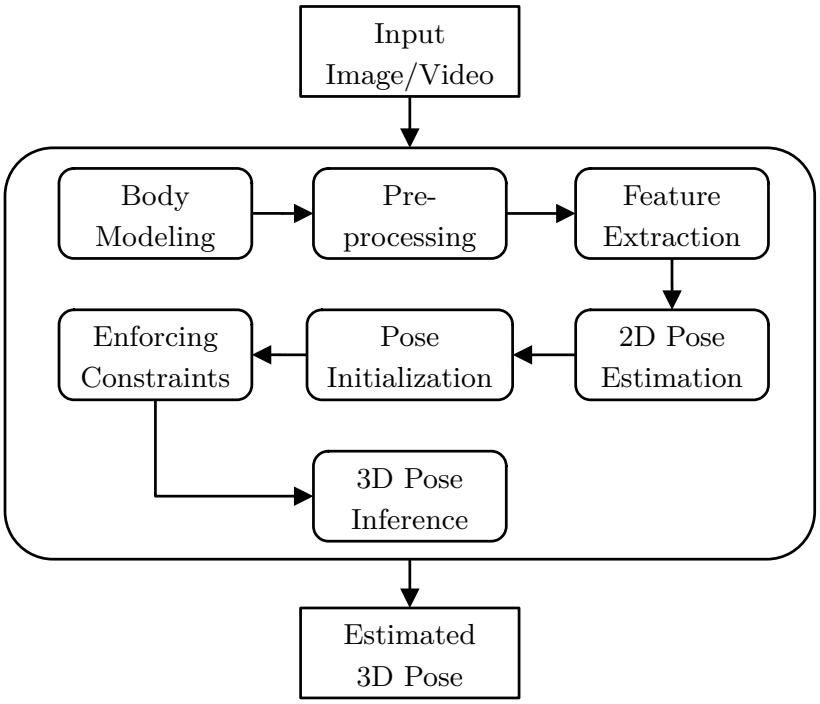


Abb. 4.5: Allgemeine Schritte eines klassischen Verfahrens der 3D Human Pose Estimation. Gegeben ein Einzelbild oder eine Videosequenz wird die 3D-Pose durch die Anwendung einiger oder aller der dargestellten Schritte geschätzt. Abbildung in Anlehnung an [Sarafianos et al., 2016, S. 3].

mit einer Optimierungstechnik geschätzt (engl. 3D Pose Inference). Dabei werden Randbedingungen berücksichtigt (engl. Enforcing Constraints), um anthropometrisch unrealistische Posen zu verwerfen. Die Ausgabe des Verfahrens ist eine geschätzte 3D-Pose (engl. Estimated 3D Pose), beispielsweise in Form von Gelenkpositionen im 3D-Raum. Die geschätzten 2D-Posen können als Gütemaß eingesetzt werden, um die Genauigkeit geschätzter 3D-Posen zu bewerten. Dazu wird eine 3D-Pose in das zugehörige 2D-Bild projiziert und mit der geschätzten 2D-Pose auf Übereinstimmung geprüft.

Verfahrensklassen der Pose Estimation, welche die genannten Schritte auf unterschiedliche Weise implementieren, werden in Abb. 4.6 gezeigt (Mitte). Es handelt sich dabei um die **Generativen Methoden** (engl. Generative Methods), zu denen auch die sogenannten **Körperteilbasierten Methoden** (engl. Part-based Methods) gehören. Weiterhin gibt es die **Diskriminativen Methoden** (engl. Discriminative Methods), die sich in die **Lernbasierten Methoden** (engl. Learning-based Methods) sowie die **Beispielbasierten Methoden** (engl. Example-based Methods) unterteilen. Es gibt darüber hinaus hybride Verfahren, bestehend aus einer generativen und einer diskriminativen Komponente. Wie ein Verfahren konkret konzipiert ist, hängt auch von den Eingabedaten ab, die es verarbeitet. Dabei kann es sich um Einzelbilder oder Videosequenzen handeln, die von einer Kamera (monokular) oder einem Multi-View-Kamerasystem

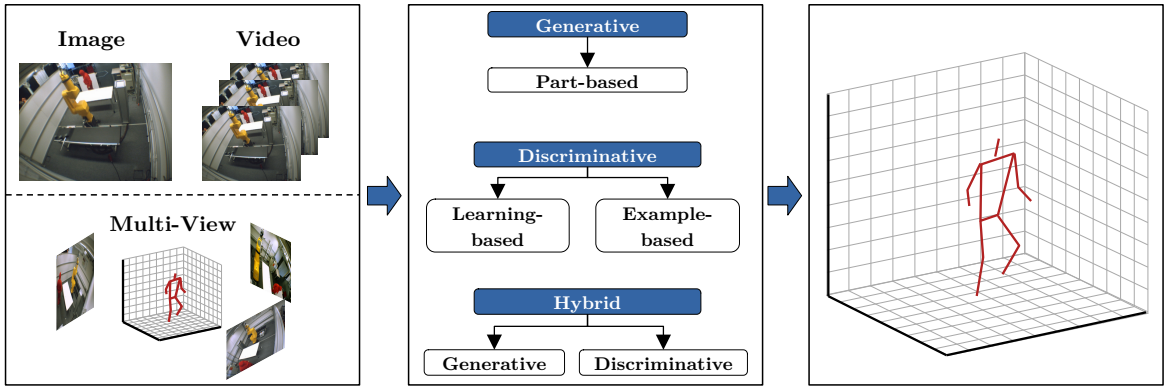


Abb. 4.6: Taxonomie von klassischen Verfahren der 3D Pose Estimation. Als Eingabe ist gegeben ein Bild oder Video eines monokularen oder Multi-View-Kamerasystems. Es wird unterschieden zwischen generativen Methoden (mit der Unterkategorie der Part-based Methoden), diskriminativen Methoden (lernbasierte und beispielbasierte Methoden) sowie hybriden Ansätzen bestehend aus einer generativen und einer diskriminativen Komponente. Abb. in Anlehnung an [Sarafianos et al., 2016, S. 4].

aufgenommen werden, wie in Abb. 4.6 (links) zu sehen. In [Sarafianos et al., 2016] werden ausgewählte Verfahren detaillierter vorgestellt und bewertet und entsprechend ihrer Schlüsselcharakteristiken in die genannten Kategorien eingeordnet. Auf eine Wiederholung der Beschreibungen zu den Einzelverfahren wird an dieser Stelle verzichtet. Stattdessen werden die genannten Verfahrensklassen noch genauer beschrieben. Die nachfolgenden Informationen dieses Abschnitts entstammen überwiegend [Ikeuchi, 2014]. Ergänzungen durch andere Referenzen sind explizit gekennzeichnet.

Nach [Ikeuchi, 2014] besteht das Ziel der Posenschätzung darin, die Bestimmung der plausibelsten Konfiguration einer Person im Bild bzw. die Schätzung der A-Posteriori-Wahrscheinlichkeitsdichtefunktion $p(\mathbf{x}|\mathbf{z})$ vorzunehmen. Mit $\mathbf{x}$ ist dabei die Körperpose definiert und mit $\mathbf{z}$ die Beobachtung, die aus extrahierten Bildmerkmalen besteht. Für dieses Ziel können die Generativen Methoden eingesetzt werden, die synonym auch als **Top-Down-Ansätze** oder in [Poppe, 2007] als **Modellbasierte Methoden** bezeichnet werden. Bei den generativen Methoden wird die gewünschte A-Posteriori-Dichte $p(\mathbf{x}|\mathbf{z})$ als Produkt aus einer Likelihood- und einer A-Priori-Dichte ausgedrückt, wie in Formel (4.1) angegeben.

$$p(\mathbf{x}|\mathbf{z}) \propto \underbrace{p(\mathbf{z}|\mathbf{x})}_{likelihood} \cdot \underbrace{p(\mathbf{x})}_{prior} \tag{4.1}$$

Die Charakterisierung einer hochdimensionalen A-Posteriori-Dichte ist typischerweise schwer. Deshalb basieren die meisten Ansätze auf Maximum-A-Posteriori-Lösungen (MAP). Diese suchen nach den Konfigurationen, welche die Bilddaten gut beschreiben und damit zu einer hohen Likelihood führen und zugleich eine hohe A-Priori-

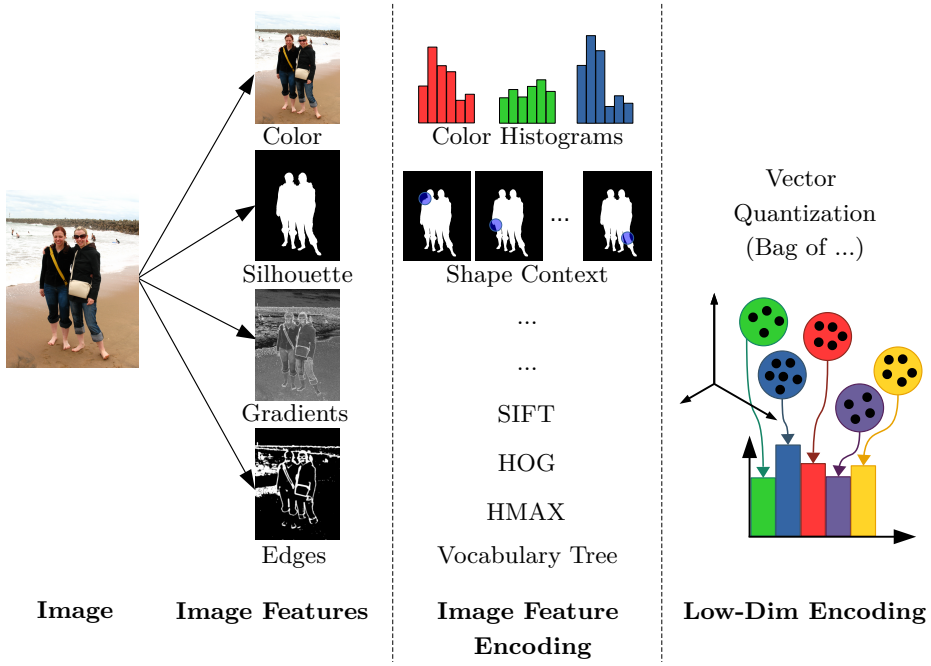


Abb. 4.7: Allgemeine Bildmerkmale und Enkodierungstechniken für die Pose Estimation, in Anlehnung an [Ikeuchi, 2014, S. 364]. Die Performanz jedes Pose-Estimation-Ansatzes hängt stark von den verwendeten Beobachtungen bzw. Bildmerkmalen und ihrer Kodierung ab. Aus dem gesamten Bild (Image) werden Bildmerkmale berechnet wie beispielsweise Silhouetten, Kanten, Bewegungsgradienten und Farben (Image Features). Passende Enkodierungstechniken, auch genannt Deskriptoren, sind beispielsweise Farbhistogramme oder SIFT-Merkmale (Image Feature Encoding). Manche Ansätze führen eine Dimensionsreduktion durch, was jedoch mit Verlusten wie beispielsweise der räumlichen Struktur einhergehen kann (Low-Dim Encoding).

Wahrscheinlichkeit mitbringen (vgl. Formel (4.2)).

$$\mathbf{x}_{\text{MAP}} = \text{argmax}\ p(\mathbf{x}|\mathbf{z}) \tag{4.2}$$

Zur Berechnung eines Likelihood-Werts wird die Distanz zwischen dem Bild bzw. Bildmerkmalen und einer projizierten synthetischen Modellinstanz bestimmt. Es wird nach der Posenkonfiguration gesucht, welche die Distanz und damit den Fehler minimiert. Diese Vorgehensweise wird in der Literatur mitunter auch als **Analysis By Synthesis Approach** bezeichnet [Poppe, 2007]. Zur Bestimmung der Likelihood kommen Matching-Funktionen zum Einsatz. Die Wahl dieser hängt von den gewählten Bildmerkmalen (engl. Features) ab. In Abb. 4.7 sind mehrere Features und Enkodierungstechniken dargestellt. Bei der Verwendung von projizierten Silhouetten der Modellinstanzen kann z. B. die Überlappung zu Silhouetten berechnet werden, die aus dem Bild extrahiert werden. Bei Kanten wird häufig eine Distanz zwischen den synthetischen Kanten des Modells und den jeweils nahesten Kanten, die im Bild gefunden werden, berechnet. In [Poppe, 2007] wird beschrieben, dass eine Likelihood-Funktion robuster sein kann, wenn mehrere Deskriptoren verwendet werden, das heißt, wenn eine Kombination

mehrerer Matching-Funktionen eingesetzt wird. Jedoch sollte die Wahl mit Bedacht erfolgen, da es nicht ungewöhnlich ist, dass eine Parametrisierung, welche einen guten Match bzw. Fitwert für einen Deskriptor liefert, ebenfalls einen guten Match für einen anderen Deskriptor erzeugt. Wenn die Fitwerte einfach miteinander multipliziert werden, so kann dies zu scharfen Spitzen in der Likelihood-Funktion führen, was wiederum in einer weniger effektiven Schätzung resultieren kann.

Bei den generativen Ansätzen erfolgt die Suche nach der besten Posenkonfiguration oft in hochdimensionalen Räumen (40+), was entsprechend herausfordernd ist. Hierbei besteht die Gefahr des Konvergierens in lokalen Optima. Globale hierarchische Suchverfahren, wie z. B. der Annealed Partikelfilter [Deutscher et al., 2000], haben Erfolge für einfache Skelettmodelle gezeigt, wobei der Körper dabei meistens aufgerichtet und für mehrere Kameras sichtbar ist.

Eine Unterkategorie der generativen Methoden ist die der körperteilbasierten Methoden. Charakteristisch für diese ist, dass die Modellierung des menschlichen Körpers nicht als kinematische Baumstruktur sondern als eine Menge von Körperteilen erfolgt, die zusammengesetzt werden müssen. Jedes Körperteil besitzt eine eigene Position und Orientierung im Raum und ist mit den anderen Körperteilen über statistische oder physikalische Constraints miteinander verbunden. Durch die voneinander unabhängige Parametrisierung der Körperteile treten Redundanzen im Parameterraum auf, die in einer höheren Anzahl an Dimensionen resultieren. Die einzelnen Körperteile können in 2D, 2,5D oder 3D definiert werden, wobei 2D-Modelle bisher vermutlich am häufigsten vorkommen.

Das Prinzip der körperteilbasierten Methoden ist das Folgende: In den Bildern wird nach einzelnen wahrscheinlichen Körperteilregionen unabhängig voneinander gesucht, nur unter Berücksichtigung benachbarter Körperteile durch die Constraints. So können Körperteildetektoren wie Gesichtsdetektoren, Kopfdetektoren, Kopf-Schulter-Partie-Detektoren und weitere eingesetzt werden, um die Körperteile im Bild zu finden. Die gefundenen Teile werden anschließend zu einem einzigen Körpermodell zusammengesetzt, womit die Parametrisierung des Modells erzeugt wird. Diese Vorgehensweise wird z. B. in [Poppe, 2007] als **Bottom-Up-Ansatz** bezeichnet und ist das Gegenstück zu den generativen Top-Down-Ansätzen. Der Vorteil dieses Prinzips liegt in der Reduktion der Inferenzkomplexität. Kaskaden von Körperteildetektoren können die Performanz verbessern und die Inferenz beschleunigen, trotz der gegebenenfalls höheren Dimensionalität des Parameterraums. So können schnelle Likelihoods für die einzelnen Detektoren bereits einen Großteil des Suchraums eliminieren, bevor im Nachgang komplexere und rechnerisch aufwändigere Likelihood-Funktionen für die Gesamtkonfiguration angewendet werden. Der traditionellste und erfolgreichste Ansatz ist die Repräsentation des

Körpers durch ein Markov Random Field [Ikeuchi, 2014]. Problematisch sind jedoch verdeckte Teile, die nicht von den Körperteildetektoren gefunden werden können.

Die dritte Klasse der Inferenztechniken für die Pose Estimation ist die der diskriminativen Methoden, die auch als **modellfreie Ansätze** bezeichnet werden [Poppe, 2007]. Sie lernen die A-Posteriori-Dichte basierend auf Trainingsbeispielen aus annotierten (mit Labeln versehenen) Datensätzen mit zugehörigen Bildern, die z. B. künstlich erzeugt wurden. Mithilfe der gelernten Abbildungsfunktion des Bildraums auf den Posenraum ermittelt ein Klassifikator anschließend die Pose, die zu einer Beobachtung gehört. Die Inferenz ist eine Form von probabilistischer Regression. Es wird eine Regressionsfunktion gelernt, wie z. B. eine lineare Regression, eine nichtparametrische Nearest-Neighbor- oder eine Kernel-Regression. Die Verfahren sind sehr effektiv, wobei die nichtparametrischen Methoden besser mit den komplexen nichtlinearen Beziehungen zwischen verschiedenen Bildmerkmalen und den Posen umgehen können. Allerdings sind sowohl die Modell- als auch die Inferenzkomplexität Funktionen der Trainingsdatengröße. In der Praxis sind die nichtparametrischen Methoden damit langsamer als die parametrischen. In [Poppe, 2007] werden diese sogenannten lernbasierten Verfahren von den beispielbasierten Verfahren unterschieden, die das Lernen einer Mapping-Funktion vermeiden. Stattdessen wird eine große Anzahl an Beispielen in einer Datenbank zusammen mit ihrer zugehörigen Posenbeschreibung abgelegt. Für jedes Bild wird bei der Posenschätzung eine Ähnlichkeitssuche anhand von Bildmerkmalen durchgeführt. Die gefundenen Kandidatenposen werden interpoliert, um die gewünschte Posenschätzung zu erhalten.

Hybride Verfahren kombinieren diskriminative und generative Methoden miteinander. Hierbei werden Beobachtungs-Likelihoods zur Verifizierung von Posenhypothesen verwendet, die man von den diskriminativen Mapping-Funktionen erhält [Wang et al., 2021].

In [Ikeuchi, 2014] wird das folgende Resümee zu den klassischen Verfahren der Pose-Estimation gezogen: Die Körperposenschätzung ist immer noch ein weitgehend ungelöstes Problem. Fortschritte wurden bisher bei größtenteils unverdeckten und isolierten Personen erzielt. Die bestehenden Probleme betreffen mehrere potentiell interagierende Personen sowie die mangelnde Toleranz gegenüber unerwarteter Verdeckungen.

Mit den Bottom-Up-Methoden konnte bis 2014 der größte Erfolg erzielt werden. Dies betrifft sowohl die diskriminativen als auch die körperteilbasierten Ansätze. Trotz ihrer Popularität haben die diskriminativen Methoden jedoch ihre Grenzen. Zum einen können sie nicht direkt 3D-Positionen im Raum bestimmen. Dies würde riesige Datenmengen benötigen, welche das gesamte sichtbare 3D-Volumen aus Sicht der Kamera abdecken

müsste. Zum anderen bleibt die Generalisierungsfähigkeit der lernbasierten Verfahren ein Problem durch die entstehende Divergenz der Dichten aus Test- und Trainingsdaten. Überdies bleibt das effiziente Lernen diskriminativer Modelle aus riesigen Datenmengen eine Herausforderung, die sich jedoch nicht vermeiden lässt, wenn man eine Menge realistischer Aktivitäten und Posen abdecken möchte.

Das Top-Down-Prinzip kann nützlich sein, um eine globale Posenkonsistenz zu erzwingen. Es ist wahrscheinlich, dass die Kombination aus einer Top-Down- und Bottom-Up-Inferenz zu einem größeren Erfolg führen kann als die Inferenz in nur eine Richtung. Ein dahingehendes Beispiel aus [Andriluka et al., 2010] ist vielversprechend. Frühere Ansätze, die hierarchische Modelle verwenden [Zhang et al., 2006] können auch sinnvoll für die Weiterentwicklung der Pose Estimation sein. Es ist auch zu erkennen, dass zunehmend Informationen über die Zeit aggregiert werden [Andriluka et al., 2010], um die Performanz gegenüber der separaten Einzelbildschätzung zu verbessern und um eine Unterdrückung des Rauschens bei aufeinanderfolgenden Schätzungen zu erzielen. Das Ausnutzen aller möglichen Quellen von Vorwissen, wie z. B. zur Beschaffenheit des Körpers, ist entscheidend für die Weiterentwicklung der (3D-)Körperposenschätzung.

Im nachfolgenden Abschnitt wird ein neueres Forschungsgebiet der Pose Estimation vorgestellt, dessen Ansätze in Teilen auch Konzepte der klassischen Methoden (insbesondere der modellfreien Methoden) anwenden. Das Herausstellungsmerkmal ist jedoch der Einsatz spezieller Techniken des Deep Learnings.

4.4.3 Deep 3D Human Pose Estimation

Vorangetrieben durch leistungsfähige Deep-Learning-Techniken sowie kürzlich gesammelte umfangreiche Datensätze hat sich die Pose Estimation in der letzten Dekade stark weiterentwickelt (vgl. [Wang et al., 2021], [Zheng et al., 2020]), insbesondere die Schätzung basierend auf monokularen Bildern (vgl. [Chen et al., 2020]).

In Abbildung 4.8 ist eine Einordnung von Verfahren des Deep Learnings aus [Wang et al., 2021] zu sehen, die kurz erläutert werden soll, um einen Überblick über den Forschungsbereich der Deep 3D Human Pose Estimation zu geben. Dabei wird wie in [Sarafianos et al., 2016] zwischen den Eingabedaten, die verarbeitet werden, unterschieden: monokulare Einzelbilder oder Videosequenzen, für eine einzelne Perspektive oder mehrere. Weiter wird bezüglich der Posenrepräsentation danach differenziert, ob die Pose durch ein Skelett oder eine Kontur (engl. Shape) repräsentiert wird. Für die konturbasierten Repräsentationen werden parametrische Modelle wie SCAPE, SMPL und DensePose eingesetzt, um die Körperform zu ergänzen. Details dazu sind [Wang et al., 2021] zu entnehmen. Methoden, die eine Abbildung der Eingabebilder direkt auf

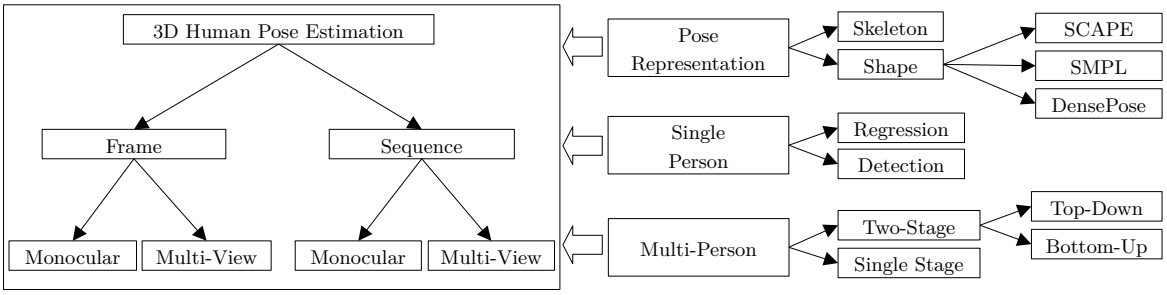


Abb. 4.8: Taxonomie der Deep 3D Human Pose Estimation, in Anlehnung an [Wang et al., 2021, S. 3].

3D-Gelenkpositionen vornehmen, können als regressionsbasiert oder detektionsbasiert kategorisiert werden. Die Human Pose Estimation ist im Wesentlichen ein Regressionsproblem, bei welchem direkt die Lage der Gelenke relativ zum Wurzelgelenk geschätzt wird. Eine Lösung dieses Problems zu finden, ist das Ziel regressionsbasierter Methoden. Detektionsbasierte Methoden sagen im Vergleich dazu eine **Likelihood Heatmap** für jedes Gelenk vorher. Der Ort eines Gelenks wird dann durch die maximale Likelihood in der Heatmap bestimmt.

Existierende Ansätze, die eine 3D-Posenschätzung mehrerer Personen vornehmen, werden in [Wang et al., 2021] danach unterschieden, ob sie einstufig (engl. Single Stage) oder zweistufig (engl. Two Stage) arbeiten. Die einstufigen Verfahren schätzen üblicherweise die Position des Wurzelgelenks und die Verschiebungen der anderen Gelenke dazu gleichzeitig. Die zweistufigen Verfahren werden weiter unterteilt, danach ob sie ein Top-down- oder ein Bottom-up-Prinzip verfolgen. Bei den Top-Down-Ansätzen wird in einem ersten Schritt die Person geschätzt und in einem zweiten Schritt ihre individuellen Gelenke. Bei den Bottom-up-Ansätzen hingegen werden zunächst die Körpergelenke einzeln geschätzt und danach einer entsprechenden Person zugewiesen.

Für die öffentlich zugänglichen Benchmark-Datensätze Human3.6M, HumanEva, und MPI-INF-3DHP wurden zuletzt gute Ergebnisse erzielt [Wang et al., 2021]. Für das Schätzen mehrerer Personen sind die einstufigen Methoden jedoch noch wenig präsent, was nach [Wang et al., 2021] ein Indiz dafür ist, dass die 3D Human Pose Estimation in Realwelt-Szenarios noch weit davon entfernt ist, etabliert zu werden. Ein Hauptgrund, der generell für den Mangel an Generalisierungsfähigkeit der Verfahren beim Einsatz in Realweltszenarios angeführt wird, ist der Mangel an „In-the-wild“-Datensätzen für das Training (mit 3D-Ground-Truth-Informationen). Darauf wird im nachfolgenden Abschnitt 4.4.4 konkreter eingegangen.

Die Anwendung von Deep Learning zur Lösung der Aufgabe der Human Pose Estimation wird in Gänze als sehr effektiv bewertet. Deshalb werden in den nächsten Jahren viele

Innovationen erwartet, einhergehend mit der Weiterentwicklung von Deep-Learning-Ansätzen und deren Einsatz in dem betrachteten Forschungsbereich.

4.4.4 Herausforderungen

Die Pose Estimation bleibt ein schwieriges und weitgehend ungelöstes Problem [Ikeuchi, 2014], auch wenn Fortschritte insbesondere durch das Deep Learning und für die 2D-Posenschätzung erzielt wurden (vgl. [Wang et al., 2021], [Zheng et al., 2020], [Chen et al., 2020]). Dafür werden in [Ikeuchi, 2014] folgende Gründe benannt:

- Variabilität des menschlichen Aussehens in den Bildern,
- Variabilität der menschlichen Physis und der vollzogenen Bewegungen,
- Informationsverlust bei der Abbildung der 3D-Welt auf die Bildebene,
- Selbstverdeckungen sowie Verdeckungen durch weitere Objekte in der Szene,
- Variabilität der Beleuchtung,
- Komplexität der menschlichen skelettalen Struktur,
- Hohe Dimensionalität der parametrischen Menschmodelle.

Die Beziehung zwischen einer Pose und der Beobachtung im Bild ist mehrdeutig in beide Richtungen. Es liegen Ambiguitäten vor, welche die Suche nach der besten Pose erschweren. So kann ein und dieselbe Pose zu vielen unterschiedlichen Beobachtungen führen, aufgrund von: Variationen des Aussehens und der Form von Personen, unterschiedlichen Blickrichtungen der Kameras, unterschiedlichen Umgebungen sowie verschiedenen Beleuchtungssituationen. Anders herum können verschiedene Posen in einer gleichen Beobachtung resultieren, was in Abb. 4.9 veranschaulicht wird. Da die Beobachtung eine Projektion der realen Welt auf die Bildebene ist, gehen zwangsläufig Informationen verloren. Wenn nur ein einzelner Sensor verwendet wird, können Selbstverdeckungen und Tiefenambiguitäten entstehen. Damit können kleine Änderungen der Pose in der Beobachtung unbemerkt bleiben. In [Sminchisescu und Triggs, 2003] wird geschätzt, dass grob ein Drittel aller Freiheitsgrade der Gelenke eines Skelettmodells aus Sicht einer Kamera fast unbeobachtbar sind. Dies bezieht sich hauptsächlich auf Bewegungen in die Tiefenrichtung, aber auch auf Rotationen von annähernd zylindrischen Gliedmaßen um ihre eigene Achse. Diese Einschränkungen können zum Teil kompensiert werden, wenn mehrere Kameras Verwendung finden [Poppe, 2007].

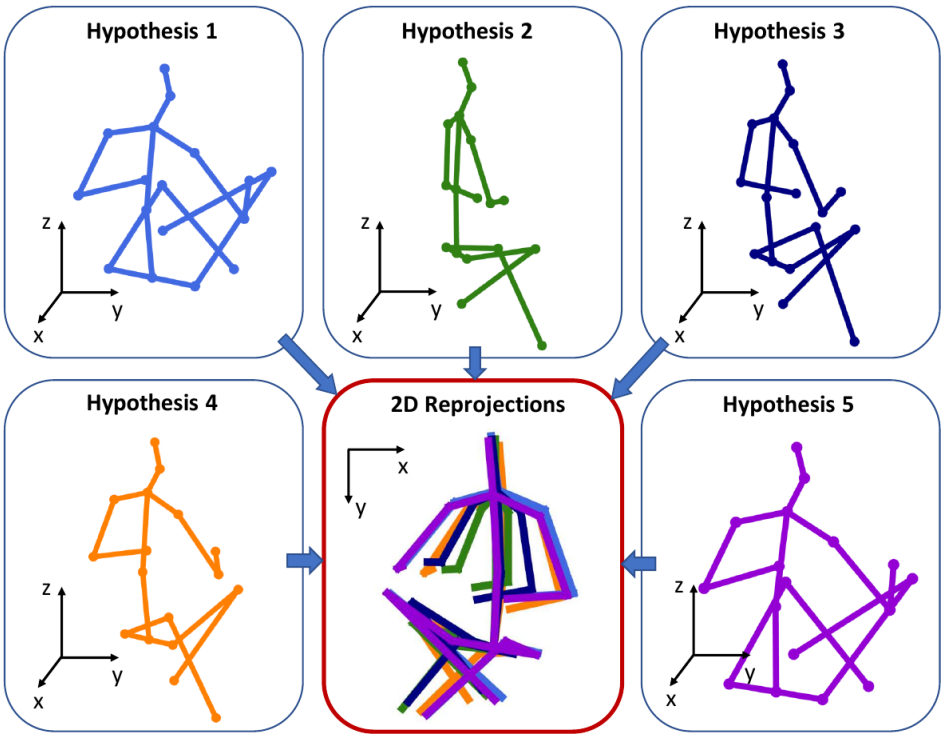


Abb. 4.9: Illustration von Tiefenambiguitäten bei der Pose Estimation, entnommen aus [Li und Lee, 2019, S. 9887].

Befinden sind allerdings noch weitere Objekte im Raum, so können die Ambiguitäten verstärkt werden, da zusätzlich zu den Selbstverdeckungen weitere Körperregionen im Bild verdeckt werden. Zudem werden Ambiguitäten erzeugt, wenn sich das Aussehen der einzelnen Körperteile nicht ausreichend voneinander unterscheidet und diese in den Bildern nicht voneinander separiert werden können.

Wie bereits erwähnt, werden die Verfahren der Pose Estimation für die Verarbeitung unterschiedlicher Eingabedaten konzipiert (vgl. Abb. 4.6):

- Monokulares Einzelbild,
- Monokulare Videosequenz,
- Multi-View Einzelbilder,
- Multi-View Videosequenzen.

Abhängig von den zur Verfügung stehenden Eingabedaten sind jeweils verschiedene der oben genannten Herausforderungen spezifisch. In [Wang et al., 2021] und anderen Arbeiten ist dazu Folgendes zu finden:

Monokulares Einzelbild: Die Generierung einer 3D-Pose aus einem monokularen Einzelbild ist ein prinzipiell unterbestimmtes Problem, da wie oben beschrieben, ähnliche

Bildprojektionen von komplett unterschiedlichen 3D-Posen hervorgerufen werden können. In solchen Fällen sind Selbstverdeckungen ein übliches Phänomen. Die resultierenden Ambiguitäten halten existierende Techniken davon ab, adäquat zu funktionieren. Zudem können kleine Fehler in der Lokalisierung von 2D-Körperteilen große Konsequenzen im 3D-Raum mit sich bringen.

Monokulare Videosequenz: Bei der Generierung einer 3D-Pose aus einer Sequenz monokularer Bilder kann das Erzwingen temporaler Konsistenz zu besseren Schätzergebnissen führen. Die zeitlichen Informationen ermöglichen eine Reduktion der Tiefenambiguitäten sowie die Schätzung exakterer Posen bei Selbstverdeckungen. Dennoch besteht neben der erhöhten Dimensionalität eine Schwierigkeit darin, damit umzugehen, dass sich Form und Aussehen des menschlichen Körpers über die Zeit in der Sequenz drastisch verändern können, aufgrund von Hintergrundänderungen oder Kamerabewegungen, Beleuchtungsänderungen, Rotationen von Gliedmaßen in die Tiefe sowie Kleidung, die nicht eng anliegt.

Multi-View Einzelbilder: Die 3D-Posenschätzung aus Einzelbildern eines Multi-View-Kamerasystems wird insbesondere eingesetzt, wenn 3D-Posen mehrerer Personen geschätzt werden sollen bzw. sich mehrere Personen gleichzeitig im betrachteten Überwachungsraum befinden. Hierbei stellen die unbekannten Korrespondenzen der Personen in unterschiedlichen Kameraperspektiven eine Herausforderung dar. Neben Selbstverdeckungen können Körperteile von anderen Personen oder Objekten der Szene verdeckt werden, falls solche präsent sind. Interaktionen, die Personen miteinander und mit Objekten durchführen, können die Verdeckungs- und Schätzproblematik weiter verschärfen. Weiterhin ergibt sich durch die Anzahl an Posen, die gleichzeitig für mehrere Individuen geschätzt werden sollen eine Erhöhung der Dimensionalität des Zustandsraums, die in irgendeiner Form überwunden werden muss.

Multi-View Videosequenzen: Für die 3D-Posenschätzung aus Videosequenzen eines Multi-View-Kamerasystems ergibt sich eine Kombinatorik der bereits beschriebenen Herausforderungen. Dennoch können Informationen, die sich sowohl aus zeitlichen Zusammenhängen als auch aus verschiedenen Perspektiven auf den Überwachungsraum gewinnen lassen zukünftig die 3D-Posenschätzung stark verbessern, insbesondere in komplexen Situationen mit mehreren Personen und weiteren Objekten.

Mangel an „In-the-wild“-Datensätzen: In [Wang et al., 2021] wird verdeutlicht, dass der Mangel an Datensätzen, die in „freier Wildbahn“ erzeugt wurden einen echten Flaschenhals für die Forschung der 3D-Posenschätzung darstellt. Damit sind Datensätze z. B. von Multi-View-Kamerasystemen gemeint, die nicht in künstlichen Umgebungen oder mit geplanten Bewegungen generiert wurden. Insbesondere Videosequenzen im Freien liefern verschiedenste Beleuchtungs- und Hintergrundbedingungen sowie oft ungezwungene

Bewegungen. Das Entscheidende bei den Datensätzen ist das Vorliegen von Ground-Truth-Werten für die 3D-Posen, die geschätzt werden sollen und beispielsweise bei Lernverfahren als Teach-Werte eingesetzt werden können. Für die 2D-Posenschätzung ist es möglich, selbst größere „In-the-wild"-Datensätze zu erstellen, indem manuell die 2D-Posen in den Bildern annotiert werden. Hingegen ist das Hinzufügen von 3D-Annotationen zu Datensätzen aus Bildern, die nur 2D-Projektionen der Realität darstellen eher schwierig. Meist werden die Datensätze durch markerbasierte Motion-Capture-Systeme erzeugt, wie beispielsweise HumanEva und Human3.6M. Dies ist sehr ressourcenintensiv (Hardware-Ausrüstung), insbesondere wenn größere Datensätze generiert werden sollen. Neuere markerlose Motion-Capture-Systeme ermöglichen weniger Einschränkungen der Akteure, sind aber aktuell ebenfalls ressourcenintensiv.

Verfahren, die mit Datensätzen trainiert werden, weisen ohne die Verwendung vielseitiger Datensätze meist ein Generalisierungsproblem auf. Das heißt, sie führen beim Einsatz unter veränderten Bedingungen nur bedingt zu der gewünschten Performanz. Laut Wang [Wang et al., 2021] kann daher die 3D-Posenschätzung erst dann erfolgreich in Realweltanwendungen eingesetzt werden, wenn sie auch ausreichend gut unter verschiedensten Bedingungen funktioniert, was das Training und Testen mit entsprechenden Datensätzen voraussetzt. Verschiedene Ansätze, werden in [Wang et al., 2021] vorgestellt, die versuchen diese bestehende Problematik abzumildern, beispielsweise durch den Einsatz von RGBD-Kameras und Simulationen. Dennoch ist fraglich, ob diese Bestrebungen alleinig ausreichen werden.

4.5 Rekonstruktionsbasierte Verfahren

Die Verwendung von Multi-View-Kamerasystemen erfordert das Zusammenführen der Informationen bzw. Ergebnisse der Einzelkameras in einem Verfahrensschritt der Sensordatenfusion, um am Ende ein Gesamtergebnis für das Tracking oder die Pose Estimation für jeden betrachteten Frame generieren zu können. Eine Möglichkeit besteht darin, basierend auf den Einzelbildern, zunächst die Rekonstruktion einer Visuellen Hülle vorzunehmen und damit Raumbelegungen und Oberflächen zu approximieren. Anschließend kann ein Tracking oder eine Posenschätzung, losgelöst von den einzelnen Kameraperspektiven, im 3D-Raum durchgeführt werden, wobei die Kamerabilder noch als zusätzliche Informationsquelle dienen können. In Abschnitt 9.1 werden ausgewählte Verfahren dieser Klasse konkret beschrieben, die sich in ihrer Vorgehensweise mitunter deutlich unterscheiden.

Die Mehrheit der in der Literatur gefundenen rekonstruktionsbasierten Ansätze verwenden Skelett- und Oberflächenmodelle und lassen sich daher der Pose Estimation

Rekonstruktions-basierte Ansätze	**Human Tracking**	**Human Pose Estimation**
Publikationen	[Canton-Ferrer et al., 2011] [Hofmann, 2011] [Marrón et al., 2009]	[Caillette, 2006] [Cheung et al., 2003] [Corazza et al., 2010] [De Aguiar et al., 2004] [Mikić et al., 2003] [Moschini und Fusiello, 2009] [Padeleris et al., 2013] [Tran und Trivedi, 2008]
Anwendung	Überwachung Lokalisierung	Motion Capture
Objektmodell (falls vorhanden)	Primitiv	Körpermodell
Personen	1 bis mehrere	1
Statische Objekte	0 bis mehrere	keine

Abb. 4.10: Gegenüberstellung rekonstruktionsbasierter Verfahren des Human Trackings und der Pose Estimation.

zuordnen. Es werden aber auch 3D-Primitive als Objektmodelle eingesetzt, wie ein Ellipsoidmodell in [Canton-Ferrer et al., 2011] oder ein Zylindermodell in [Hofmann, 2011]. Solche Verfahren werden als Human Tracking klassifiziert, insbesondere wenn zusätzlich eine Filtertechnik des Trackings eingesetzt wird (was jedoch nicht immer der Fall ist). Die Abb. 4.10 zeigt Publikationen der zwei genannten Klassen. Aus [Munder, 2013] sind weitere Quellen zur rekonstruktionsbasierten Pose Estimation entnehmbar sowie ein detaillierter Vergleich dieser. Im Übersichtsartikel von [Tran und Trivedi, 2008] erfolgt ebenfalls ein Vergleich verschiedener Verfahren der Pose Estimation.

Unterschiede zwischen rekonstruktionsbasierten Verfahren des Human Trackings und der Pose Estimation in Bezug auf ihre Einsetzbarkeit sind in Abb. 4.10 eingetragen. Verfahren des Human Trackings zielen auf eine Personenlokalisierung in Innenräumen wie beispielsweise Büros oder Smart Homes ab, was aus den betrachteten experimentellen Aufbauten hervorgeht. Charakteristisch hierfür ist die mögliche Präsenz statischer und dynamischer Objekte. Hingegen wurden die Verfahren mit komplizierteren Körpermodellen für eine detaillierte Bewegungserfassung (Motion Capture) in leeren Räumen untersucht. Die Erfolge der Posenschätzungen gehen dabei häufig mit Einschränkungen einher, worauf im nachfolgenden Abschnitt näher eingegangen wird. Zudem bringen die Verfahren größtenteils einen erhöhten Berechnungsaufwand mit

(a) (b) (c)

Abb. 4.11: Ansatz und Bilder aus [Canton-Ferrer et al., 2011, S. 3,7]. Für das Beobachtungsmodell wird die Visuelle Hülle rekonstruiert (a) und nachträglich koloriert (b). Dargestellt sind die besten Schätzungen von fünf Partikelfiltern, die ein Ellipsoidmodell tracken (c).

sich. Für diese Verfahrensklasse ist derzeit nicht erkennbar, ob zukünftig weitere Bestrebungen unternommen werden, um die Posenschätzung für mehrere Personen im Überwachungsraum, auch in Anwesenheit statischer Objekte, zu ermöglichen.

Das Tracking mit einfachen Objektmodellen basierend auf 3D-Rekonstruktionsdaten kann im Vergleich zur Pose Estimation für das betrachtete Anwendungsszenario (vgl. Abschnitt 1.4) als erfolgreich bewertet werden und stellt eine für diese Dissertation relevante Verfahrensklasse dar. In Abbildung 4.11 ist dazu die Vorgehensweise von [Canton-Ferrer et al., 2011] abgebildet, bei welcher mehrere Personen im Raum getrackt werden.

4.6 Bewertung relevanter Forschungsgebiete

In diesem Kapitel wurden verschiedenste Forschungsgebiete vorgestellt. Von den in Abb. 4.1 schraffierten Forschungsgebieten, die Multi-View-Kamerasysteme einsetzen, sind insbesondere die Verfahrensklassen von Interesse, die eine Schätzung der Parameter von 3D-Objektmodellen bzw. Körpermodellen vornehmen. Konkret wurden dazu die folgenden Verfahrensklassen betrachtet:

- Human Pose Estimation mit klassischen Verfahren, ohne 3D-Rekonstruktion
- Human Pose Estimation mit Deep-Learning-Techniken (Deep 3D Human Pose Estimation), ohne 3D-Rekonstruktion
- Human Pose Estimation mit 3D-Rekonstruktion
- Human Tracking mit 3D-Rekonstruktion

Als ein Ergebnis der Recherche hat sich die Notwendigkeit des Einsatzes eines Multi-View-Kamerasystems für die Aufgabenstellung dieser Dissertation bestätigt. Befinden sich mehrere Personen im Überwachungsraum, so werden ausreichend Sensorinformationen benötigt, um die Vielzahl an Ambiguitäten, die entstehen können, aufzulösen und gute 3D-Schätzungen zu erzielen. Es ist zu erwarten, dass die Präsenz statischer Objekte im Überwachungsraum Verdeckungen verursacht, welche die Ambiguitäten weiter „verstärken“. Somit kann davon ausgegangen werden, dass der Einsatz eines Multi-View-Kamerasystems dadurch zusätzlich gerechtfertigt ist.

Bei der Pose Estimation spielen statische Objekte und Objektverdeckungen, zu denen diese führen können, bisher eine untergeordnete Rolle, worauf im Folgenden näher eingegangen wird. Anschließend wird die wissenschaftliche Lücke aufgezeigt, auf die in dieser Dissertation nachfolgend der Schwerpunkt gelegt wird und die sich auf die Verfahrensklasse des Human Trackings bezieht, bei der eine 3D-Rekonstruktion vorgenommen wird.

4.6.1 Umgang mit Verdeckungen

Allgemein ist das Problem der 3D-Posenschätzung aktuell noch nicht ausreichend für verschiedenste Situationen gelöst und die Herausforderungen sind groß (vgl. Abschnitt 4.4.4). In dem relativ jungen Forschungsbereich der 3D Deep Human Pose Estimation wurden dahingehend im letzten Jahrzehnt zwar deutliche Fortschritte erbracht. Dennoch stellt der derzeitige Mangel an „In-the-wild“-Datensätzen mit 3D-Annotationen (aufgrund der aufwändigen Erstellung) für die Lernverfahren aktuell einen Flaschenhals dar. Ohne geeignete Datensätze ist jedoch meist keine echte Generalisierungsfähigkeit zu erwarten, um die trainierten Verfahren auch in beliebigen Realweltszenarien einsetzen zu können. Befinden sich statische Objekte im Überwachungsraum, welche zu Objektverdeckungen bei den Personen führen können, so kommt dies erschwerend hinzu. In [Cheng et al., 2020] werden gegebene Trainingsdaten künstlich erweitert. Dazu werden die „Heatmaps“, die von einem „2D Keypoint Estimator“ generiert werden, zufällig maskiert (einzelne, mehrere oder alle). Auch werden virtuelle Verdeckungsbereiche definiert und alle Keypoints maskiert, die in diese Bereiche hineinfallen. Es konnte gezeigt werden, dass sich die Ausgaben der Pose Estimation für kurzzeitige vollständige oder partielle Objektverdeckungen verbessern. Diese Vorgehensweise erscheint sehr plausibel. In dieser Dissertation wird jedoch die 3D Deep Human Pose Estimation durch den genannten Mangel an geeigneten 3D-Datensätzen nicht weiterverfolgt.

Mit klassischen Verfahren der 3D-Posenschätzung konnte bisher noch kein Durchbruch bezüglich komplexer Situationen mit Verdeckungen erzielt werden. In [Munder, 2015]

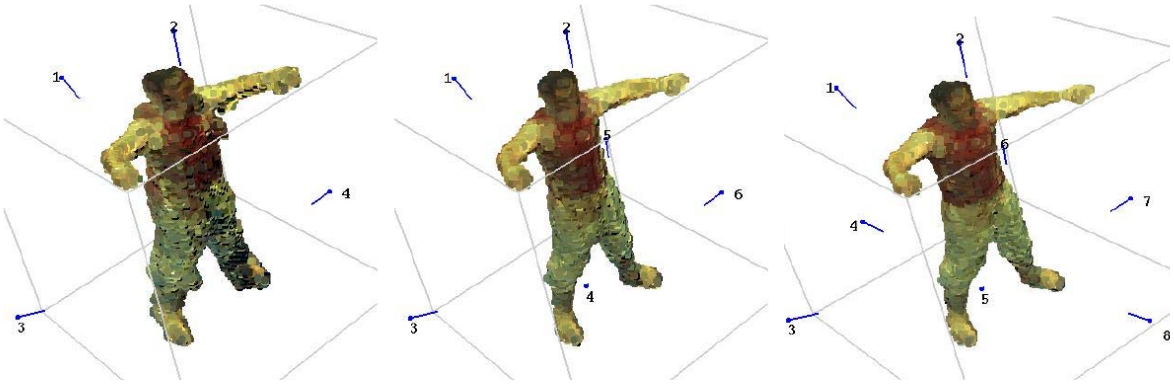


Abb. 4.12: Einfluss der Kameraanzahl auf die Rekonstruktionsergebnisse einer nachträglich kolorierten Visuellen Hülle, entnommen aus [Kehl et al., 2005, S. 3]. Zu sehen sind die Ergebnisse von vier Kameras (a), sechs Kameras (b) und acht Kameras (c).

wurde untersucht wie mit einem Bottom-Up-Ansatz die Gelenkwinkel eines Skelettmodells bestimmt werden können. Besonderheit hierbei war, dass einzelne Körperteile für das Multi-View-Kamerasystem (in einem oder mehreren Bildern der SIMERO-Roboterarbeitszelle) durch Selbstverdeckungen oder statische Objekte verdeckt waren. Die Untersuchungen verdeutlichten die Herausforderungen, die entstehen, wenn die Likelihood-Berechnungen der einzelnen Kameras fusioniert werden müssen, jedoch Informationen durch verdeckte Körperteile fehlen. Die Ergebnisse führten zu der Entscheidung, das Verfahren aus [Munder, 2015] nicht für die Ziele dieser Dissertation weiterzuentwickeln oder einen alternativen klassischen Ansatz für die Pose Estimation zu konzipieren.

Rekonstruktionsbasierte Verfahren der 3D-Posenschätzung finden derzeit wenig Beachtung. Mit den veröffentlichten Ansätzen kann die Pose einer Person in einem ansonsten leeren Raum geschätzt werden. Bezüglich der eingesetzten Algorithmen konnte bis auf die Behandlung der Selbstverdeckungen in [Kehl und Gool, 2006] keine Einbeziehung von Verdeckungsinformationen gefunden werden. In [Kehl und Gool, 2006] werden Objektselbstverdeckungen explizit in der Likelihood-Funktion berücksichtigt (vgl. Abschnitt 9.1). Eine Nachimplementierung dieses Verfahrens in [Munder, 2013] zeigte jedoch, dass das Verfahren schnell an seine Grenzen gelangt, wenn weitere Verdeckungen durch statische Objekte in der Szene hinzukommen und die Sichtbarkeit der Person(en) in den Kameras herabgesetzt wird.

Bei Präsenz statischer Objekte und mehrerer Personen im Überwachungsraum ist generell davon auszugehen, dass die Sichtbarkeit aufgrund der gegenseitigen Verdeckungen lokal deutlich variiert. Für die statischen Objekte der SIMERO-Arbeitszelle wird dies für den an späterer Stelle betrachteten Voxelraum dargestellt (vgl. Abb. 5.8 und 5.9).

Die Sichtbarkeitsgrade der Voxel zeigen an, für wie viele Kameras ein Voxel maximal vollständig sichtbar ist. Die lokal variierende Sensorabdeckung kann ortsabhängig zu deutlich unterschiedlichen Rekonstruktionsergebnissen führen, was die geometrische Approximation der Personen sowie die Güte der Voxeleinfärbung anbelangt. In [Kehl und Gool, 2006] ist dahingehend zwar kein Vergleich von Rekonstruktionsergebnissen unter der Anwesenheit anderer Objekte zu finden, aber es wird der Einfluss der Kameraanzahl auf die Rekonstruktion dargestellt. Dies wird in Abb. 4.12 verdeutlicht. Es ist erkennbar, dass sowohl die geometrische Approximation als auch die Güte der Kolorierung der Person bei acht Kameras und auch bei sechs Kameras besser ist als bei vier Kameras. Die Präsenz weiterer Objekte würde mit einer lokalen Herabsetzung der Sichtbarkeit einhergehen, was sich zusätzlich negativ auch auf das Rekonstruktionsergebnis auswirken sollte.

Auch wenn die rekonstruktionsbasierte Posenschätzung noch Entwicklungspotential bietet, so könnte eine lokal variierende Rekonstruktionsgüte, die durch die Verdeckungen der statischen Objekte hervorgerufen wird, möglicherweise die erzielbare Qualität der Einpassung detaillierter Körpermodelle in die 3D-Daten limitieren. Konkrete Untersuchungen hierzu wären notwendig, um bessere Abschätzungen treffen zu können. Für die Zielstellung dieser Dissertation werden diese jedoch nicht vorgenommen. Die Pose Estimation wird in Gänze nicht weiter verfolgt.

4.6.2 Betrachtete wissenschaftliche Lücke

Beim rekonstruktionsbasierten Human Tracking, das einfache Objektmodelle verwendet, konnten bereits Tracking-Erfolge für mehrere Personen in Anwesenheit statischer Objekte erzielt werden (vgl. Abschnitt 4.5). Es lässt sich annehmen, dass die einfachen Objektmodelle den Vorteil bieten, dass sie weniger detailreiche Rekonstruktionsdaten benötigen als entsprechende Körpermodelle der Pose Estimation und damit vermutlich weniger störanfällig bei ihrer Einpassung in Rekonstruktionsdaten variierender Güte sind.

Aus den experimentellen Untersuchungen der rekonstruktionsbasierten Verfahren des Human Trackings lässt sich nicht entnehmen, ob auch „größere“ Objektverdeckungen der Personen, hervorgerufen von den statischen Objekten, vorlagen. Die statischen Objekte sind mitunter so am Rand des Überwachungsraums platziert, dass die getrackten Personen durch diese vermutlich nicht merklich verdeckt werden wie in [Hofmann, 2011]. Bei [Marrón et al., 2009] werden statische Objekte erst zur Laufzeit in den Überwachungsraum eingebracht, wodurch diese – inklusive Verdeckungsvolumina – Teil der Rekonstruktion sind. Bei [Canton-Ferrer et al., 2011] befinden sich statische Objekte,

wie Tische und Stühle, in der Raummitte. Die dargestellten Situationen zeigen auch am Tisch sitzende Personen. Sie werden partiell verdeckt, jedoch anscheinend nicht in größerem Maßstab.

Die Bewertung dieser Tracking-Verfahren bezüglich ihres Umgangs mit den Verdeckungen stellt sich aus genannten Gründen als schwierig heraus. Die tatsächlich aufgetretene Größenordnung der Objektverdeckungen in den Videosequenzen wird nicht quantifiziert und thematisiert. Die Betrachtung vollständiger Objektverdeckungen geht aus den Experimenten nicht hervor und auch bei partiellen Objektverdeckungen ist deren Dauer und Größenordnung nicht bekannt. Zudem wurde bei den bestehenden recherchierten Verfahren dieser Klassen keine explizite Behandlung statischer Objekte und deren Verdeckungen vorgenommen. Dies stellt eine wissenschaftliche Lücke dar, auf die der Fokus dieser Dissertation gelegt werden soll.

Das Verfahren von [Canton-Ferrer et al., 2011] wird für die gewählte Aufgabenstellung (vgl. Abschnitt 1.5) als vielversprechend bewertet. Darin wurde ein Mehrpersonen-Tracking mit Ellipsoidmodellen umgesetzt, das u. a. einen Partikelfilter für das Tracking verwendet. Details gehen aus Abschnitt 9.1 hervor. Dieser Ansatz wird als Grundlage genommen und weiterverfolgt. Die Untersuchungen dieser Dissertation können demnach als Fortsetzung dazu betrachtet werden. Der Fokus liegt auf der Integration von Wissen zu Verdeckungsvolumina in die Likelihood-Funktion, die von statischen Objekten verursacht werden. Dieses Wissen wird in Form von 3D-Voxelmerkmalen bereitgestellt. Die Vorgehensweise wird in Abschnitt 6.1 erläutert.

4.7 Zusammenfassung

In diesem Kapitel wurde eine kurze Einführung in verschiedene Forschungsgebiete gegeben, die für die Aufgabenstellung dieser Dissertation von Interesse sind. Dabei rückten Verfahren des Human Trackings und der Human Pose Estimation in den Fokus. Gewählte Objektmodelle der zu verfolgenden Personen besitzen beim Human Tracking eine eher geringe Anzahl an Dimensionen. Hingegen werden kompliziertere Körpermodelle mit erhöhter Dimensionalität zur Posenschätzung eingesetzt. Grundlegende Vorgehensweisen des Trackings im Bildraum und im Zustandsraum wurden vorgestellt. Für die Human Pose Estimation wurden verschiedene Kategorien klassischer Verfahren beschrieben sowie Kategorien zu Verfahren der 3D Deep Human Pose Estimation. Die positiven Entwicklungen im letztgenannten Forschungsgebiet sind hervorzuheben. Weitere Fortschritte sind erwartbar, sobald mehr geeignete „In-the-Wild“-Datensätze für das Training der Lernverfahren zur Verfügung stehen.

Die Präsenz mehrerer Personen und statischer Objekte im Überwachungsraum führt zu Ambiguitäten bei der Schätzung. Zum Umgang mit diesen sind insbesondere Multi-View-Verfahren interessant, die Daten mehrerer Kameras verarbeiten und fusionieren. Ein Anteil dieser Verfahren nimmt die 3D-Rekonstruktion einer Visuellen Hülle vor. Für die Betrachtungen dieser Dissertation wurde die Kategorie rekonstruktionsbasierter Verfahren des Human Trackings als geeignet eingestuft. Ein erfolgreiches Mehrpersonen-Tracking konnte für diese Kategorie bereits gezeigt werden. Jedoch ist eine Bewertung der Verfahren bezüglich ihrer Güte im Umgang mit Verdeckungen, die durch statische Objekte verursacht werden, schwierig. Dies liegt daran, dass die Größe, Dauer und Häufigkeit der tatsächlich erzeugten Objektverdeckungen in den Experimenten nicht quantifiziert oder beschrieben wird. Die Integration von Wissen zu statischen Objekten und deren Verdeckungen stellt für die genannte Verfahrensklasse eine wissenschaftliche Lücke dar, die in dieser Dissertation adressiert werden soll. Die gewählte Vorgehensweise in [Canton-Ferrer et al., 2011] ist für ein Überwachungsszenario sehr schlüssig und wird in Teilen übernommen, weshalb die Untersuchungen dieser Dissertation auch als eine Art Erweiterung dieses Ansatzes verstanden werden können.

Kapitel 5

3D-Rekonstruktion

Im vorangegangenen Kapitel wurde der Einsatz von Rekonstruktionsdaten eines Multi-View-Kamerasystems für das Personen-Tracking motiviert. In diesem Kapitel wird detailliert auf die Rekonstruktion eingegangen. Zunächst wird in den Abschnitten 5.1 und 5.2 ein Überblick über Anwendungsbereiche der 3D-Rekonstruktion und ihre Grundprinzipien gegeben. Anschließend wird das Konzept zur Rekonstruktion einer Visuellen Hülle in Abschnitt 5.3 erläutert. Dazu wird ausführlich ein voxelbasiertes Verfahren dargestellt, das in [Kuhn und Henrich, 2009] präsentiert wurde und ähnlich in dieser Dissertation eingesetzt wird (Abschnitt 5.3.3). Besonderheit dieses Verfahrens ist die explizite Integration von Wissen zu Verdeckungsvolumina statischer Objekte in den Rekonstruktionsprozess, mit dem Ziel der Erzeugung einer konservativen Visuellen Hülle, die sämtliche Objekte des Überwachungsraums vollständig enthält. Ein Algorithmus dazu ist in Abschnitt 5.4 beschrieben. Die Bestandteile der Rekonstruktion werden in Abschnitt 5.5 erläutert. In Abschnitt 5.6 soll eine mengentheoretische Betrachtungsweise die Rekonstruktionsbestandteile besser veranschaulichen. Aus den Mengen wird in Abschnitt 5.7 eine Klassifikation der Voxel in sogenannte Voxelzustände abgeleitet. Diese werden für die Likelihood-Berechnungen des Tracking-Verfahrens benötigt.

Als Alternative zur Visuellen Hülle wird in Abschnitt 9.2.3 die Erzeugung einer Photohülle mittels eines Space-Carving-Algorithmus vorgestellt. In [Zwicker, 2013], [Ober-Gecks et al., 2014b] und [Ober-Gecks et al., 2016] wird gezeigt, wie solch ein iteratives Verfahren unter Verwendung der „Graphics Processing Unit“ (GPU) beschleunigt werden kann, damit es für die Online-Anwendung eines Tracking-Verfahrens einsetzbar ist. Dabei entstand auch eine GPU-basierte Implementierung der Visuellen Hülle, die als Eingabe für die Berechnung der Photohülle dient. Die Algorithmen dazu sind in den Abschnitten 9.2.3 und 9.2.4 zu finden. Eine Darstellung der experimentellen Ergebnisse für das gewählte Anwendungsszenario erfolgt in Abschnitt 9.3.

5.1 Anwendungsbereiche

Der Mensch bewegt sich in einer dreidimensionalen Welt und ist es gewohnt, dreidimensionale Informationen von Objekten zu erfassen und zu verarbeiten. Von allen zur

Verfügung stehenden Sinnen kann über den visuellen Kanal (Sehnerv) mit ca. 1 MBit/s die höchste Übertragungsrate erreicht werden [Koch et al., 2006]. Dementsprechend empfänglich ist der Mensch auch für virtuelle 3D-Daten in Form von 3D-Bildern, 3D-Animationen oder einer virtuellen Realität, die mithilfe von Computertechnik erstellt werden können.

Die manuelle Erstellung dreidimensionaler Computermodelle von Objekten ist sehr aufwändig und anwendungsabhängig teilweise nicht möglich, beispielsweise bei Echtzeitanforderungen. Ein großes Hilfsmittel stellen hierfür Verfahren der 3D-Rekonstruktion dar. Diese ermöglichen die Erfassung von Objekten und Szenen der Realität durch die sensorische rechnergestützte Vermessung und die sich daran anschließende Erzeugung digitaler 3D-Modelle. Die Realität wird damit nachgebildet bzw. rekonstruiert und dem Computer werden die benötigten Geometrien und Oberflächeninformationen von Objekten für die Weiterverarbeitung geliefert.

Die 3D-Rekonstruktion hat sich in verschiedenen Anwendungsbereichen etabliert und gewinnt weiterhin an Bedeutung. Neben der Unterhaltungsbranche wird die 3D-Rekonstruktion in der Kunst und Architektur eingesetzt, um beispielsweise Skulpturen und Monumente abzutasten und für die Nachwelt zu konservieren. Auch können z. B. bestehende Gebäude digital erfasst und die rechnergestützte Planung und Entwurfserstellung vereinfacht werden. In der Robotertechnik ist die genaue Erfassung und Modellierung der 3D-Welt bedeutend, um kollisionsfreie Roboterbewegungen zu gewährleisten, wie es beispielsweise als Ziel im SIMERO-Projekt verfolgt wird (vgl. Abschnitt 2.3). Für Roboter, die mit ihrer Umwelt interagieren und manipulatorisch darin eingreifen, ist es notwendig, Wissen zu Objekten ihrer Umgebung zu erfassen. Auch hierfür kann die sensorbasierte Rekonstruktion hilfreich sein, um 3D-Daten für Aufgaben der Merkmalsextraktion und Klassifikation bereitzustellen. Die Rekonstruktion kann weiterhin dazu dienen, dem Menschen Informationen zu liefern, die ihm durch die eigene Wahrnehmung nicht zugänglich sind. So werden in der Medizintechnik Organe und Gewebe des menschlichen Körpers aus Einzelaufnahmen (z. B. bei der Computertomographie) rekonstruiert, was die Diagnoseerstellung verbessern kann und auch bei Operationen unterstützend angewendet wird.

Zusammenfassend werden Verfahren der 3D-Rekonstruktion eingesetzt, um Objektgeometrien und -oberflächen möglichst fotorealistisch zum Zwecke der Visualisierung oder zur Informationsgewinnung zu erfassen. Zudem soll über weitere Verarbeitungsschritte, wie eine Merkmalsextraktion oder Klassifikation von Objekten, Wissen über die Umgebung generiert und für spezielle Anwendungen nutzbar gemacht werden.

5.2 Überblick zu Rekonstruktionsprinzipien

Wie in Abschnitt 2.1 beschrieben, stehen zur Vermessung von Räumen und Objekten verschiedene aktive und passive Sensoren mit unterschiedlichen Eigenschaften zur Verfügung, welche Eingabedaten für eine 3D-Rekonstruktion liefern können. Mitunter wird bereits bei Einzelaufnahmen mit einem 1,5D oder 2,5D-Sensor von 3D-Daten gesprochen, da man einen dreidimensionalen Ausschnitt der Umgebung erhält. Der Begriff 3D-Rekonstruktion soll in dieser Dissertation im Vergleich dazu für Verfahren stehen, die darauf abzielen, möglichst vollständige Objektgeometrien zu liefern. Dies ist jedoch durch die Verwendung einer Einzelaufnahme aus einer Blickrichtung, z. B. mit einem 2,5D-Sensor, nicht möglich. Denn wie in Abschnitt 3.2.3 dargelegt, verdeckt ein Objekt aus Sicht eines Sensors immer auch einen Teil von sich selbst (Objektselbstverdeckung). Zudem können derzeit für Überwachungsszenarien sinnvoll einsetzbare Sensoren nur bis zur nächsten Oberfläche messen und keine Aussage über dahinter liegende Raumbereiche treffen, außer bei transparenten Objekten.

Ein idealer 3D-Sensor würde die Belegungsinformation jedes Raumpunkts bestimmen können und damit auch die Objektgeometrien vollständig liefern. Für eine Annäherung an solch ideale Sensordaten können Datenaufnahmen aus unterschiedlichen Blickrichtungen (engl. Multi-Views) von dem Objekt erzeugt und fusioniert werden. Für die Datengewinnung gibt es zwei prinzipielle Vorgehensweisen: Entweder es werden verschiedene, zeitlich versetzte Aufnahmen von Objekten mit einem einzelnen Sensor gemacht oder es werden mehrere Sensoren verwendet, welche simultan aus unterschiedlichen Blickrichtungen Bilder aufnehmen. Erstere Methode kommt beispielsweise bei der Modellerstellung für den 3D-Druck zum Einsatz. Ein Objekt wird auf einem Drehteller platziert und von einem fixen Laser-Kamera-System hochauflösend vermessen. Solch eine iterative Prozedur benötigt einen gewissen Zeitaufwand, um die Objekte von ausreichend vielen Seiten zu erfassen. Für Online-Rekonstruktionsanforderungen wie bei dem betrachteten Anwendungsszenario wird hingegen ein System bestehend aus mehreren Sensoren benötigt, was einen höheren Hardware-Aufwand bedeutet, auch um die große zu einem Zeitpunkt entstehende Datenmenge entsprechend schnell verarbeiten zu können.

In Abschnitt 2.1 wurde die Verwendung eines $C\times$2D-Farbkamerasystems im Vergleich zu einem $C\times$2,5D-Kamerasystem diskutiert (C ist die Kameraanzahl), da es einen deutlichen Unterschied macht, ob auf Sensorebene Tiefeninformationen oder Intensitäts- und Farbinformationen geliefert werden. Erstere versprechen, zumindest theoretisch, bessere Approximationsgüten der Objektgeometrien. Einschränkungen bei deren praktischem Einsatz wurden bereits benannt. Beispielsweise beeinflussen sich aktive 2,5D-Kameras

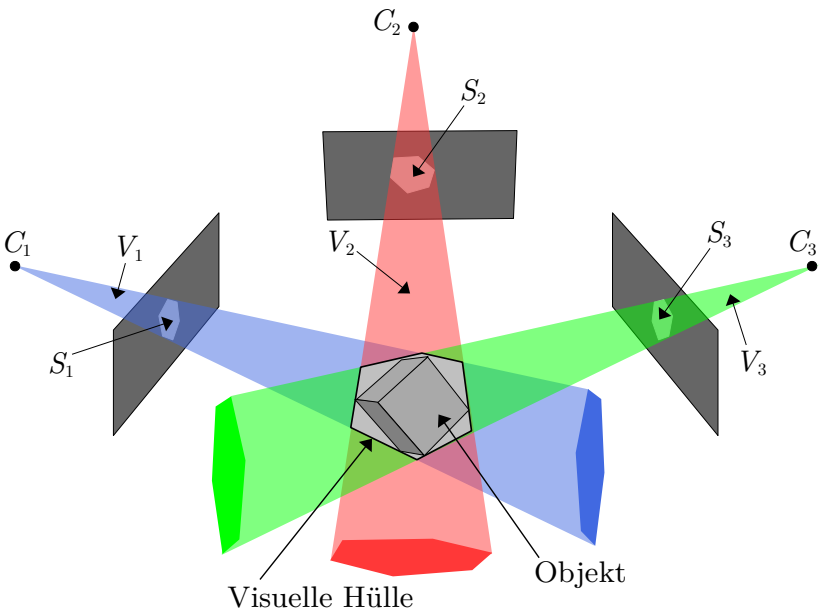


Abb. 5.1: Shape-from-Silhouette-Verfahren zur Erzeugung einer Visuellen Hülle. Ein Objekt wird von mehreren Kameras C_i aufgezeichnet. Die Silhouetten S_i des Objekts in den Bildern werden in den 3D-Raum rückprojiziert, was die Volumina V_i ergibt. Diese werden miteinander verschnitten, wodurch eine approximierte Form des Objekts entsteht, die Visuelle Hülle genannt wird. Abbildung in Anlehnung an [Ohsawa et al., 2013, S. 4].

gegenseitig. Passive 2,5D-Kameras (Stereokameras) hingegen können erhebliche Probleme bei der Korrespondenzanalyse bekommen. Die Entscheidung für ein Kamerasystem in dieser Dissertation ist deshalb auf ein passives Cx2D-Farbkamerasystem gefallen, bestehend aus Farbkameras mit Wide-Baseline-Anordnung. In Abschnitt 2.1 wurde dies ausführlicher begründet.

Für Bilddaten passiver Farbkameras stehen verschiedene Rekonstruktionsmöglichkeiten zur Verfügung, um eine Approximation der Objektgeometrien im Überwachungsraum zu erhalten und damit auch Tiefeninformationen zu erzeugen, die aus Einzelaufnahmen von 2D-Farbkameras nicht hervorgehen. Zwei grundlegend verschiedene Vorgehensweisen können als **Volumenverschneidung** und **Korrespondenzanalyse** bezeichnet werden. Bei der Volumenverschneidung nach dem **Shape-from-Silhouette-Verfahren** werden Objekte in den Kamerabildern segmentiert und in den Raum rückprojiziert. Die rückprojizierten Volumina mehrerer Kameras werden miteinander verschnitten, sodass sich eine **Visuelle Hülle** (VH) ergibt, die das Objekt enthält (vgl. Abb. 5.1). Das Konzept dazu wird in Abschnitt 5.3 detaillierter vorgestellt.

Bezüglich einer Korrespondenzanalyse wurde in Abschnitt 2.1 bereits erwähnt, dass eine punktweise Korrespondenzsuche, wie sie bei passiven Stereokameras erfolgt, bei einem Kamerasystem mit Wide-Baseline-Anordnung schwierig ist. Der Einfluss von Verdeckungen (z. B. Objektselbstverdeckungen) wird größer, sodass nicht immer ausreichend

viele korrespondierende Oberflächenpunkte in den unterschiedlichen Kamerabildern existieren, die eine erfolgreiche Erfassung der Objektgeometrien mittels Triangulierung ermöglichen. Spezielle Verfahren für Wide-Baseline-Kameras ermitteln korrespondierende Oberflächenbereiche auf eine andere Art und Weise. Bei der voxelbasierten Rekonstruktion einer **Photohülle** werden Oberflächenvoxel im Verlauf eines iterativen Rekonstruktionsprozesses bestimmt. Hierfür muss in jeder Iteration mit Hilfe sogenannter Sichtbarkeitstests mit größerem Aufwand die Sichtbarkeit der Voxel in den Kameras ermittelt werden. Erst nachfolgend kann für jedes Voxel anhand des Korrespondenzmerkmals Farbe überprüft werden, ob es sich möglicherweise auf einer Objektoberfläche befindet. Dies ist nach Annahme dann der Fall, wenn ein Voxel in allen Kameras, in denen es sichtbar ist, auch dieselbe Farbe erzeugt (Farbkonsistenz).

Für die Rekonstruktion einer Visuellen Hülle als auch einer Photohülle existieren zahlreiche Modifikationen und Erweiterungen der grundlegenden Verfahren. So können Oberflächenrepräsentationen in Kombination mit Energieminimierungsfunktionen zum Einsatz kommen oder auch Volumenmodelle (Primitive) von zu rekonstruierenden Objekten verwendet werden, um bessere Ergebnisse zu erzielen. Auch Nachbarschaftsanalysen und morphologische Operatoren werden hierfür eingesetzt. Viele Verfahren gehen damit über eine grundständige Rekonstruktion hinaus, indem Zusatzwissen von den zu rekonstruierenden Objekten in Form von Annahmen oder Objektmodellen integriert wird. Für den Fokus dieser Dissertation wird ein Verfahren der Visuellen Hülle gewählt, das Objekte ausschließlich basierend auf den Bilddaten rekonstruiert und sich überdies auch für eine Online-Rekonstruktion einsetzen lässt. Wie bereits zu Beginn dieses Kapitels erwähnt, wird in Abschnitt 9.2 die Umsetzung einer Photohülle für die gegebenen Anforderungen vorgestellt.

5.3 Konzept der Visuellen Hülle

Zur Berechnung der geometrischen Approximation eines Objekts im 3D-Raum wird häufig ein Shape-from-Silhouette-Verfahren (SfS) eingesetzt. Dabei werden Bilder unterschiedlicher Perspektiven des Objekts verwendet, um das ursprüngliche Objekt zu rekonstruieren und so ein digitales Objektmodell zu erzeugen. Genauer gesagt, werden für jede Kamera die Bildpunkte, auf die das Objekt projiziert, hergenommen und die Kontur der sich ergebenden Fläche (Silhouette) im Bild in den 3D-Raum rückprojiziert (vgl. Abb. 5.1). Die Rückprojektion einer Silhouette bildet einen **Sichtkegel**, auch Konturkegel genannt (engl. Visual Cone). Jeder Sichtkegel weist eine unendliche Tiefe auf und umschließt das zu rekonstruierende Objekt vollständig, sofern die zugehörige Silhouette auch alle Projektionspunkte des Objekts enthält. Verschneidet man die

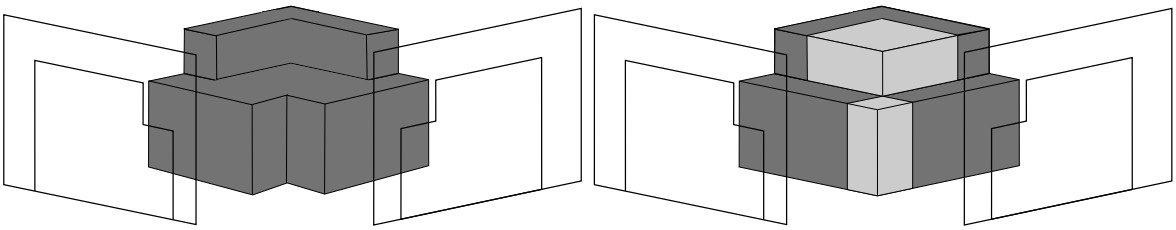


Abb. 5.2: Grenzen des Shape-from-Silhouette-Verfahrens, entnommen aus [Ober-Gecks et al., 2016, S. 5]. Zu sehen ist ein nicht-konvexes Objekt im 3D-Raum sowie zwei Silhouettenbilder des Objekts von zwei unterschiedlichen Kameraperspektiven (a). Die Rekonstruktion, die aus der Rückprojektion der Silhouetten resultiert (b), ist größer als das reale Objekt (zusätzliche hellgraue Quader), was bedeutet, dass nicht alle Details rekonstruiert werden können.

Sichtkegel mehrerer Silhouettenbilder (die von Kameras mit unterschiedlicher Perspektive stammen) miteinander, so erhält man als Resultat des Volumenverschnitts eine Approximation der Objektform (engl. Shape). Nicht alle Details können bei dieser Vorgehensweise genau rekonstruiert werden. In Abbildung 5.2 wird dies anhand eines nicht-konvexen Objekts veranschaulicht, dessen Rekonstruktion Teile enthält, die im originalen Objekt nicht vorhanden sind.

Die Idee des SfS wurde bereits 1974 veröffentlicht [Baumgart, 1974]. Laurentini führte den Begriff der Visuellen Hülle (engl. Visual Hull) ein. Details zur konkreten Definition können in [Laurentini, 1991] und [Laurentini, 1994] nachgeschlagen werden.

5.3.1 Eigenschaften

Nach [Laurentini, 1994] ist die VH die bestmögliche Approximation eines 3D-Objekts, die allein durch das Verschneiden von Sichtkegeln erreicht werden kann. Umgekehrt betrachtet, stellt die VH die maximale Geometrie dar, welche die gegebenen Silhouettenbilder erzeugen können. Jedes größere Objekt würde eine Veränderung der Silhouetten bewirken. Es existieren jedoch unendlich viele kleinere Objekte, die zu den selben Silhouettenbildern führen. Aufgrund dieser Mehrdeutigkeit kann man bei der Rekonstruktion tatsächlich nur von einer Approximation der Objektform sprechen. Genaueres zur Rekonstruierbarkeit verschiedener Geometrien findet man in [Laurentini, 1994] und [Laurentini, 1995]. Darin wird beschrieben, welche Objektoberflächen einen Einfluss auf die Silhouettenbilder haben. Wie in Abb. 5.3 dargestellt, klassifiziert Laurentini die Objektoberflächen nach silhouettenaktiv und silhouetteninaktiv. Nur die silhouettenaktiven Oberflächen beeinflussen bei einer Veränderung die VH.

Die Berechnung der VH ist unter Verwendung geeigneter Datenstrukturen, wie Voxel, ein einfaches und schnelles Verfahren.Die Präzision der Approximation lässt sich verbessern,

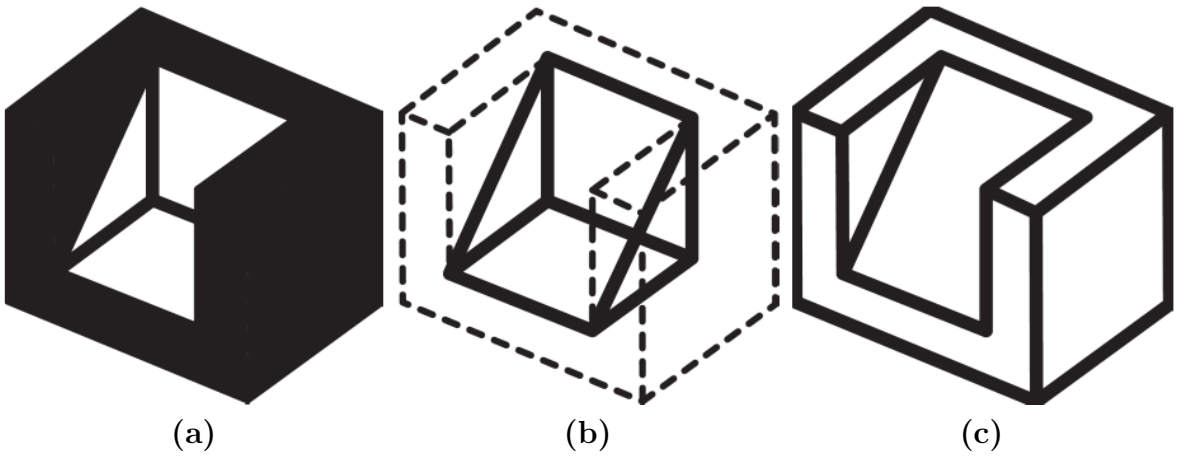


Abb. 5.3: Kategorisierung der Oberflächen eines Objekts nach ihrem Einfluss auf die resultierende VH, entnommen aus [Laurentini, 1994]. Zu sehen sind silhouettenaktive Flächen (schwarz) und silhouetteninaktive Flächen (weiß) in (a). Frei veränderbare inaktive Flächen beeinflussen die Rekonstruktion nicht (b). Die VH ist die beste Approximation, die man durch ein SfS-Verfahren von der gegebenen Geometrie erzeugen kann (c). Die nicht konvexe Aussparung des Quaders kann damit nicht präziser rekonstruiert werden als in (c) dargestellt.

indem man die Anzahl der Sichtkegel zum Verschneiden durch das Hinzufügen weiterer Kameras erhöht. Dennoch bleibt das Ergebnis im allgemeinen Fall eine Approximation des Objekts, was auch die Verwendung unendlich vieler Kameras nicht ändern würde. Bei verschiedenen nicht-konvexen Objekten und ungünstigen Kameraperspektiven auf das Objekt kann die VH gegenüber der tatsächlichen Objektgeometrie besonders groß ausfallen. Hinzu kommen spezifische Ungenauigkeiten, die aus der gewählten Datenstruktur resultieren. Beispielsweise hängen die Diskretisierungsfehler bei Voxeldaten von der gewählten Voxelgröße ab.

Weiterhin kann eine korrekte VH nur dann berechnet werden, wenn keine falsch-positiven und falsch-negativen Silhouettenpixel vorliegen. Oftmals wird eine automatisierte Objektdetektion eingesetzt, um die Objektsilhouetten in den Bildersequenzen zu bestimmen. Typischerweise handelt es sich dabei um Verfahren, die ein Background Subtraction (BS) vornehmen. Das Prinzip von BS-Verfahren und ihre Grenzen wurden bereits in Abschnitt 2.2 erläutert. Durch Störungen wie Schatteneffekte können die Silhouetten zu groß sein, wodurch sich die geometrische Approximation verschlechtert. Es können aber auch unvollständige Silhouettenbilder entstehen, die dazu führen, dass die rekonstruierte VH das entsprechende Objekt nicht in Gänze enthält. Eine Ursache hierfür sind Objektverdeckungen, die durch statische Objekte hervorgerufen werden. Lösungen für dieses Verdeckungsproblem werden in Abschnitt 5.3.3 erläutert. Eine weitere Fehlerquelle, die zu ähnlichen Effekten führen kann, ist eine schlechte Kamerakalibrierung.

Aufgrund der beschriebenen Eigenschaften und der Fehleranfälligkeit ist das SfS-Verfahren nicht für jede Anwendung geeignet. Es wird jedoch gerne verwendet, wenn

kein weiteres Wissen über die zu detektierenden Objekte zur Verfügung steht, so wie es häufig bei Überwachungsszenarien der Fall ist. Die VH kann auch als initiale Rekonstruktion eingesetzt werden, welche anschließend, beispielsweise durch ein stereobasiertes Verfahren, verfeinert wird.

5.3.2 Literaturübersicht

Zur Beschreibung der Form von Sichtkegeln sowie der resultierenden VH kommen oberflächenbasierte oder volumenbasierte Datenstrukturen zum Einsatz. Die Verwendung von Polyedern zur Oberflächenbeschreibung (vgl. [Franco und Boyer, 2009], [Fischer und Henrich, 2009] und [Lazebnik et al., 2007]) erfordert eine rechenaufwändige Extraktion der Konturkanten im Bild, z. B. mithilfe eines Suchverfahrens von Objektkanten. Zudem können hierbei numerische Probleme entstehen, die den algorithmischen Aufwand vergrößern bzw. Rekonstruktionsfehler begünstigen. Weitaus weniger komplex ist die Verwendung von Volumenmodellen wie Voxelräumen (vgl. [Kutulakos und Seitz, 2000] und [Ladikos et al., 2008]). Hierfür genügt ein Segmentierungsverfahren der Objektdetektion, welches die gesamte Fläche im Bild extrahiert, auf die das Objekt projiziert. Die segmentierten Pixel können dann einzeln betrachtet in den 3D-Raum rückprojiziert werden, z. B. vereinfacht als Pyramiden- oder Kegelvolumina [Henrich et al., 2008]. Die Fusion dieser einzelnen Volumina einer Kamera ergibt den Sichtkegel eines Silhouettenbildes. Eine alternative Datenstruktur zu den Voxeldaten stellen Conexeldaten dar, wie in [Casas und Salvador, 2006] und [Kuhn und Henrich, 2010] beschrieben. Bei Voxeldaten projiziert ein Voxel abhängig von seinem Abstand zur jeweiligen Kamera auf eine unterschiedliche Pixelanzahl, da jedes Voxel (üblicherweise Quader) das gleiche Volumen umfasst. Ein Conexel hingegen projiziert in jeder Kamera auf genau ein Pixel. Dadurch ergeben sich variierende Größen und Formen der Conexel. Dies hat den Vorteil, dass Quantisierungsfehler, die aufgrund der Raumunterteilung entstehen, vermieden werden. Nachteilig sind jedoch die hohe Speicherkomplexität der Conexel sowie die aufwendige Weiterverarbeitung bzw. Interpretation dieser ungleichmäßigen Raumaufteilung. Deshalb wird dieser Ansatz hier nicht weiter verfolgt, auch vor dem Hintergrund, dass ein Quantisierungsfehler ja bereits durch Kamerapixel selbst erzeugt wird, der mit dem Conexel-Ansatz nicht adressiert werden kann. Weiterhin besitzt das Quantisierungsproblem mit steigender Kameraauflösung eine geringer werdende Relevanz.

Zahlreiche SfS-Verfahren für eine optimierte VH existieren inzwischen. Viele davon zielen auf eine Beschleunigung der Berechnungsdauer ab. Hierfür eignen sich Octrees (vgl. [Ladikos et al., 2008], [Szeliski, 1993], [Werner und Henrich, 2014]) oder der Einsatz der Grafikkarte (vgl. [Ladikos et al., 2008] und [Schick und Stiefelhagen, 2009]).

(a) (b) (c)

Abb. 5.4: Verdeckende statische Objekte bei Background-Subtraction-Verfahren, Bilder einer Simulation. Ein Referenzbild der Szene mit statischen Objekten ist zu sehen in (a), ein Szenenbild mit dynamischem Objekt (Mensch) in (b). Als Ausgabe wird ein binäres Silhouettenbild erzeugt (c). Teile des Menschen sind verdeckt, wodurch die segmentierte Silhouette unvollständig ist.

In [Zwicker, 2013] wurde eine Lösung für OpenGL erarbeitet, die in Abschnitt 9.2.4 gründlich beschrieben wird. Zur Verbesserung der Qualität automatisiert erzeugter Silhouettenbilder werden probabilistische Ansätze verfolgt. So werden Wahrscheinlichkeitsbilder im Raum fusioniert, anstatt binäre Silhouettenbilder zu verschneiden, wie z. B. in [Salvador und Casas, 2008]. Für die anwendungsbezogene Bestimmung von Raumbelegungen, kodiert in sogenannten Occupancy-Grids, kommt ebenfalls die VH zum Einsatz (vgl. [Guan et al., 2008] und [Hofmann, 2011]). Eine Übertragung des Konzepts der VH als **Tiefenhülle** (engl. Depth Hull) erfolgte für den Einsatz von Tiefenkameras [Bogomjakov und Gotsman, 2008].

5.3.3 Verfahren zum Umgang mit verdeckenden Objekten

Unvollständige Silhouettenbilder, die bei der Objektdetektion entstehen, führen zu fehlerhaften Rekonstruktionen. Zur Erinnerung an Abschnitt 2.2: Beim Background Subtraction werden Referenzbilder der Szene – ohne dynamische Objekte – von jeder Kamera aufgenommen und zur Erstellung von Hintergrundmodellen verwendet. Während des Systembetriebs werden die Kamerabilder mit den zugehörigen Hintergrundmodellen abgeglichen. Es erfolgt eine automatische Segmentierung dynamischer Objekte, weil sich ihr Aussehen in der Regel von den Hintergrundobjekten unterscheidet, die in den Hintergrundmodellen kodiert sind. Verdeckungsvolumina, die von Hintergrundobjekten (statische Objekte) erzeugt werden, können dynamische Objekte verbergen, woraus sich unvollständige Silhouettenbilder ergeben, wie in Abbildung 5.4 visualisiert. Daraus resultiert eine Fehlerfortpflanzung bei der Rekonstruktion, wie in Abbildung 5.5(a) veranschaulicht, da eine VH nur dann Objekte vollständig enthält, wenn auch deren Objektsilhouetten vollständig sind.

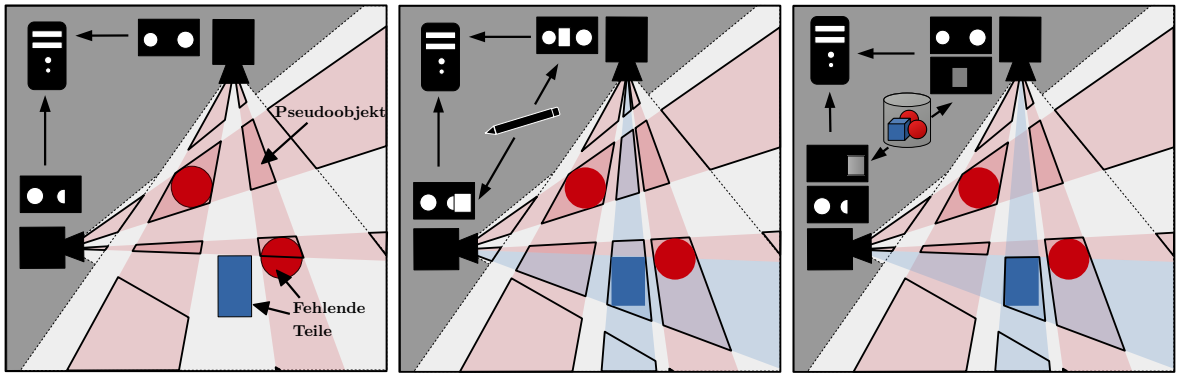


Abb. 5.5: Konzepte für die VH zum Umgang mit Objektverdeckungen, die durch statische Objekte verursacht werden. Fehlerhafte VH basierend auf unvollständigen Silhouettenbildern eines Background Subtractions (a). Zu den Silhouettenbildern werden manuell segmentierte Bilder der statischen Objekte (Occlusion Masks) hinzugefügt (b). Alle Objekte befinden sich damit innerhalb der VH. Nachteilig ist, dass bei der Verschneidung zusätzlicher Sichtkegel weitere Pseudoobjekte entstehen können (leere Volumina) sowie Volumina vergrößert werden können, die Objekte enthalten. Diese Effekte lassen sich durch das Einbringen von Wissen über die Geometrien der statischen Objekte in Form von Tiefenbildern verringern (c).

Verschiedene Lösungen existieren für dieses Verdeckungsproblem. In [Ladikos et al., 2008] werden Binärmasken bzw. Silhouettenbilder der Hintergrundobjekte verwendet, genannt **Occlusion Masks**. Diese werden zu jedem Zeitpunkt zu den Silhouettenbildern aus der Objektdetektion hinzugefügt (vgl. Abb. 5.5(b)). Während der Rekonstruktion werden sowohl die Silhouettenpixel als auch die Pixel der Occlusion Masks rückprojiziert. Dadurch befinden sich alle Objekte der Szene innerhalb der VH, vorausgesetzt die Silhouettenbilder und Occlusion Masks sind vollständig. Die binären Occlusion Masks statischer Objekte können manuell oder mit Hilfe von Lernverfahren erstellt werden [Guan et al., 2008]. Beispielsweise wird in [Guan et al., 2007] eine Bayes'sche Formulierung präsentiert, bei der extrahierte Verdeckungsmerkmale aus einer Bildersequenz mehrerer Kameras akkummuliert werden, sodass sich die Masken der Hintergrundobjekte in den Bildern abzeichnen. Die Verdeckungsmerkmale werden aus der Bewegung von Objekten im Raum gewonnen. In [Keck und Davis, 2008] wird ebenso ein iteratives Lernverfahren zur Detektion und Modellierung statischer Hintergrundobjekte für ein Multi-View-Kamerasystem vorgeschlagen. In Vorarbeiten des SIMERO-Projekts (vgl. Abschnitt 2.3) werden Binärmasken des dynamischen Roboters durch ein Online-Rendering des Robotermodells in die Kamerabilder erzeugt. Dies ist möglich, da die Roboterparametrisierung ständig von der Robotersteuerung abgefragt werden kann. Damit lassen sich die dynamischen Verdeckungsvolumina in die VH integrieren.

Nachteilig an der Verwendung von Occlusion Masks ist, dass die rückprojizierten Sichtkegel der Masken (wie auch die Sichtkegel der Silhouettenbilder) keine Information

darüber liefern, wo genau und in welchem Abstand zur Kamera sich die Objekte befinden und die Verdeckungsvolumina beginnen. Deshalb müssen die vollständigen Sichtkegel miteinander verschnitten werden, was aber zusätzliche leere Volumina erzeugt, wie Abb. 5.5(b) zeigt. Diese werden in der Literatur auch als **Pseudoobjekte** (engl. Pseudo Objects) oder **Geistervolumina** (engl. Ghost Volumes) bezeichnet. Neben Pseudoobjekten können sich aber auch Volumina, die ein Objekt enthalten, vergrößern und damit eine schlechtere geometrische Approximation der Form erzeugen.

Zur Reduktion der geschilderten Effekte beschäftigen sich [Schick und Stiefelhagen, 2009] und [Kuhn und Henrich, 2009] mit der Bestimmung des Freiraums bis zu den nächsten Objektoberflächen aus Kamerasicht. Da die Hintergrundobjekte oftmals statisch und damit längerfristig Teil des Überwachungsraums sind, ist es möglich, Apriori-Wissen zu diesen Objekten in die sensorbasierte Rekonstruktion einzubringen. In [Kuhn und Henrich, 2009] werden geometrische Modelle aller statischen Objekte im Überwachungsraum erstellt und damit Tiefenbilder generiert, die für jedes Kamerapixel den Abstand bis zur Oberfläche des nächsten statischen Objekts kodieren. Dadurch kann die Rekonstruktion im Vergleich zu den Occlusion Masks verbessert werden, wie in Abb. 5.5(c) veranschaulicht. In [Kuhn, 2012] wird ein ausführlicher Vergleich mit den Occlusion Masks von [Ladikos et al., 2008] durchgeführt. Nach jedem Rekonstruktionsschritt werden in [Kuhn, 2012] zusätzlich Volumina entfernt, in denen sich nach getroffenen konservativen Annahmen, den sogenannten „Plausibilisierungen“, keine Person befinden kann. Die Anwendbarkeit dieser Annahmen erfordert eine Segmentierung der unverdeckten Teile der Personen in den Kamerabildern (ohne falsch-negativ klassifizierte Pixel). Die Plausibilisierungen werden in dieser Dissertation nicht angewandt (vgl. Abschnitt 2.3.2). Die synthetischen Tiefenbilder werden jedoch ebenfalls zur Freiraumbestimmung genutzt, um bessere Rekonstruktionsergebnisse zu erhalten als mit den Occlusion Masks.

Für das eingesetzte Verfahren ähnlich [Kuhn und Henrich, 2009] ist in Abschnitt 9.2.4 ein GPU-basierter Algorithmus zu finden.

5.4 Rekonstruktionsalgorithmen

Im diesem Abschnitt wird der Algorithmus zur 3D-Rekonstruktion für eine „konservative VH mit Verdeckungsbehandlung“ hergeleitet, welche für diese Dissertation eine zentrale Rolle spielt. Die Inhalte und die Notation wurden weitestgehend von [Ober-Gecks et al., 2016] übernommen und stützen sich auf die Vorarbeiten von [Ober-Gecks et al., 2014b] und [Zwicker, 2013].

Eine Rekonstruktion wird für eine Menge diskreter Zeitpunkte k_i aus einem Zeitintervall

von k_{start} bis k_{end} ausgeführt. Um die folgenden Darstellungen zu vereinfachen, wird jeweils nur ein Zeitpunkt k aus der Menge betrachtet und auf einen zusätzlichen Zeitindex i verzichtet, ohne dass dadurch ein Verlust an Generalität entsteht.

Für die algorithmische Beschreibung wird in Abschnitt 5.4.1 die Datenstruktur Voxelraum (engl. Voxel Space), in welcher die Rekonstruktionsdaten abgelegt werden, definiert. Anschließend werden in Abschnitt 5.4.2 die Kamerabilder formal beschrieben.

5.4.1 Voxelraum

Zur Speicherung geometrischer Eigenschaften von Objekten auf dem Computer kann der 3D-Raum **diskretisiert** werden, d. h. es können Daten aus dem kontinuierlichen Bereich in einen diskreten Bereich (unter Informationsverlust) überführt werden. Hierfür steht eine Vielzahl an Datenstrukturen zur Verfügung, die hauptsächlich aus der Computergrafik und von CAD-Anwendungen stammen [Foley et al., 1996]. Häufig kommen Dreiecksnetze (engl. Triangle Meshes) zum Einsatz, da diese auf der Grafikkarte sehr performant Geometrien repräsentieren können. Dreiecksnetze beschreiben Oberflächen von Objekten, weshalb sie zu den Oberflächenmodellen (engl. Surface Models) gehören. Möchte man allerdings Informationen zu den Volumina von Objekten (z. B. deren Belegung) speichern, so muss man auf volumenbasierte Datenstrukturen (engl. Volume-based Models) zurückgreifen, wie z. B. Voxel, Octrees, Vereinigungen von 3D-Primitiven und andere.

Die orthogonale, gitterförmige Diskretisierung eines dreidimensionalen Raums wird als **Voxelraum** bezeichnet. Der Begriff **Voxel** stammt aus der Zusammensetzung von

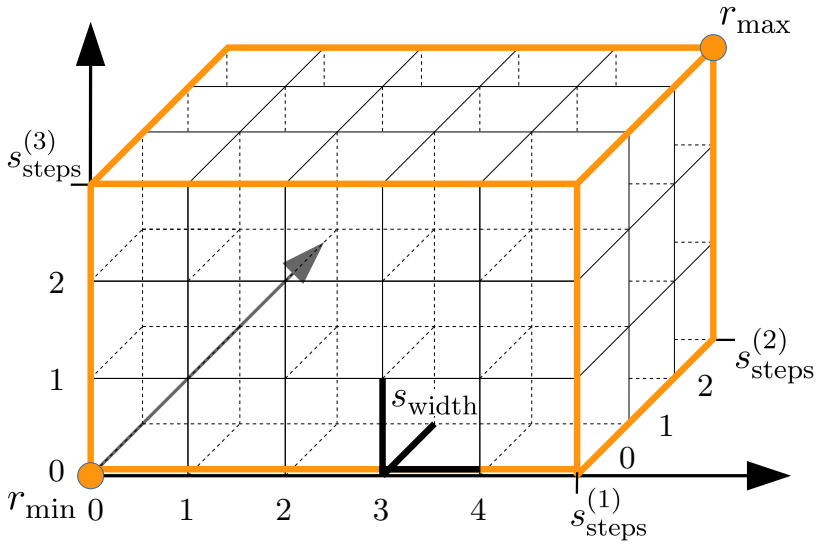


Abb. 5.6: Darstellung eines Voxelraums im Voxelraumkoordinatensystem (VKS)

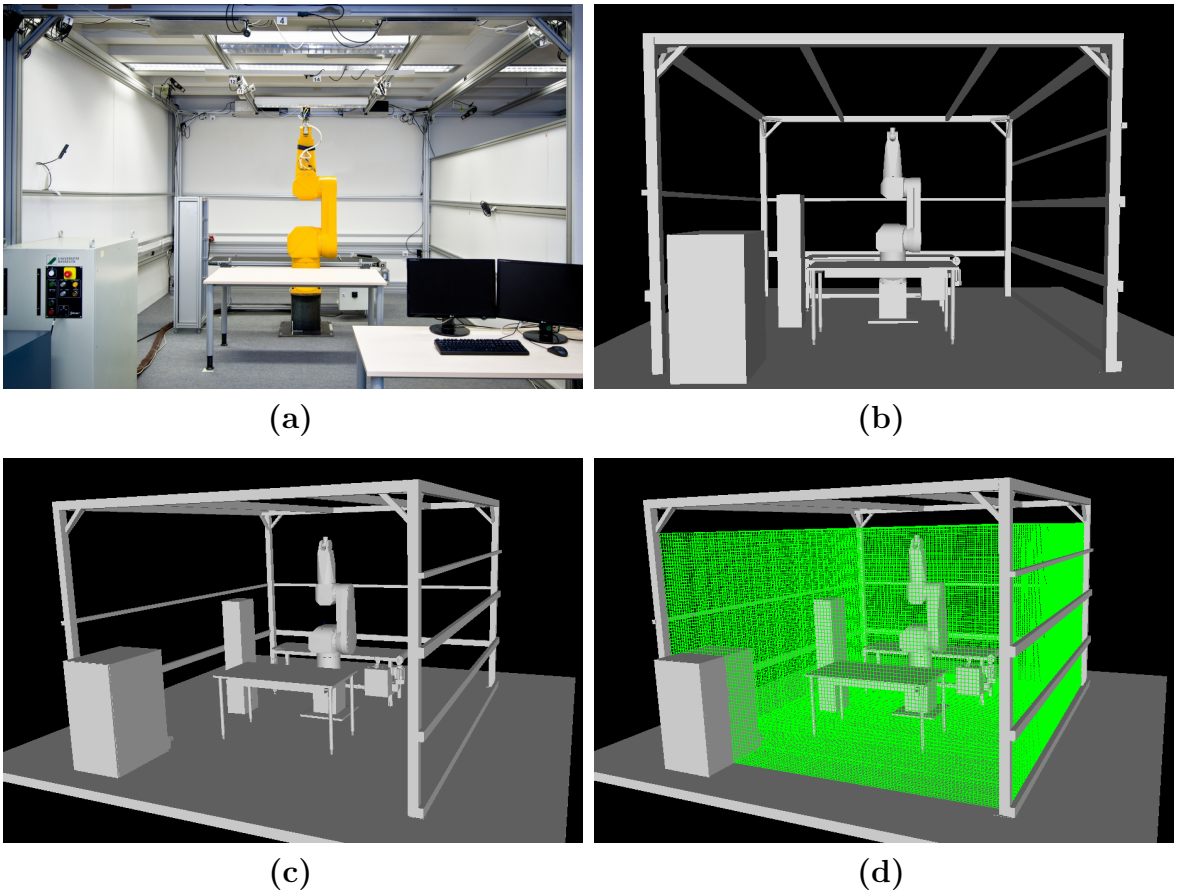


Abb. 5.7: Roboterarbeitszelle (a), Bilder einer mit Blender erstellten Simulationsumgebung (b) und (c), Voxelraum (d).

„Volumetric“ und „Pixel“ [Foley et al., 1996] und repräsentiert ein Volumenelement auf einem Gitter als Entsprechung zu einem Pixel im 2D-Bild.

Mit der Verwendung von Voxelräumen geht eine Vielzahl positiver Eigenschaften einher. Operationen wie die Verschneidung oder Vereinigung von Voxelräumen können leicht durchgeführt werden. Zudem können in Voxeln unterschiedliche Informationen abgelegt werden, wie beispielsweise eine binäre Belegungsinformation, eine Belegungswahrscheinlichkeit, eine Dichte oder eine Farbinformation. Die Speicherung der Voxelinformationen kann entweder vollständig für alle gegebenen Voxel erfolgen (engl. Dense) oder auch nur für einzelne Elemente wie die belegten Voxel (engl. Sparse). Für Voxelräume mit geringen Belegungen können zur beschleunigten Abfrage Octrees eingesetzt werden [Foley et al., 1996]. Nachteilig an der Verwendung von Voxelräumen ist die begrenzte Genauigkeit, die von der gewählten Anzahl an Raumunterteilungen und damit der Voxelgröße abhängt. Objektflächen, die nicht an den Koordinatenachsen ausgerichtet sind, müssen in einem Voxelraum über Stufen angenähert werden. Je nach Winkel der Fläche und Größe eines einzelnen Voxels ergibt sich ein entsprechend großer Diskretisierungsfehler. Dreiecksnetze sind hierbei genauer. Das gleiche Problem tritt auch

bei gekrümmten Oberflächen auf. Durch die Erhöhung der Anzahl an Unterteilungen kann die Genauigkeit zwar verbessert werden, dennoch geben die Größe des Speichers und die nutzbaren Rechenkapazitäten eine Grenze vor. Der Begriff Voxelraum wird im Folgenden für die Zielstellung der Dissertation formal definiert.

Ein begrenzter orthogonaler Unterraum $[0,1]^3 \subset \mathbb{R}^3$, genannt Überwachungsraum, wird diskretisiert. Dies resultiert in einem Voxelraum wie in Abbildung 5.6 dargestellt. Der Unterraum wird dabei von zwei Raumpunkten $r_{\min}, r_{\max} \in \mathbb{R}^3$ aufgespannt, für die gilt: $r_{\min}^{(d)} < r_{\max}^{(d)}$ mit $d \in \{1,2,3\}$. Die Diskretisierung des Unterraums in gleich große Subelemente erreicht man, indem dieser entlang jeder Dimension in eine Anzahl äquidistanter Abschnitte (Intervalle) unterteilt wird, gegeben durch $s_{\text{steps}} \in \mathbb{N}^3$.

Damit lässt sich nach [Kuhn, 2012] der Voxelraum eindeutig über das Tripel VoxelSpace $:= (r_{\min}, r_{\max}, s_{\text{steps}})$ definieren. Ein Voxelraum besteht aus quader- oder würfelförmigen Zellen, den sogenannten Voxeln. Es sei mit $V := \{v_0, \ldots, v_{|V|-1}\}$ die endliche Menge aller zugehörigen $|V| \in \mathbb{N}$ Voxel v_i gegeben. Jedes Voxel v_i hat die Ausmaße s_{width} (Länge, Breite, Höhe) wie in Abb. 5.6 veranschaulicht. Die Komponenten von $s_{\text{width}} \in \mathbb{R}^3$ ergeben sich zu $s_{\text{width}}^{(d)} := (r_{\max}^{(d)} - r_{\min}^{(d)})/s_{\text{steps}}^{(d)}$.

Die Zuordnung eines Voxels zu dem Diskretisierungsgitter, das im Folgenden als **Voxelraumkoordinatensystem** (VKS) bezeichnet wird, sei gegeben durch eine Funktion $\text{idx} : V \to \mathbb{N}^3$. Diese liefert für ein Voxel v_i ein Tupel an Indizes zurück, die die Position des Voxels im VKS angeben. Der Koordinatenursprung des VKS befindet sich typischerweise im Punkt $r_{\min}$. Damit ergibt sich für die Voxelindizes in jeder Dimension $d \in \{1,2,3\}$ ein Definitionsbereich von $0 \leq \text{idx}(v_i)^{(d)} \leq s_{\text{steps}}^{(d)} - 1$ $(\forall v_i \in V)$. Jedes Tupel an Voxelindizes repräsentiert von einem Voxel den Eckpunkt mit den kleinsten Raumkoordinaten, gegeben im Weltkoordinatensystem. Für die weiteren Betrachtungen wird eine Funktion $\text{center} : V \to \mathbb{R}^3$ definiert, die das Zentrum (den Schwerpunkt) des Voxels im Ursprungsraum zurückliefert, gegeben durch $r_{v_i} := \text{center}(v_i)$.

Abbildung 5.7(a) zeigt die SIMERO-Roboterarbeitszelle sowie die in dieser Dissertation verwendete Modellierung statischer Objekte in Form von Dreiecksnetzen (engl. Meshes) in (b) und (c). Die Modelle wurden mit Blender erstellt [Stoychev, 2013]. Der Voxelraum ist in Abb. 5.7(d) als grünes Gitternetz dargestellt und hat eine Auflösung von $126 \times 110 \times 73$ Voxel bei einer Voxelseitenlänge von ca. 3,4 cm in allen drei Dimensionen.

Der Einfluss von (statischen) Objekten auf die Sichtbarkeit des Überwachungsraums kann durch sogenannte **Voxelsichtbarkeitsgrade** greifbarer gemacht werden. Der Begriff Sichtbarkeitsgrad wurde in Abschnitt 3.2 definiert und gibt an, wie viele Kameras freie Sicht auf einzelne Raumpunkte bzw. zusammenhängende Volumina von Raumpunkten haben. Übertragen auf eine Voxeldatenstruktur sei der Voxelsichtbarkeitsgrad definiert als der kleinste Sichtbarkeitsgrad aller im Voxel enthaltenen Raumpunkte. So-

mit entspricht der Voxelsichtbarkeitsgrad der Mindestanzahl an Kameras, von denen ein Voxel in Gänze gesehen wird. Ein Algorithmus zur Bestimmung der Voxelsichtbarkeiten kann in Anhang 9.4 nachgeschlagen werden.

Die Voxelsichtbarkeitsgrade 0 bis 7 für das verwendete Kamerasystem, das aus sieben Kameras besteht, gehen aus den Abbildungen 5.8 und 5.9 hervor. Dargestellt sind die Voxelsichtbarkeiten des Überwachungsraums für die Abwesenheit und Präsenz der statischen Objekte im Vergleich zueinander (linke Spalte respektive rechte Spalte). Die Bilder einer Zeile stehen jeweils für denselben Voxelsichtbarkeitsgrad, beginnend in der obersten Zeile von Abb. 5.8 mit dem Wert 0. Die Voxelsichtbarkeitsgrade der leeren Arbeitszelle (linke Spalten) werden alleinig durch die sensorische Abdeckung des Kamerasystems bestimmt. Die Anwesenheit statischer Objekte führt zu einer Reduktion der Voxelsichtbarkeitsgrade. Dies lässt sich daran erkennen, dass bei den niedrigeren Sichtbarkeitsgraden von 0 bis 3 die Voxelanzahl jeweils mit den statischen Objekten größer ist (rechte Spalte) als ohne sie (vgl. Abb. 5.8). Bei den höheren Voxelsichtbarkeitsgraden der Abb. 5.9 hingegen ist erkennbar, dass die Voxelanzahl bei gegebenen statischen Objekten geringer ist als im leeren Überwachungsraum. Die statischen Objekte führen zu weniger Voxeln mit hohen Sichtbarkeitsgraden und dafür zu mehr Voxeln mit geringeren Sichtbarkeitsgraden.

Diese quantitative Verschiebung der Voxelsichtbarkeitsgrade kann zu einer qualitativen Verminderung der Rekonstruktionsgüte führen. Die Stärke dieses Effekts hängt allerdings von den geometrischen Objekteigenschaften, der Anzahl an Objekten sowie ihren Raumpositionen ab und ist ortsabhängig. Die Voxelsichtbarkeitsgrade ändern sich natürlich ebenfalls mit allen dynamischen Objekten, die sich im Überwachungsraum bewegen. Anhand einer Simulationsumgebung könnte man die Änderung dieser auch für dynamische Objekte zu jedem Zeitpunkt k bestimmen. Damit wäre es möglich, Rückschlüsse für erreichbare Approximationsgüten der 3D-Rekonstruktionen in Abhängigkeit von den Voxelsichtbarkeitsgraden für bestimmte Situationen zu ziehen.

Generell gilt: Je mehr Objekte sich im Überwachungsraum befinden, desto stärkere Einbußen in Bezug auf die Sichtbarkeit und Rekonstruktionsqualität der einzelnen Objekte sind im Mittel zu erwarten. Im Vorfeld wurde bereits beschrieben, dass sich die Rekonstruktionsgüte oft mit einer Erhöhung der Kameraanzahl verbessern lässt. Umgekehrt verringert eine zunehmende Präsenz von Objekten im Überwachungsraum die Rekonstruktionsqualität der einzelnen Objekte aufgrund der lokalen Herabsetzung von Sichtbarkeitsgraden bei einer prinzipiell gleichbleibenden sensorischen Erfassung des Kamerasystems.

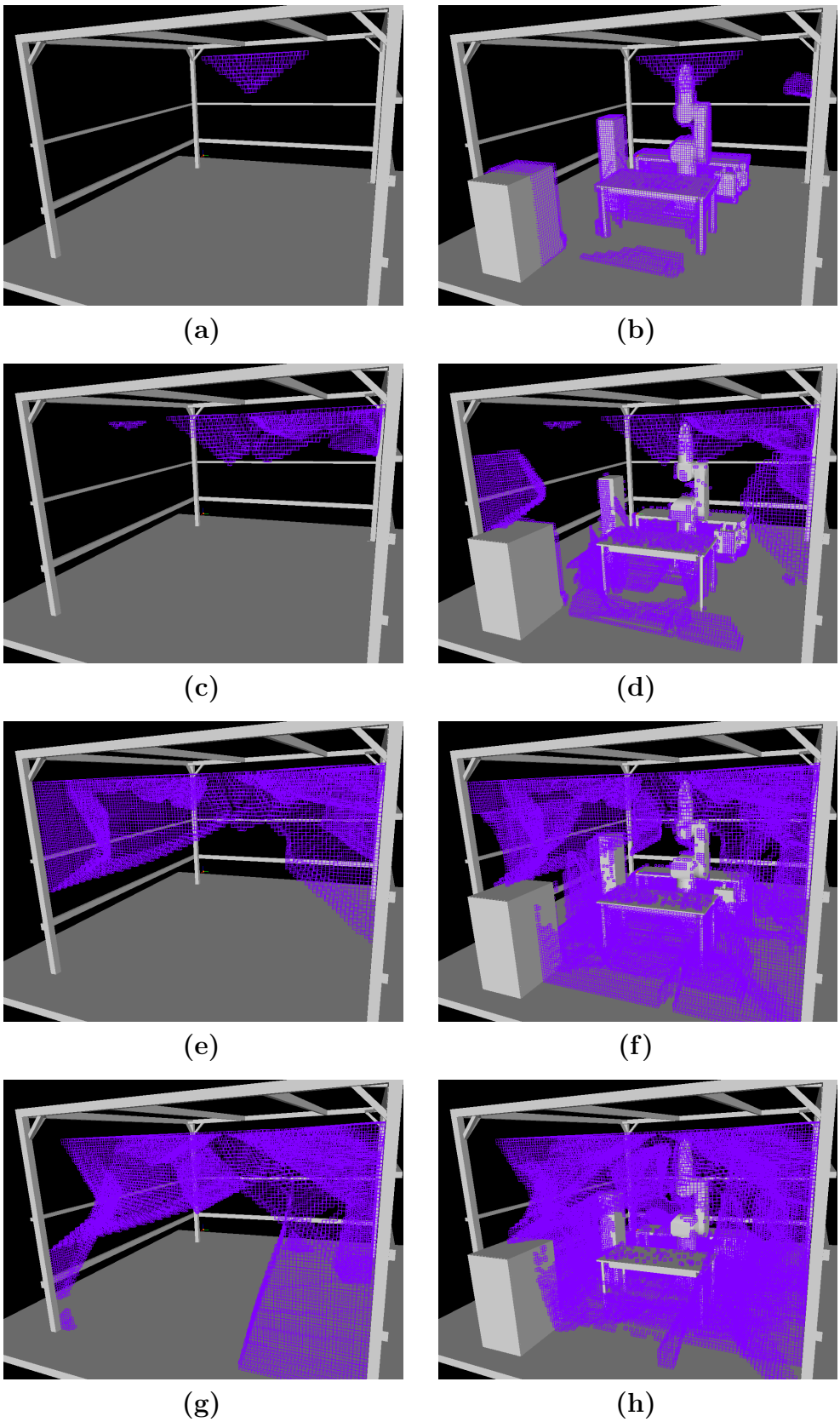


Abb. 5.8: Überwachungsraum mit und ohne statische Objekte. Zu sehen sind die Voxel mit Sichtbarkeitsgraden von 0 (Zeile 1) bis 3 (Zeile 4).

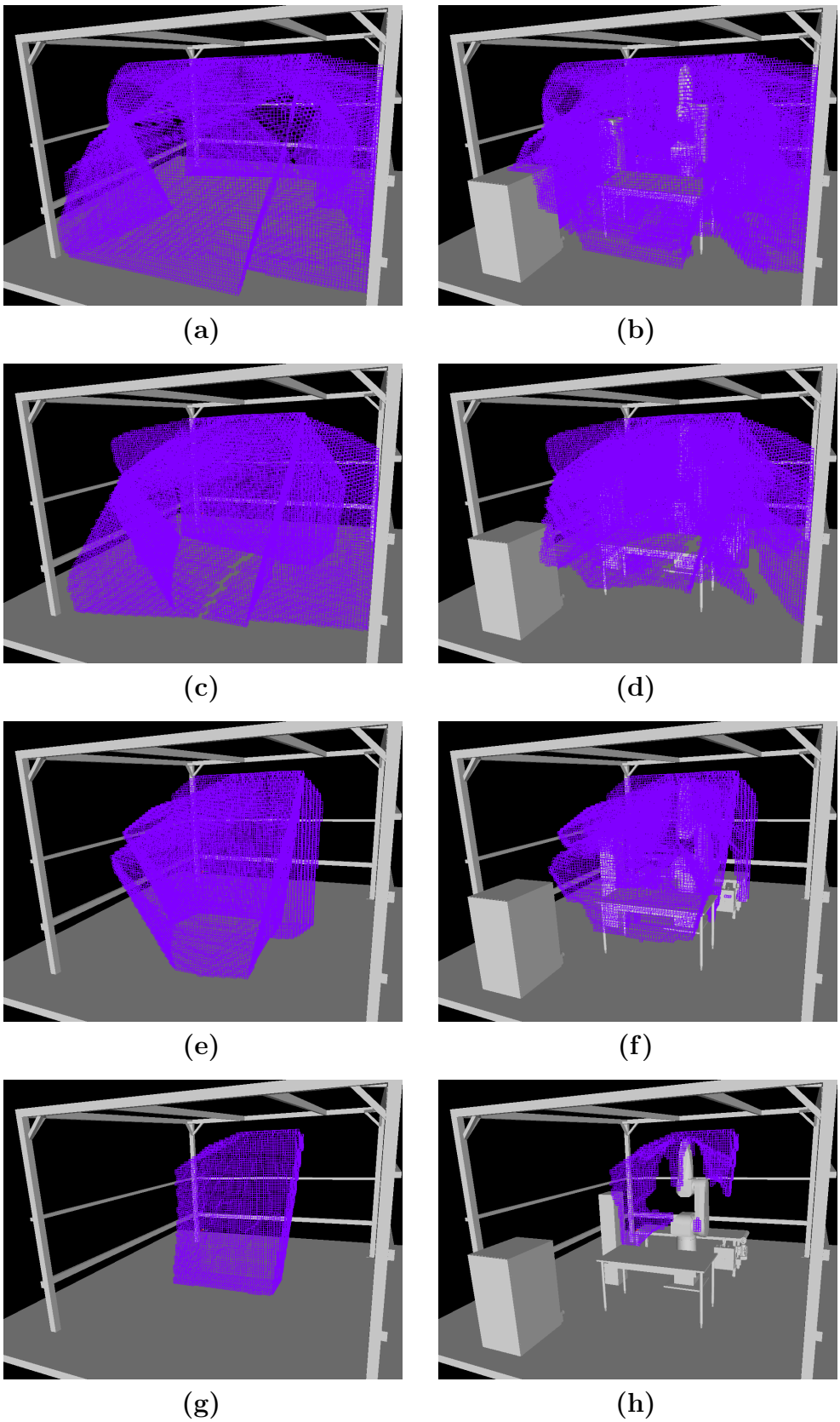


Abb. 5.9: Überwachungsraum mit und ohne statische Objekte. Zu sehen sind die Voxel mit Sichtbarkeitsgraden von 4 (Zeile 1) bis 7 (Zeile 4).

5.4.2 Kamerabilder

Zur Raumüberwachung wird ein Multi-View-Farbkamerasystem eingesetzt, wie in Abschnitt 3.1 definiert. Entsprechend sei $C := \{c_1, \ldots, c_{|C|}\}$ eine endliche Menge an $|C| \in \mathbb{N}$ kalibrierten und synchronisierten Farbkameras c_j. Die Funktion $\omega : C \to \mathbb{R}^3$ gibt die Position des Zentralprojektionspunkts $e_{c_j} := \omega(c_j)$ einer Kamera an. Es wird ein Lochkameramodell ohne Verzerrungen angenommen. Zudem wird für die folgenden Beschreibungen vorausgesetzt, dass jedes Voxel von allen Kameras prinzipiell gesehen werden könnte. Dadurch können einige Implementierungsdetails vernachlässigt und die Darstellung vereinfacht werden.

Jede Kamera enthält eine Anzahl an Sensorelementen, genannt Pixel, definiert als eine endliche Menge $P_{c_j} := \{p_1, \ldots, p_{|P_{c_j}|}\}$ mit $|P_{c_j}| \in \mathbb{N}$ Pixeln $p_l \in P_{c_j}$. Die Menge der Pixel aller Kameras sei gegeben als $P := \cup_{c_j \in C} P_{c_j}$. Zu jedem Zeitpunkt liefert eine Kamera c_j für jedes Pixel diskrete Daten als eine Abbildung $\text{image}_{\{\text{col,bin,dep}\},c_j} : P_{c_j} \to D_{\{\text{col,bin,dep}\}}$, mit $D_{\text{col}} := [1, 2^m]^3 \in \mathbb{N}^3$ (Farbbild mit m bit Farbauflösung in 3 Kanälen), $D_{\text{bin}} := \{0, 1\}$ (Binärbild) sowie $D_{\text{dep}} := \mathbb{R}$ (Tiefenbild). Auf die Farbbilder $I_{\text{col}} := \{\text{image}_{\text{col},c_1}, \ldots, \text{image}_{\text{col},c_{|C|}}\}$ aller Kameras des Kamerasystems wird ein BS-Verfahren angewendet, welches eine Anzahl segmentierter binärer Silhouettenbilder erzeugt, gegeben als $I_{\text{bin}} := \{\text{image}_{\text{bin},c_1}, \ldots, \text{image}_{\text{bin},c_{|C|}}\}$. Das BS-Verfahren benötigt für jede Kamera mindestens ein Referenzbild, das zu einem Zeitpunkt k^{ref} aufgenommen wird. Zu diesem Zeitpunkt dürfen keine Objekte von Interesse (Personen) im Überwachungsraum präsent sein. Die Menge an Referenzbildern aller Kameras wird als $I_{\text{col}}^{\text{ref}}$ bezeichnet. Es kommt ein BS-Verfahren der OpenCV-Bibliothek zum Einsatz [Pavlenko, 2012]. Dieses kann aber durch eine beliebige andere Methode ersetzt werden, welche für die gegebenen Rekonstruktionsbedingungen der spezifischen Umgebung geeignet ist, z. B. ein adaptives Verfahren bei sich stark verändernden Beleuchtungsbedingungen. Das Vorgehen eines BS-Verfahrens wurde bereits ausführlich in Abschnitt 2.2 beschrieben.

Nach Ausführung des BS-Verfahrens ist der Wert eines jedes Pixels $p_l \in P_{c_j}$ entweder gegeben als $\text{image}_{\text{bin},c_j}(p_l) = 0$, wenn p_l als Hintergrund (engl. Background) klassifiziert wurde, oder als $\text{image}_{\text{bin},c_j}(p_l) = 1$ bei einer Klassifizierung von p_l als Vordergrund (engl. Foreground). Die segmentierten binären Silhouettenbilder I_{bin} werden als Eingabe für die VH verwendet.

5.4.3 Visuelle Hülle

Das Konzept der VH wurde in Abschnitt 5.3 erläutert und die Problematik verdeckender statischer Objekte bei der Rekonstruktion in Abschnitt 5.3.3 veranschaulicht. Auch

der Lösungsansatz von [Kuhn und Henrich, 2009] wurde beschrieben, der in dieser Dissertation Anwendung findet. Für ein besseres Verständnis wird die Darstellung aus [Ober-Gecks et al., 2016] übernommen und der eingesetzte Algorithmus im Vergleich mit einem Standardverfahren in zwei Schritten hergeleitet.

Standardverfahren einer Visuellen Hülle

Ziel der Berechnung einer VH bei Überwachungsszenarien ist oftmals die Rekonstruktion a priori unbekannter Objekte wie Personen. Zur Erzeugung segmentierter Silhouettenbilder, die als Eingabe zur Rekonstruktion dienen, wird dabei ein BS-Verfahren eingesetzt. Dieses verarbeitet Bilder, die zu diskreten Zeitpunkten von allen Kameras synchron aufgezeichnet werden. Die erzeugte VH ist eine Teilmenge aller Voxel des Voxelsraums V, gegeben als $V_{\mathrm{VH}} \subset V$. Die Voxel der VH sollen zusammengenommen möglichst genau die Objektgeometrien und damit auch die Raumbelegung im Überwachungsraum repräsentieren.

Mit $\Phi_{v_i,c_j} \subset P_{c_j}$ sei die Menge der Projektionspixel eines Voxels v_i in der Kamera c_j definiert. Ein Standardverfahren zur Berechnung einer VH wird in Algorithmus 5.1 dargestellt. Ein Voxel wird verworfen (gecarvt), sobald ein Projektionspixel gefunden wird, der als Hintergrund (engl. Background) klassifiziert wurde (Zeilen 7 und 8). Das Voxel ist damit nicht Teil der VH und wird als leer oder transparent markiert.

Algorithmus 5.1 Standardverfahren für eine Visuelle Hülle

```
1:  procedure StandardVisualHull(V, C, I_bin)
2:      V_VH ← V                                  ▷ initialize visual hull with full voxelspace
3:      for all v_i ∈ V_VH do                                                  ▷ for each voxel
4:          for all c_j ∈ C do                                                 ▷ for each camera
5:              Φ_{v_i,c_j} ← projectVoxelToCam(v_i, c_j)
6:              for all p_l ∈ Φ_{v_i,c_j} do                  ▷ for each projection pixel of voxel
7:                  if getImageValue(p_l, I_bin, c_j) = 0 then          ▷ if pixel is background
8:                      V_VH ← V_VH \ {v_i}                                        ▷ carve voxel
9:                  end if
10:             end for
11:         end for
12:     end for
13:     return V_VH                                    ▷ return set of uncarved voxels
14: end procedure
```

Die resultierende VH setzt sich aus Voxeln $v_i \in V$ zusammen, für welche die Bedingung in Formel (5.1) erfüllt wird. Das heißt, alle Voxel der VH projizieren vollständig auf Pixel, die in den segmentierten Bildern I_{bin,c_j} als Vordergrund (engl. Foreground) klassifiziert worden sind und damit zu den Silhouettenpixeln gehören.

$$\forall c_j \in C : \forall p_l \in \Phi_{v_i,c_j} : \text{image}_{\text{bin},c_j}(p_l) = 1 \tag{5.1}$$

Visuelle Hülle mit verdeckenden Hintergrundobjekten

Die genannte Annahme von Algorithmus 5.1, nach der die Projektion der Objekte in die Kameras vollständig in den Silhouetten der segmentierten Bilder I_{bin,c_j} enthalten sein müssen, um eine korrekte Rekonstruktion zu ermöglichen, kann nicht garantiert werden. Fehler bei der Objektdetektion können auftreten (vgl. Abschnitt 2.2). Statische Objekte (Hintergrundobjekte), die sich während der Referenzbilderstellung im Überwachungsraum befinden, sind normalerweise nicht in den Objektsegmenten enthalten. Fehlt die Segmentierung der Hintergrundobjekte, so hat dies zwei Effekte: Einerseits werden die Hintergrundobjekte nicht rekonstruiert. Andererseits können sie die Objekte von Interesse teilweise oder vollständig verdecken, was in fehlenden oder unvollständigen Silhouettenbildern resultiert, wie bereits in Abb. 5.4 dargestellt. Das Ergebnis eines solchen Falls ist, dass die Formel (5.1) fälschlicherweise nicht wahr ist und die rekonstruierte VH die Objekte von Interesse auch nicht vollständig enthält.

Möchte man garantieren, dass keine Teile der dynamischen Objekte in der Rekonstruktion fehlen, so besteht eine Möglichkeit darin, sicherzustellen, dass alle Objekte im Überwachungsraum in den Bildern segmentiert werden. Dies kann bewerkstelligt werden, indem man die Referenzbilder $I_{\text{col}}^{\text{ref}}$ ohne Anwesenheit verdeckender Hintergrundobjekte im Überwachungsraum aufzeichnet. Dies ist allerdings nur bedingt praktikabel. Eine alternative Lösung hierfür wurde in Abschnitt 5.3.3 vorgestellt: Bei der Verwendung von Occlusion Masks [Ladikos et al., 2008] werden (meist manuell) Binärmasken für die Hintergrundobjekte erzeugt, die zu den Silhouettenbildern hinzugefügt werden. Die in dieser Arbeit favorisierte Lösung von [Kuhn und Henrich, 2009] verzichtet hingegen auf die Segmentierung von Hintergrundobjekten und verwendet stattdessen Tiefeninformationen von diesen, die in den Rekonstruktionsprozess integriert werden. Damit kann eine VH mit besserer geometrischer Approximation rekonstruiert werden (vgl. Abb. 5.5).

Die genaue Vorgehensweise wird nun beschrieben. Gegeben sei $O := \{o_1, \ldots, o_{|O|}\}$ eine endliche Menge von $|O| \in \mathbb{N}$ statischen a priori bekannten Objekten $o_q \in O$ im Überwachungsraum. All diese Objekte seien geometrisch modelliert (z. B. durch ein Dreiecksnetz) und ihre Lage im Raum sei bekannt. Mithilfe dieser Modelle werden synthetische Tiefenbilder $I_{\text{dep}} = \{\text{image}_{\text{dep},c_1}, \ldots, \text{image}_{\text{dep},c_{|C|}}\}$ der statischen Objekte

für alle Kameras in einem Offline-Schritt erzeugt, welche dieselbe Auflösung besitzen wie die realen Kamerabilder. Ein virtuelles Tiefenbild kodiert die maximale freie Sicht jedes Pixels $p_l \in P_{c_j}$, die durch die nächste Objektoberfläche eines Hintergrundobjekts begrenzt wird. Wenn ein Kamerapixel p_l nicht auf ein Hintergrundobjekt schaut, so ist der zugehörige Tiefenwert auf unendlich gesetzt.

Im Rekonstruktionsprozess wird die Sichtbarkeit der Voxel in ihren Projektionspixeln mithilfe der synthetischen Tiefenbilder ausgewertet. Nur Projektionspixel, die eine freie Sicht auf einen betrachteten Voxel haben, dürfen auch für die Entscheidung des Entfernens (engl. Carving) basierend auf den Silhouettenbildern herangezogen werden. Gegeben sei der Zentralprojektionspunkt $e_{c_j} \in \mathbb{R}^3$ einer Kamera c_j, das Zentrum eines Voxels $r_{v_i} \in \mathbb{R}^3$ sowie eine Konstante $d_v \in \mathbb{R}$, die die Hälfte der Diagonale eines Voxels beschreibt, um konservativ zu sein. Diesbezüglich lässt sich die Formel (5.1) so erweitern, dass für alle Voxel $v_i \in V_{\mathrm{VH}}$, die zur VH gehören, die Bedingung in Formel (5.2) gilt. Danach sind alle Objekte Teil der Rekonstruktion. Der Betrag der Differenz der Positionsvektoren des Voxelzentrums und des Zentralprojektionspunkts wird mit der L_2-Norm berechnet und entspricht der Distanz zwischen beiden Positionen.

$$\forall p_l \in \Phi_{v_i,c_j} : \ (\mathrm{image}_{\mathrm{bin},c_j}(p_l) = 1) \ \vee \ (|r_{v_i} - r_{c_j}| + d_v \geq \mathrm{image}_{\mathrm{dep},c_j}(p_l)) \tag{5.2}$$

In Algorithmus 5.1 muss die Zeile 7 so angepasst werden, dass die Bedingung aus Formel (5.3) erfüllt wird.

$$(|r_{v_i} - r_{c_j}| + d_v < \mathrm{image}_{\mathrm{dep},c_j}(p_l)) \ \wedge \ (\mathrm{image}_{\mathrm{bin},c_j}(p_l) = 0) \tag{5.3}$$

Konservative Visuelle Hülle mit verdeckenden Hintergrundobjekten

Die Formeln (5.1) und (5.2) berücksichtigen noch nicht die Diskretisierungsfehler, die bei einer Voxelraum-Datenstruktur auftreten. Diese können jedoch den Rand der VH deutlich beeinflussen. Bisher genügt ein einzelner Projektionspixel $p_l \in \Phi_{v_i,c_j}$ einer Kamera mit einem Wert von $\mathrm{image}_{\mathrm{bin},c_j}(p_l) = 0$ (Background), um ein Voxel zu verwerfen, selbst wenn alle anderen Projektionspixel $p_k \in \Phi_{v_i,c_j}$ mit $l \neq k$ den Wert $\mathrm{image}_{\mathrm{bin},c_j}(p_k) = 1$ (Foreground) aufweisen. Damit werden im Ergebnis der Rand sowie das Innere der VH nur von Voxeln gebildet, die vollständig auf die Silhouettenpixel in allen Kameras projizieren. Um jedoch zu ermöglichen, dass alle Objekte bei idealen Silhouettenbildern vollständig in der VH enthalten sind, wird die folgende konservative Formulierung benötigt:

So wie in Formel (5.4) angegeben, darf ein Voxel nur dann entfernt werden, wenn für dieses alle Projektionspixel Φ_{v_i,c_j} von mindestens einer Kamera vollständig als

Algorithmus 5.2 Verfahren für eine konservative VH mit Verdeckungsbehandlung

```
1: procedure ConservativeVisualHull(V, C, I_bin, I_dep)
2:     V_VH ← V                          ▷ initialize visual hull with all voxels from voxelspace
3:     for all v_i ∈ V_VH do                                              ▷ for each voxel
4:         for all c_j ∈ C do                                             ▷ for each camera
5:             flag_foreground ← nil                  ▷ a flag with three values is required
6:             flag_occludedPixelExist ← false
7:             Φ_{v_i,c_j} ← projectVoxelToCam(v_i, c_j)
8:             for all p_l ∈ Φ_{v_i,c_j} do                ▷ for each projection pixel of the voxel
9:                 if |r_{v_i} − r_{c_j}| + d_v < getImageValue(p_l, I_dep, c_j) then ▷ if voxel is
10:                                                                                    visible
11:                    if getImageValue(p_l, I_bin, c_j) = 1 then ▷ if pixel is foreground
12:                        flag_foreground ← true
13:                    end if
14:                    if flag_foreground = nil then      ▷ if first projection pixel in c_j
15:                                                                          is background
16:                        flag_foreground ← false
17:                    end if
18:                else                                  ▷ voxel is not visible in pixel
19:                    flag_occludedPixelExist ← true
20:                end if
21:            end for
22:            if (flag_foreground = false) ∧
23:               (flag_occludedPixelExist = false) then
24:                V_VH ← V_VH \ {v_i}                                    ▷ carve voxel
25:            end if
26:        end for
27:    end for
28:    return V_VH
29: end procedure
```

Hintergrund klassifiziert werden. Das ist gleichbedeutend damit, dass kein verdecktes oder belegtes Pixel (Foreground) in der Kamera für das Voxel erlaubt ist.

$$\exists c_j \in C : \forall p_l \in \Phi_{v_i,c_j} : \ (\text{image}_{\text{bin},c_j}(p_l) = 0) \ \wedge \ (|r_{v_i} - r_{c_j}| + d_v < \text{image}_{\text{dep},c_j}(p_l)) \tag{5.4}$$

Umgekehrt genügt ein belegtes oder ein verdecktes Pixel in einer einzelnen Kamera, um das Voxel in der Rekonstruktion zu belassen, wie in Formel (5.5) formuliert.

$$\forall c_j \in C : \exists p_l \in \Phi_{v_i,c_j} : \ (\text{image}_{\text{bin},c_j}(p_l) = 1) \ \vee \ (|r_{v_i} - r_{c_j}| + d_v \geq \text{image}_{\text{dep},c_j}(p_l)) \tag{5.5}$$

Der Algorithmus 5.2 berücksichtigt sowohl die Verdeckungsproblematik als auch die Fehler der Voxelraumdiskretisierung, die den Rand der VH beeinflussen. Diese obere Abschätzung einer VH wird als „konservative Visuelle Hülle mit Verdeckungsbehandlung“ bezeichnet. Ein Entwurf des Algorithmus für die GPU aus [Zwicker, 2013] wird in Abschnitt 9.2.4 beschrieben.

5.5 Rekonstruktionsbestandteile

Die im vorangegangenen Abschnitt vorgeschlagene „konservative Visuelle Hülle mit Verdeckungsbehandlung“ kann auch als Vereinigung betrachtet werden von: einer Visuellen Hülle der dynamischen Objekte (ohne Präsenz statischer Objekte) und einer Tiefenhülle der statischen Objekte (ohne Präsenz dynamischer Objekte). Mit dieser separaten Betrachtung von VH und TH lassen sich die Rekonstruktionsbestandteile gut verdeutlichen, was im Folgenden geschehen soll. Auch hilft diese Unterteilung bei der Herleitung der Voxelzustände des Abschnitts 5.7.

In Abb. 5.10 wird eine Überwachungssituation mit drei Objekten in einer 2D-Aufsicht gezeigt. In der ersten Zeile werden zwei Kameras zur Rekonstruktion eingesetzt und in der zweiten Zeile drei Kameras. In den Teilabbildungen (a) und (d) der linken Spalte ist eine VH dargestellt, wie sie generiert werden würde, wenn die Objekte dynamisch wären (rot). In (c) und (f) der rechten Spalte wird eine TH gezeigt, wie sie im Falle statischer Objekte (blau) unter Einsatz von Tiefenbildern entstehen würde. Die mittlere Spalte veranschaulicht die kombinierte Rekonstruktion einer VH und einer TH zweier dynamischer Objekte und eines statischen Objekts.

Die Rekonstruktion ist eine Zusammensetzung aller schwarz eingerahmten Bereiche. Diese bestehen aus Verdeckungsvolumina (lila schraffiert), sogenannten Artefakten (grün kariert) sowie den sichtbaren Objektoberflächen. Letztere gehören nach eigener Definition zwar weder zu den Artefakten noch zu den Verdeckungsvolumina, sind jedoch aus Gründen der Übersichtlichkeit nicht extra hervorgehoben. Sichtbare Objektoberflächen und Verdeckungsvolumina sind immer Bestandteil der Rekonstruktion eines nichtleeren Raums. Die ausführliche Definition des Begriffs Verdeckungsvolumen wurde in Abschnitt 3.2.2 gegeben. Zur Wiederholung: Die freie Sicht jeder Kamera wird durch die nächstgelegenen sichtbaren Objektoberflächenpunkte unterbrochen, wodurch die Sichtstrahlen einer Kamera in verdeckte und sichtbare Strahlabschnitte unterteilt werden. Topologisch zusammenhängende Mengen verdeckter Raumpunkte werden in dieser Dissertation als Verdeckungsvolumina bezeichnet. Verdeckungsvolumina der VH und der TH unterscheiden sich nicht. Sämtliche Bereiche hinter den sichtbaren Ober-

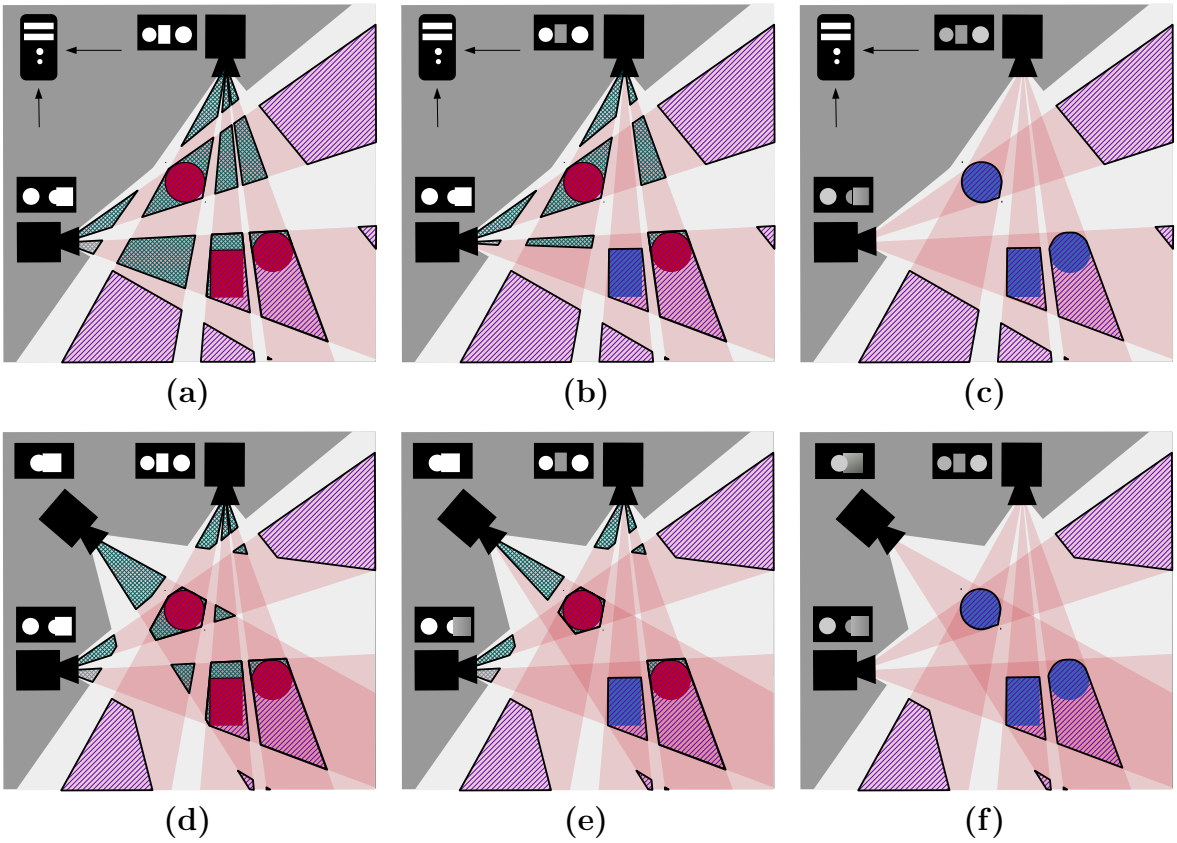


Abb. 5.10: Artefakte (grün kariert) und Verdeckungsvolumina (lila schraffiert) im Vergleich für eine Visuelle Hülle und eine Tiefenhülle. Es sind Rekonstruktionen mit zwei und drei Kameraperspektiven dargestellt (1. Zeile bzw. 2. Zeile). Zu sehen sind eine VH in (a) und (d), eine TH in (c) und (f) sowie die Kombination aus einer VH von dynamischen Objekten (rot) und aus einer TH von statischen Objekten (blau) in (b) und (e).

flächen werden rekonstruiert, unabhängig davon, ob Farb-, Intensitäts- oder Tiefenbilder zugrunde liegen. Dies geht auch aus Abb. 5.10 hervor.

Verschneidet man die Verdeckungsvolumina einzelner Kameras miteinander, so geht dies mit einer Verkleinerung der verdeckten Volumina einher, wenn die mangelnde Sichtbarkeit einer Kamera durch andere Kameras kompensiert werden kann. Für ein gesamtes Kamerasystem gelten nur die Raumpunkte als verdeckt, welche von keiner einzigen Kamera gesehen werden (Sichtbarkeitsgrad 0). In Abb. 5.10 wird dies bei Betrachtung der unterschiedlichen Rekonstruktionen für zwei respektive drei Kameras deutlich. Im besten Fall würden die rekonstruierten Verdeckungsvolumina nur aus belegten Raumpunkten bestehen. Dann würden die Objektgeometrien maximal gut approximiert werden. Dies ist jedoch meist nicht der Fall, wie auch in den gezeigten Beispielen der Abbildung 5.10: Verdeckte leere Raumpunkte sind oft Teil der Rekonstruktion. Da sie bei dynamischen Objekten dieser Arbeit nicht identifiziert und lokalisiert werden können, ist es auch nicht möglich, diese aus der Rekonstruktion zu entfernen. Stattdessen müssen sie als zu den Objekten zugehörig behandelt werden.

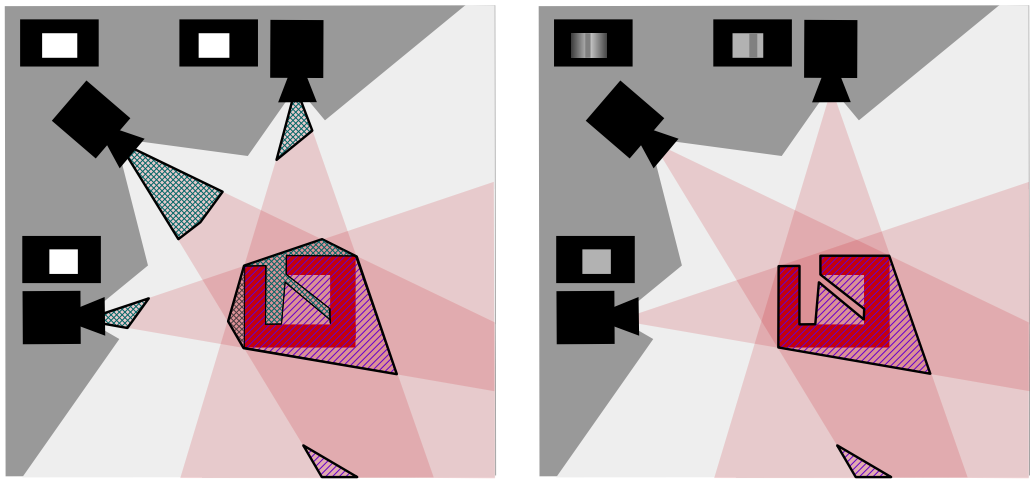


Abb. 5.11: Veranschaulichung der Visuellen Hülle eines nicht-konvexen Objekts (a) im Vergleich zu einer idealen Tiefenhülle vom selben Objekt (b). Die Objektoberfläche ist nicht vollständig erfassbar. Auch bei der Tiefenhülle bleiben leere verdeckte Volumina bestehen (lila schraffiert). Im Vergleich zur Tiefenhülle können mit der Visuellen Hülle Objekte oft schlechter aufgrund rekonstruierter Artefakte (grün kariert) approximiert werden.

Neben leeren verdeckten Raumpunkten werden bei der VH auch leere sichtbare Raumpunkte rekonstruiert, die sich vor den sichtbaren Objektoberflächen befinden (in Abb. 5.10 grün kariert). Zusammenhängende Raumpunkte dieser Art werden als **Artefakte** bezeichnet. Sie gehören nicht zu den Verdeckungen, weil sie nicht verdeckt sind. Artefakte entstehen bei der VH, da die rückprojizierten Sichtstrahlen der Vordergrundpixel immer vollständig miteinander verschnitten werden müssen. Es existiert keine Information darüber, in welchem Abstand die verdeckten Volumina im Raum beginnen. Bei der TH hingegen liegen diese Informationen vor. Deshalb treten solche Artefakte bei der TH auch nicht auf, wie in Abb. 5.10 für die statischen Objekte (blau) dargestellt.

Betrachtet man Abb. 5.10 genauer, so sind rekonstruierte Volumina (Artefakte und Verdeckungsvolumina) erkennbar, die keinerlei Objekte enthalten. Wie bereits erwähnt, werden solch leere Volumina in der Literatur auch als Pseudoobjekte bezeichnet. Derartige Störungen werden in der Dissertation von [Kuhn, 2012] durch die Anwendung von Plausibilisierungsfunktionen entfernt, unter der Annahme, dass sie dann keine Person enthalten können, wenn sie ein Mindestvolumen unterschreiten. Die Einsetzbarkeit der Plausibilisierungsfunktionen setzt ungestörte Silhouettenbilder und eine Modellierung aller statischen Objekte im Raum voraus. Problematischer für das Personen-Tracking sind jedoch jene leeren Volumina, die mit belegten Volumina topologisch zusammenhängen. Diese führen zu vergrößerten Objektgeometrien, was eine korrekte Einpassung von Objektmodellen erschwert. Ohne die Verwendung von Zusatzwissen, das meist nicht zur Verfügung steht, lassen sie sich jedoch nicht entfernen.

Zu den bisherigen Ausführungen noch ein ergänzender Aspekt: Die erreichbare geometrische Approximation ist nicht alleinig eine Frage der Kameratechnologie und Anzahl an Sensoren, sondern auch eine Frage davon, ob eine Platzierung der Sensoren so erfolgen kann, dass die Objektoberflächen vollständig gesehen werden können. Dies hängt auch von den Objektgeometrien ab. In Abbildung 5.11 ist ein nicht-konvexes Beispiel dargestellt, bei welchem ein Teil der Objektoberfläche sowohl bei der VH als auch bei der TH verdeckt ist und nur die äußere Objektgeometrie richtig erfasst werden kann. Auch ist es für ein Kamerasystem oft nicht möglich, ein gegenseitiges Verdecken der Objekte zu kompensieren, selbst wenn dieses aus vielen Kameras besteht. Prinzipiell gilt: Ein leerer Raumpunkt, für den in alle Richtungen nur Objektoberflächen sichtbar sind, kann nicht durch Kameras rekonstruiert werden, die sich außerhalb der „konvexen Hülle“ des Objekts befinden.

5.6 Betrachtung der Rekonstruktionsbestandteile als Mengen

In diesem Abschnitt werden die Bestandteile eines rekonstruierten Volumens als Mengen betrachtet, in Vorbereitung auf die sich in Abschnitt 5.7 anschließende Beschreibung der Voxelzustände, die für das Tracking zur Integration des Wissens zu den statischen Objekten und ihren Verdeckungsvolumina relevant sind. Dabei wird die separate Darstellung einer VH der dynamischen Objekte und einer TH der statischen Objekte fortgeführt. Zunächst wird auf die Mengen anhand von Beispielbildern eingegangen. Anschließend werden die Mengen formal beschrieben.

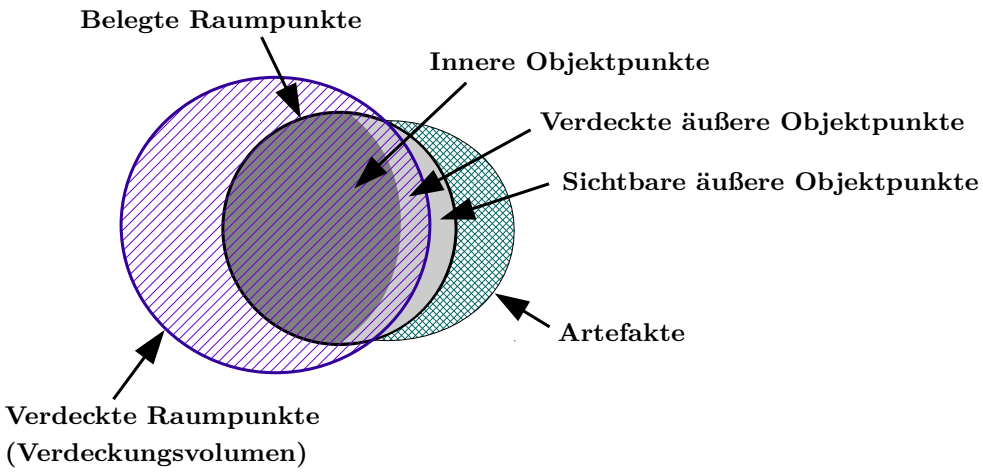


Abb. 5.12: Bestandteile einer rekonstruierten Visuellen Hülle, dargestellt als Raumpunktmengen.

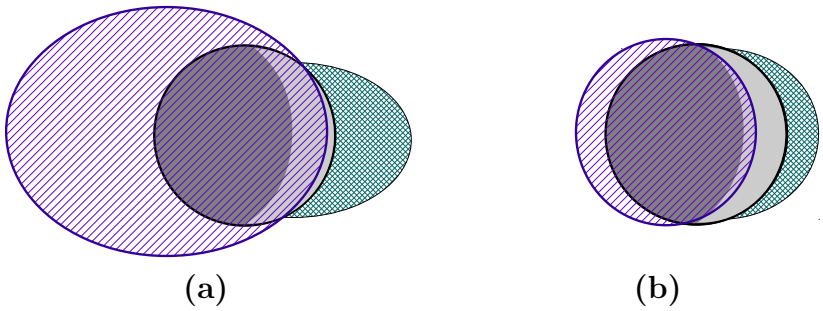


Abb. 5.13: Zwei verschiedene Rekonstruktionen einer Visuellen Hülle, dargestellt als Raumpunktmengen. Die Rekonstruktion mit einer geringen Anzahl an Kameraperspektiven (a) führt zu einer schlechteren Approximation der Objektgeometrien als die Verwendung von einer hohen Anzahl an Kameraperspektiven (b), was aus den Anteilen der Raumpunktmengen hervorgeht.

In Abbildung 5.12 ist die VH in verschiedene Mengen unterteilt. Eine rekonstruierte VH enthält im Idealfall die Menge aller belegten Raumpunkte der Objekte vollständig, die in Abb. 5.12 als schwarz umrandeter und grau ausgefüllter Kreis dargestellt ist. Dazu gehören die äußeren Objektpunkte (hellgraue Fläche) sowie die inneren Objektpunkte (dunkelgraue Fläche). Die äußeren Objektpunkte, auch Oberflächenpunkte genannt, können sichtbar oder verdeckt sein, so wie in Abb. 5.12 veranschaulicht. Innere Objektpunkte sind immer verdeckt. Sichtbare Oberflächenpunkte erzeugen ein Signal in den Kameras.

Eine erhöhte Oberflächensichtbarkeit geht in der Regel mit einer Reduktion der Verdeckungsvolumina (in Abb. 5.12 lila schraffiert) einher. Verdeckungsvolumina enthalten im Idealfall sämtliche inneren Objektpunkte. Zusätzlich können sie aus verdeckten Oberflächenpunkten sowie leeren verdeckten Raumpunkten bestehen. In Abb. 5.13(b) ist das Volumen dieser Menge im Vergleich zu Abb. 5.13(a) verringert, weil eine höhere Anzahl an Kameraperspektiven zur Verfügung steht. Im Beispiel von Abbildung 5.14(c) sind sämtliche Objektoberflächen für das Kamerasystem vollständig sichtbar. Damit

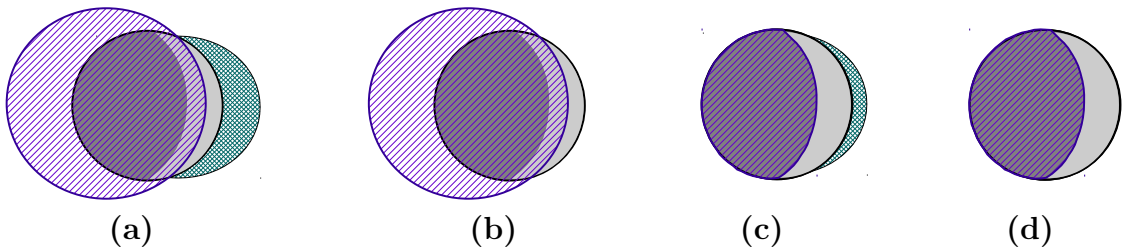


Abb. 5.14: Verschiedene Rekonstruktionen, dargestellt als Raumpunktmengen. Eine Rekonstruktion mit wenigen Kameraperspektiven in (a) und (b) ist im Vergleich zu einer Rekonstruktion mit vielen Kameraperspektiven in (c) und (d) zu sehen. Bei einer Visuellen Hülle (a) und (c) entstehen Artefakte (grün schraffiert), die bei einer idealen Tiefenhülle (b) und (d) nicht auftreten. Die Verdeckungsvolumina sind jedoch jeweils identisch.

sind leere verdeckte Raumpunkte nicht Teil der Rekonstruktion (Diskretisierungsfehler und andere Störeffekte außen vorgelassen). Die Sichtbarkeit der Objektoberflächen ist besonders wichtig, wenn Merkmale von diesen erfasst werden sollen. So verwendet das in Abschnitt 9.2 vorgestellte Verfahren einer Photohülle Oberflächenfarben zur Rekonstruktion der Geometrien. Farbinformationen werden zudem auch oft in Tracking-Verfahren ausgewertet, um zwischen verschiedenen Objekten differenzieren zu können. Daher besteht häufig ein Interesse darin, die Oberflächen möglichst vollständig zu erfassen.

In Abb. 5.14 wird nochmals die VH der TH gegenübergestellt (in (a) und (c) respektive (b) und (d)). Erkennbar ist hierbei, wie oben erwähnt, dass Artefakte (leere sichtbare Raumpunkte) rekonstruktionsbedingt nur bei der VH auftreten und nicht bei der TH. Dennoch können die Artefakte situationsabhängig recht gering ausfallen, sodass die geometrische Approximation nur wenig beeinträchtigt wird und auch die VH nahe an eine TH herankommt.

Zur folgenden formalen Beschreibung der Mengen werden verschiedene Definitionen aus Kapitel 3 übernommen: Es sei ein Überwachungsraum $R \subset \mathbb{R}^3$ im Weltkoordinatensystem mit den Koordinaten x, y, z gegeben. O_k sei die Menge aller Objekte o zu einem betrachteten Zeitpunkt k mit $O_k := O_{\text{stat}} \dot{\cup} O_{\text{dyn},k}$ (disjunkte Vereinigung), wobei mit O_{stat} die Menge aller statischen Objekte bezeichnet wird und mit $O_{\text{dyn},k}$ die Menge aller dynamischen Objekte. Statische Objekte stehen als modellierte 3D-Objekte zur Verfügung. Sie befinden sich bereits während der Referenzbilderstellung des BS-Verfahrens im Raum und sind im Hintergrundmodell kodiert. Dynamische Objekte betreten den Überwachungsraum erst im Überwachungsmodus und sollen von der Objektdetektion erfasst werden.

Alle Raumpunkte, die zu Objekten gehören und im Überwachungsraum liegen, gelten als belegt und sind Element der Menge $R_{\text{filled},k} \subset R$. Damit ist ein jedes Objekt o gegeben durch die Menge an Raumpunkten $R_{\text{o}} \subset R_{\text{filled},k}$, wobei R_{o} zusammenhängend und abgeschlossen ist. Alle Punkte, die nicht zu Objekten, aber zum Überwachungsraum gehören, sind leer, werden als frei bezeichnet und bilden die Menge $R_{\text{free},k}$, wobei gilt: $R := R_{\text{filled},k} \dot{\cup} R_{\text{free},k}$. Jeder belegte Raumpunkt hat eine Objektzugehörigkeit und kann nur zu genau einem Objekt $o \in O_k$ gehören. Umgekehrt enthält jedes Objekt mindestens einen belegten Raumpunkt. Es gelten die Zusammenhänge: $\dot{\bigcup}_{o \in O_{\text{stat}}} R_{\text{o}} := R_{\text{filled,stat}}$ sowie $\dot{\bigcup}_{o \in O_{\text{dyn},k}} R_{\text{o}} := R_{\text{filled,dyn},k}$. Für die statischen Raumpunktmengen wird auf die Zeitvariable k verzichtet, da diese als konstant betrachtet werden können. Die Menge aller belegten Raumpunkte setzt sich zusammen aus: $R_{\text{filled},k} := R_{\text{filled,stat}} \dot{\cup} R_{\text{filled,dyn},k}$. Anmerkung: Damit diese Definitionen gelten, wird zum einen die Annahme getroffen, dass Objekte in ihrem Inneren belegt sind, und zum anderen, dass sich Objekte entweder vollständig im Überwachungsraum befinden oder vollständig außerhalb (aufgrund der

gewählten Abgeschlossenheit von Objekten im Überwachungsraum). Die Aufzählung dieser Fälle würde die Darstellung nur unnötig verkomplizieren, ohne dabei für diese Ausarbeitung einen weiteren Nutzen zu erbringen.

Ein Kamerasystem bestehend aus $C := \{c_1, \ldots, c_{|C|}\}$ einer endlichen Menge von $|C| \in \mathbb{N}$ kalibrierten und synchronisierten Kameras wird zur Überwachung des Raums R eingesetzt (vgl. Abschnitt 3.1). Geht man nur von der Existenz statischer Objekte aus, wie während der Referenzbilderstellung, so entstehen zusammenhängende Volumina verdeckter Raumpunkte (Verdeckungsvolumina), die als $R_{\text{occ},C,\text{stat}}$ bezeichnet werden. Die Verdeckungsvolumina $R_{\text{occ},C,k}$, die während des Personen-Trackings zu den Zeitpunkten k auftreten, werden von der Menge an Objekten O_k, bestehend sowohl aus statischen Objekten als auch aus dynamischen Objekten gebildet, die sich zusammen im Überwachungsraum befinden . Die komplementäre Menge der unverdeckten Punkte $R_{\text{vis},C,k}$ sei definiert als $R_{\text{vis},C,k} := R \setminus R_{\text{occ},C,k}$ (vgl. auch Formel (3.13)).

Es stellt sich die Frage, welches Wissen man aus der Rekonstruktion zur Weiterverarbeitung beim Personen-Tracking nutzen kann. Die Rekonstruktion als Ausgabe des konservativen Algorithmus der VH mit Verdeckungsbehandlung (Algorithmus 5.4.3) lässt sich als Menge an Raumpunkten $R_{\text{recon},k}$ auffassen. Diese Menge kann dabei nach Formel (5.6) als Vereinigung zweier weiterer Mengen verstanden werden, wobei die Menge R_{stat} einer TH von den statischen Objekten entspricht. Die verbleibende Teilmenge der Rekonstruktion, welche den dynamischen Objekten zugeordnet werden kann, wird als $R_{\text{dyn},k}$ bezeichnet. Anzumerken ist hierbei jedoch, dass die dynamischen Objekte nicht zwangsläufig vollständig darin enthalten sein müssen. Dies ist dann der Fall, wenn sich diese mindestens teilweise in einem statischen Verdeckungsvolumen befinden, das initial leer war.

$$R_{\text{recon},k} := R_{\text{stat}} \dot{\cup} R_{\text{dyn},k} \tag{5.6}$$

Die Menge der Raumpunkte R_{stat} von der TH besteht wie in Formel (5.7) angegeben aus allen belegten Raumpunkten $R_{\text{filled,stat}}$, die zu statischen Objekten gehören, sowie allen zugehörigen Verdeckungsvolumina $R_{\text{occ},C,\text{stat}}$. Diese Mengen können bereits a priori für das Kamerasystem unter Zuhilfenahme der 3D-Modelle bestimmt werden und verändern sich im Verlauf des Systembetriebs nicht.

$$R_{\text{stat}} := R_{\text{filled,stat}} \cup R_{\text{occ},C,\text{stat}} \tag{5.7}$$

Für die Zustandsschätzung beim Personen-Tracking lässt sich daraus bereits Wissen ableiten. So kann davon ausgegangen werden, dass die Raumpunkte, die eine zu trackende Person einnimmt, nicht zur Menge der statischen Objekte $R_{\text{filled,stat}}$ gehören können.

Ein weiterer Aspekt ergibt sich aus folgender Betrachtung: Von der Menge R_{stat} können durch die gegebene Menge $R_{\text{filled,stat}}$ die Raumpunkte bestimmt werden, die permanent verdeckt und beim Systemstart leer sind, da sie nicht zu den statischen Objekten gehören. Diese Volumina werden durch Formel (5.8) beschrieben.

$$R_{\text{occEmpty}} := R_{\text{stat}} \setminus R_{\text{filled,stat}} := R_{\text{occ},C,\text{stat}} \setminus (R_{\text{filled,stat}} \cap R_{\text{occ},C,\text{stat}}) \tag{5.8}$$

Im Verlauf der Überwachung können diese verdeckten Raumpunkte teilweise oder vollständig von dynamischen Objekten (abhängig von ihrer Größe) belegt werden, wie oben schon erwähnt. Dies gilt es beim Tracking entsprechend zu berücksichtigen.

Der dynamische Teil der Rekonstruktion $R_{\text{dyn},k}$ ergibt sich aus dem Volumenverschnitt der rückprojizierten segmentierten Silhouetten unter Berücksichtigung der Verdeckungen durch die statischen Objekte. Als Menge $R_{\text{dyn},k}$ werden die Raumpunkte zusammengefasst, die zusätzlich zum statischen Teil der Rekonstruktion R_{stat} entstehen, auch wenn es möglich ist, dass sich Verdeckungsvolumina dynamischer Objekte mit Verdeckungsvolumina statischer Objekte (einzeln betrachtet) überschneiden und sich dynamische Objekte in den Raumpunkten R_{occEmpty} befinden können.

$$R_{\text{dyn},k} := (R_{\text{filled,dyn},k} \setminus (R_{\text{filled,dyn},k} \cap R_{\text{occ},C,\text{stat}})) \cup (R_{\text{occ},C,k} \setminus R_{\text{occ},C,\text{stat}})) \cup R_{\text{artefact},k} \tag{5.9}$$

Die in Abschnitt 5.5 beschriebenen Artefakte sind Raumpunkte $R_{\text{artefact},k}$, die sichtbar und frei sind und damit weder zu Objekten noch zu Verdeckungsvolumina gehören. Es gilt: $R_{\text{artefact},k} \in (R_{\text{free},k} \cap R_{\text{vis},C,k})$. Für die Menge $R_{\text{dyn},k}$, die man durch die Rekonstruktion erhält, stehen keine Informationen zur Verfügung, mit denen sich aufschlüsseln lässt, zu welcher der angegebenen Teilmengen $R_{\text{filled,dyn},k}$, $R_{\text{occ},C,k}$ oder $R_{\text{artefact},k}$ ein Raumpunkt gehört. Aus diesem Grund muss beim Tracking davon ausgegangen werden, dass es sich bei jedem Punkt der Menge $R_{\text{dyn},k}$ um einen belegten Raumpunkt handeln kann, der zu einem dynamischen Objekt gehört. Alle Raumpunkte der Menge $R_{\text{dyn},k}$ werden deshalb beim Tracking gleich behandelt.

Zusammenfassend lässt sich ein jeder Raumpunkt nach der Rekonstruktion genau einer der beschriebenen Mengen zuordnen, die zusammen den Überwachungsraum bilden (Formel 5.10): zur Menge statisch belegter Objekte oder zur Menge initial leerer, statischer Verdeckungen oder zum definierten dynamischen Teil der Rekonstruktion oder zur verbleibenden Menge freier sichtbarer Raumpunkte des Überwachungsraums, die als $\hat{R}_{\text{free},k} \in R_{\text{free},k}$ bezeichnet wird.

$$R := R_{\text{filled,stat}} \,\dot\cup\, R_{\text{occEmpty}} \,\dot\cup\, R_{\text{dyn},k} \,\dot\cup\, \hat{R}_{\text{free},k} \tag{5.10}$$

Durch die Zuordnung der Raumpunkte zu einer dieser Mengen für jeden Zeitpunkt k kann im Beobachtungsmodell des Trackers eine unterschiedliche Gewichtung der Raumpunkte vorgenommen werden. Allerdings wird tatsächlich nicht die Zugehörigkeit einzelner Raumpunkte bestimmt, sondern die Zugehörigkeit der Voxel des Voxelraums. Im folgenden Abschnitt wird dies näher erläutert.

5.7 Wissen in Form von Voxelzuständen

Zur Repräsentation der Rekonstruktion werden Voxeldaten verwendet. Die Beschreibung der Diskretisierung des Überwachungsraums durch eine Voxeldatenstruktur erfolgte in Abschnitt 5.4.1. Mit $V := \{v_0, \ldots, v_{|V|-1}\}$ sei die endliche Menge aller zugehörigen $|V| \in \mathbb{N}$ Voxel v_i gegeben. Der Algorithmus 5.4.3 zur Rekonstruktion einer voxelbasierten konservativen VH mit Verdeckungsbehandlung wird für das Tracking nun so modifiziert, dass eine Voxelklassifikation erfolgt, die einer Zuordnung der Voxel zu den in Formel (5.10) dargestellten Teilmengen des Überwachungsraums entspricht.

Die Voxelklassifikation ist verlustbehaftet, da mit der Verwendung rückprojizierter Silhouettenpixel und Voxeldaten Diskretisierungsfehler einhergehen. Aufgrund der räumlichen Voxelausdehnung kann deshalb auch nicht sichergestellt werden, dass jedes Voxel nur Raumpunkte aus genau einer der angegebenen Teilmengen enthält. Vielmehr bestehen für jedes Voxel $2^4 - 1$ Möglichkeiten wie die Teilmengen von Formel (5.10) zusammengesetzt sein können (Das Fehlen aller Teilmengen ist keine valide Kombination.). Es muss somit explizit entschieden werden, wie ein Voxel abhängig von seiner Raumpunktzusammensetzung klassifiziert werden soll. Die für diese Arbeit gewählte Lösung wird im Folgenden vorgestellt und ist in Formel 5.11 zusammengefasst. Dafür sei die Funktion voxelValue : $V \rightarrow \{\textit{filled}, \textit{empty}, \textit{unknown}, \textit{occluded}\}$ definiert, die jedem Voxel v_i einen Voxelzustand zuweist.

Die bisherige Konservativität bei der Entscheidung, ob ein Voxel leer ist oder zur Rekonstruktion gehört, wird implizit beibehalten. Ein Voxel soll demnach als nicht leer gelten, wenn sich auch nur ein Objektteil darin befindet, unabhängig davon, ob es sich um ein statisches oder dynamisches Objekt handelt. Ein Voxel gilt nach Formel (5.11) als **filled** (belegt), wenn es mindestens einen Raumpunkt der Teilmenge $R_{\text{filled,stat}}$ enthält. Ist dies nicht der Fall und zum Voxel gehört mindestens ein Raumpunkt aus der Teilmenge $R_{\text{dyn},k}$, so wird das Voxel als **unknown** (unbekannt) klassifiziert. Der Begriff *unknown* wurde gewählt, weil nicht bekannt ist, ob das Voxel tatsächlich belegt ist. Aus Sicherheitsgründen muss aber davon ausgegangen werden, dass solch ein Voxel Teil eines dynamischen Objekts ist. Nun kann es Voxel geben, die Raumpunkte aus beiden genannten Mengen enthalten. Diese würden bei beschriebener Vorgehensweise immer den

Algorithmus 5.3 Bestimmung von Voxelzuständen während der Rekonstruktion einer konservativen Visuellen Hülle mit Verdeckungsbehandlung

```
1: procedure VoxelSatesOfConservativeVisualHull(V, V_stat, C, I_bin, I_dep)
2:     for all v_i ∈ V do                                          ▷ for each voxel
3:         setVoxelValue(v_i) ← occluded                           ▷ voxel is not visible for any
4:                                                                 pixel in any camera
5:         for all c_j ∈ C do                                      ▷ for each camera
6:             if v_i ∈ V_stat then                                ▷ voxel is filled by a static object
7:                 setVoxelValue(v_i) ← filled                     ▷ set voxel to value filled
8:                 break
9:             end if
10:            flag_foregroundPixelExist ← false
11:            flag_occludedPixelExist ← false
12:            Φ_{v_i,c_j} ← projectVoxelToCam(v_i, c_j)
13:            for all p_l ∈ Φ_{v_i,c_j} do                        ▷ for each projection pixel of the voxel
14:                if |r_{v_i} − e_{c_j}| + d_v < getImageValue(p_l, I_dep, c_j) then   ▷ if voxel
15:                                                                is visible
16:                    if getImageValue(p_l, I_bin, c_j) = 1 then  ▷ if pixel is foreground
17:                        flag_foregroundPixelExist ← true
18:                    end if
19:                else                                            ▷ else pixel is background
20:                    flag_occludedPixelExist ← true              ▷ voxel is not visible in the pixel
21:                end if
22:            end for
23:            if (flag_foregroundPixelExist = false) ∧
24:               (flag_occludedPixelExist = false) then
25:                setVoxelValue(v_i) ← empty                      ▷ voxel does not belong to
26:                                                                the reconstruction
27:                break
28:            end if
29:            if (flag_foregroundPixelExist = true) then          ▷ at least one foreground
30:                                                                pixel exists
31:                setVoxelValue(v_i) ← unknown                    ▷ assume a dynamic
32:                                                                object in the voxel
33:            end if
34:        end for
35:    end for
36:    return V_VH
37: end procedure
```

Zustand *filled* zugewiesen bekommen. Intuitiv würde man annehmen, dass eine invertierte Klassifikation der Voxelzustände besser sei, sodass ein Voxel immer *unknown* wäre, wenn sich mindestens ein Raumpunkt aus $R_{\text{dyn},k}$ darin befindet. Die statischen Objekte würde man damit nachrangig behandeln. Dies hätte jedoch den Nachteil einer fluktuierenden Klassifikation bei einer konservativen Abschätzung der Raumbelegung (ohne weitere Voxelzustände einzuführen). Konkret könnten dadurch Voxel, die teilweise innerhalb eines statischen Objekts liegen, wechselnd als *unknown* oder als *filled* klassifiziert werden, abhängig vom dynamischen Teil der Rekonstruktion. Dies wird bei der gewählten Vorgehensweise vermieden: *filled*-Voxel verändern ihren Zustand über die Zeit nicht. Im Falle der Durchführung eines Kollisionstests könnte die Vereinigung aus *filled*- und *unknown*-Voxeln eine sichere Approximation der Raumbelegung repräsentieren.

Ein Voxel erhält den Zustand **empty** (leer oder frei), wenn sämtliche Raumpunkte des Voxels der Teilmenge $\hat{R}_{\text{free},k}$ zugehörig sind – der Menge aller sichtbaren freien Raumpunkte, die nicht Teil der Rekonstruktion sind. Hier kann davon ausgegangen werden, dass sich keine Person darin befindet.

Als letzte Möglichkeit kann ein Voxel den Zustand **occluded** zugewiesen bekommen, was der Fall ist, wenn ein Voxel zu den Verdeckungsvolumina statischer Objekte gehört, aber nicht zu den statischen Objekten selbst. Das Voxel besteht dann aus Punkten der Teilmenge R_{occEmpty}. Solche Voxel sind aufgrund der Abwesenheit dynamischer Objekte initial leer. Zur Laufzeit können sich aber dynamische Objekte darin befinden.

$$\text{VoxelValue}\,(v_i,k) := \begin{cases} \textit{filled} & \text{if} \quad \exists r \in v_i : r \in R_{\text{filled,stat}} \\ \textit{unknown} & \text{if} \quad \exists r \in v_i : r \in R_{\text{dyn},k} \ \wedge \ \nexists r \in v_i : r \in R_{\text{filled,stat}} \\ \textit{empty} & \text{if} \quad \forall r \in v_i : r \in \hat{R}_{\text{free},k} \\ \textit{occluded} & \text{otherwise} \end{cases} \tag{5.11}$$

Eine Rekonstruktion, die solch eine Zuordnung von Voxelzuständen zu den Voxeln nach Formel (5.11) vornimmt und den Überwachungsraum entsprechend unterteilt, wird von Algorithmus 5.3 präsentiert. Dieser ist nicht laufzeitoptimiert. Der Zeitpunkt k wird im Algorithmus außen vor gelassen. Die Verwendung der Voxelzustände zur Gewichtung innerhalb des Beobachtungsmodells des Partikelfilters wird in Abschnitt 6.4 beschrieben.

Basierend auf den vorangegangenen Definitionen dieses Kapitels (vgl. Abschnitt 5.4.2) seien für Algorithmus 5.3 gegeben: eine Kameramenge C mit den Einzelkameras c_j und den jeweiligen Pixeln p_l aus einer Pixelmenge P_{c_j}. Die Menge der Projektionspixel eines Voxels v_i in der Kamera c_j sei mit $\Phi_{v_i,c_j} \subset P_{c_j}$ definiert. Auf die Farbbilder I_{col} aller Kameras wird ein Background Subtraction angewendet. Das Ergebnis ist eine Anzahl segmentierter binärer Silhouettenbilder, gegeben als I_{bin}.

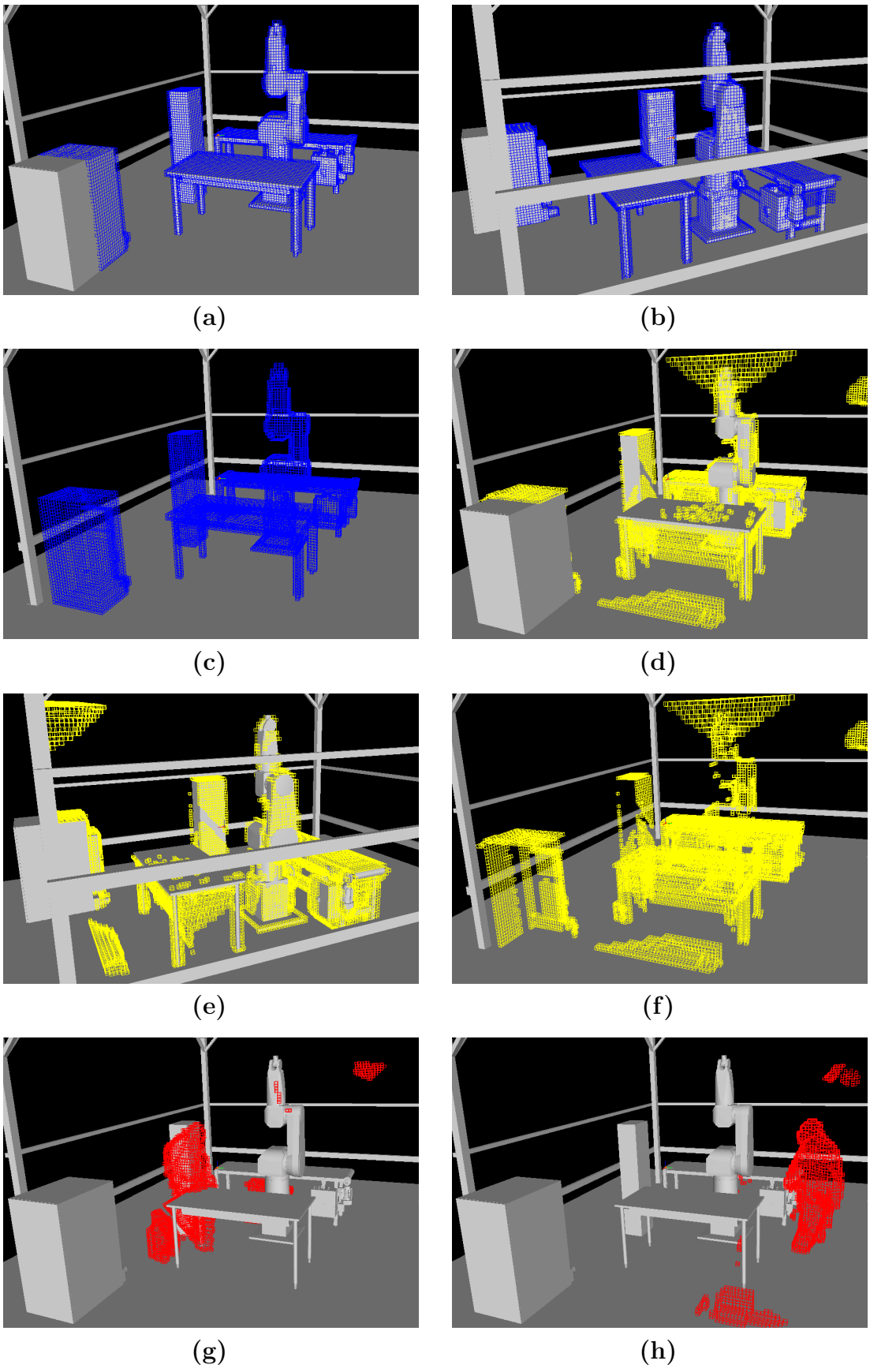


Abb. 5.15: Visualisierung verschiedener Voxelzustände in der Roboterarbeitszelle.

In einem Silhouettenbild ist der Wert eines jeden Pixels $p_l \in P_{c_j}$ entweder gegeben als $\text{image}_{\text{bin},c_j}(p_l) = 0$, wenn p_l als Hintergrund klassifiziert wurde oder als $\text{image}_{\text{bin},c_j}(p_l) = 1$ bei einer Klassifizierung von p_l als Vordergrund.

Gegeben sei ferner eine endliche Menge von statischen a priori bekannten Objekten O_{stat} im Überwachungsraum, die geometrisch modelliert sind. Die Modelle werden in einem Offline-Schritt verarbeitet, der als **Voxelisierung** bezeichnet wird. Dabei werden alle Voxel der Menge $V_{\text{stat}} \subset V$ bestimmt, die vollständig oder teilweise innerhalb der Modelle liegen und den Zustand *filled* erhalten. Das Ergebnis der Voxelisierung ist in den Teilabbildungen 5.15(a), (b) und (c) blau eingefärbt. In (a) und (b) sind zusätzlich die Dreiecksnetze der Modelle grau eingezeichnet. Wände, Boden und Aluminiumgestell liegen außerhalb des Voxelsraums, wie aus Abb. 5.7 hervorgeht. Sie werden daher nicht voxelisiert und auch nicht in die Berechnungen einbezogen.

Weiterhin werden die synthetischen Tiefenbilder I_{dep} mithilfe der Modelle für alle Kameras in einem Offline-Schritt generiert. Diese kodieren den Abstand jedes Pixels p_l bis zum nächstgelegenen statischen Objekt. Die Tiefenbilder werden verwendet, um zu bestimmen, ob die Voxel für die Kameras sichtbar sind oder nicht. Hierfür sei der Zentralprojektionspunkt $e_{c_j} \in \mathbb{R}^3$ einer jeden Kamera c_j gegeben sowie das Zentrum $r_{v_i} \in \mathbb{R}^3$ jedes Voxels v_i und eine Konstante $d_v \in \mathbb{R}$, welche die Diagonalenhälfte eines Voxels angibt (um konservativ zu sein). Die Bedingung in Zeile 13 in Algorithmus 5.3 ergibt daher Voxel, die aus Kamerasicht vollständig vor den statischen Objekten liegen. Voxel, für die das nicht zutrifft, werden an dieser Stelle ignoriert, da sie bereits durch die Konstruktion von V_{stat} in den Zeilen 5–8 berücksichtigt werden. In den Abbildungen 5.15(d), (e) und (f) sind die Voxel des Zustands *occluded* gelb eingefärbt. Voxel des Zustands *unknown* sind in den Abb. 5.15(g) und (h) rot eingefärbt. Sie werden von mindestens einer Kamera vollständig gesehen und gleichzeitig in den entsprechenden Kameras als Vordergrund detektiert werden. Sie repräsentieren die unverdeckten Teile dynamischer Objekte.

5.8 Zusammenfassung

In diesem Kapitel wurde der Fokus auf eine konservative 3D-Rekonstruktion dynamischer Objekte gelegt. Dazu wurden nach einem Überblick über verschiedene Rekonstruktionsverfahren das ausgewählte Konzept der Visuellen Hülle detailliert vorgestellt und Algorithmen dafür präsentiert. Für die Rekonstruktion werden zur Laufzeit Silhouettenbilder der dynamischen Objekte via Background Subtraction erzeugt. Die Präsenz statischer Objekte im Überwachungsraum kann zu Störungen der Silhouettenbilder führen, wenn die dynamischen Objekte dadurch verdeckt werden. Daraus resultie-

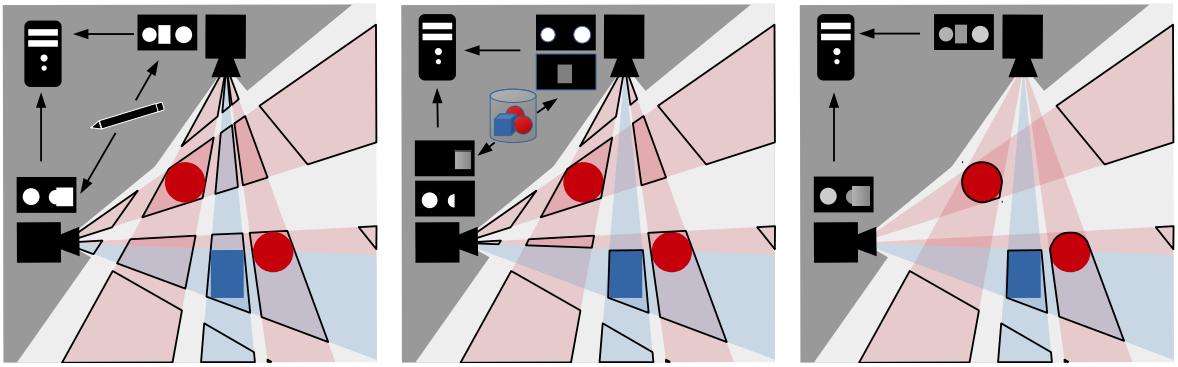


Abb. 5.16: Gegenüberstellung verschiedener Verfahren zur Rekonstruktion dynamischer Objekte (rot) und statischer Objekte (blau). Visuelle Hülle mit Occlusions Masks von den statischen Objekten (a), Visuelle Hülle mit synthetischen Tiefenbildern von den statischen Objekten (b), Rekonstruktion einer Tiefenhülle (c).

rende unvollständige Silhouetten führen bei der Rekonstruktion einer VH durch das prinzipbedingte Verschneiden rückprojizierter Sichtkegel zu unvollständigen Volumina.

Eine Möglichkeit, dieses Problem zu vermeiden, besteht in der manuellen Segmentierung der statischen Objekte in den Bildern [Ladikos et al., 2008] und dem Hinzufügen dieser zu den automatisch detektierten Silhouettenbildern. Im Ergebnis enthält die Visuelle Hülle damit auch die statischen Objekte. Eine alternative Vorgehensweise aus [Kuhn und Henrich, 2009] beinhaltet die Generierung synthetischer Tiefenbilder zu 3D-Modellen der statischen Objekte, die bei der Rekonstruktion ausgewertet werden. Damit entfallen die Artefakte vor den statischen Objekten, die andernfalls Teil der Rekonstruktion wären. Veranschaulicht kann ein damit erzeugtes Ergebnis als Volumenvereinigung der Rekonstruktionen einer TH der statischen Objekte und einer VH der dynamischen Objekte betrachtet werden. In Abbildung 5.16 werden die Varianten in einer 2D-Aufsicht visualisiert, gegeben zwei dynamische Objekte (rot) und ein statisches Objekt (blau) im Überwachungsraum. Die Rekonstruktion besteht in jedem Teilbild aus sämtlichen schwarz umrandeten Bereichen. Das Ergebnis der Verwendung von Occlusion Masks (a) ist der angewandten Methode dieser Dissertation (b) gegenübergestellt. Erkennbar ist der Wegfall der Artefakte bei dem statischen Objekt (b). Weniger leere Volumina sind damit in der gesamten Rekonstruktion enthalten. In Abb. (c) ist zum weiteren Vergleich eine rekonstruierte TH dargestellt, die entstehen könnte, wenn auch von den dynamischen Objekten Tiefendaten zur Verfügung stehen würden. Dies ist jedoch in dieser Dissertation nicht der Fall (vgl. Abschnitt 2.1).

Im Anschluss an die formale Beschreibung des Rekonstruktionsalgorithmus erfolgte eine Analyse der Rekonstruktionsbestandteile, wobei auf Verdeckungsvolumina, Artefakte sowie Pseudoobjekte eingegangen wurde. Die Bestandteile werden als Mengen betrach-

tet. Für die Kernaufgabe des Trackings wurden als Ergebnis vier Mengen abgeleitet, die sich gegenseitig ausschließen und zusammengenommen den Überwachungsraum ergeben. Diese repräsentieren unterschiedliches Wissen zur Abschätzung der Raumbelegung. Unter Beachtung der Diskretisierung durch die Voxeldatenstruktur wird eine Klassifikation aller Voxel entsprechend ihrer Zusammensetzung aus den vier Mengen in sogenannte Voxelzustände vorgenommen. Durch eine Modifikation des zuvor beschriebenen Rekonstruktionsalgorithmus werden die Voxelzustände für die Weiterverarbeitung beim Tracking direkt ausgegeben (vgl. Abschnitt 5.7). Diese werden in der Likelihood-Gewichtung des Trackings im nachfolgenden Kapitel eine Rolle spielen.

Kapitel 6

3D-Personen-Tracking

Untersuchungsgegenstand dieses Kapitels ist die Zustandsschätzung eines Tracking-Verfahrens für die 3D-Personenlokalisierung in Präsenz statischer Objekte und ihren Verdeckungsvolumina. Als Eingabe werden voxelbasierte Rekonstruktionsdaten verarbeitet, die aus Bildern eines Multi-View-Kamerasystems generiert werden. Im vorangegangenen Kapitel wurde die 3D-Rekonstruktion beschrieben und detailliert auf die dabei erzeugten Voxelzustände eingegangen. Diese werden im Beobachtungsmodell des Tracking-Verfahrens (in der Likelihood-Funktion) ausgewertet.

Zu Beginn dieses Kapitels wird in Abschnitt 6.1 der untersuchte Tracking-Ansatz dieser Dissertation erläutert. Für das Tracking wird ein Partikelfilter eingesetzt, zu welchem die relevanten theoretischen Aspekte sowie der konkret verwendete Algorithmus in Abschnitt 6.2 dargestellt werden. In Abschnitt 6.3 werden verschiedene grundlegende Tracking-Details für die gegebene Anwendung präsentiert, bestehend aus dem Zustandsraum und dem Bewegungsmodell, der Umsetzung des Mehrpersonen-Trackings sowie der Vorgehensweise bei der Track-Initialisierung und -Terminierung. Die Beschreibungen zur Likelihood-Funktion und zum Kollisionstest, welcher bei der Partikelprädiktion eingesetzt wird, stellen die wichtigsten Inhalte dieses Kapitels dar und lassen sich den Abschnitten 6.4 und 6.5 entnehmen. Das Kapitel wird in Abschnitt 6.6 zusammengefasst. Die experimentellen Untersuchungen zu dem vorgeschlagenen Tracking-Verfahren werden im sich anschließenden Kapitel 7 dargestellt.

6.1 Tracking-Ansatz

Im Stand der Forschung (vgl. Kapitel 4) wurden rekonstruktionsbasierte Ansätze des Personen-Trackings und der Pose-Estimation vorgestellt und bewertet, die sowohl komplizierte Körpermodelle einsetzen – insbesondere Skelettmodelle – als auch einfache Modelle mit geringerer Parameteranzahl. Für die komplizierten Modelle konnten mit den rekonstruktionsbasierten Verfahren Erfolge beim Tracking von Einzelpersonen in weitestgehend leeren Überwachungsräumen erzielt werden. Dabei treten keine nennenswerten Verdeckungen auf und in den Kameras kann von einer guten Sichtbarkeit des Körpers ausgegangen werden. Objektselbstverdeckungen, die das Erfassen einzelner Körperteile

erschweren können, werden mitunter durch zusätzliche Bewegungseinschränkungen, z. B. durch die Annahme eines aufrechten Gangs, vermieden.

Anders sieht es bei Szenarien des Mehrpersonen-Trackings aus. Befinden sich mehrere Personen im Raum, so treten neben Selbstverdeckungen typischerweise auch gegenseitige Objektverdeckungen auf. Begeben sich die Personen in Verdeckungsvolumina statischer Objekte, so kommt es zu weiteren Verdeckungen. Dadurch ist mit einer variierenden Sichtbarkeit der Personen und ihrer Körperteile zu rechnen, was auch mit einer inkonsistenten Qualität der Rekonstruktionsdaten einhergeht. Diese Bedingungen erschweren eine permanent gute Einpassung komplizierter Körpermodelle, was aus Sicht der Autorin ein Grund für den Einsatz einfacher Objektmodelle bei den betrachteten Tracking-Verfahren sein könnte.

Da in dieser Dissertation größere bis vollständige Objektverdeckungen (der Personen) durch gegebene Verdeckungsvolumina adressiert werden sollen, wird hierfür ebenfalls ein einfaches Objektmodell verwendet, konkret ein Ellipsoidmodell. Bei den recherchierten Verfahren zum Mehrpersonen-Tracking sind mitunter statische Objekte im Raum gegeben, in den Experimenten ist jedoch nicht erkennbar, dass sich die Personen auch in größere Verdeckungsvolumina begeben und damit überhaupt größeren Objektverdeckungen ausgesetzt sind. Solche Situationen sollen Schwerpunkt der Betrachtungen des aktuellen und des nachfolgenden Kapitels sein.

Für das Tracking wird ein Partikelfilter eingesetzt, weil dieser multimodale Aposteriori-Wahrscheinlichkeitsdichten approximieren und nichtlineare Systeme abbilden kann. Durch die einzelnen Partikel ist er in der Lage, mehrere Zustandshypothesen, die sich in den Ausprägungen der Modalitäten zeigen, gleichzeitig zu verfolgen. Für Anwendungsszenarien mit statischen Objekten, eingeschränkter Sichtbarkeit, variierender Rekonstruktionsgüte sowie entstehenden Artefakten erscheint dies sinnvoll. Gegeben seien die Eingabedaten aus der Aufgabenstellung von Abschnitt 1.5. Damit wird nach der besten Schätzung des unbekannten Zustands $\mathbf{x}_k$ einer oder mehrerer Personen zu jedem Zeitpunkt k gesucht. Für jede Person wird dabei eine neue Instanz des Partikelfilters erzeugt (vgl. Abschnitt 6.3.2). Der implementierte Partikelfilter-Algorithmus ist auch unter dem Namen „Bootstrap-Filter“ bekannt und kann [Choset et al., 2006] entnommen werden. Dieser wendet das Sequential Sampling Importance Resampling (SIR) [Gordon et al., 1993] zur Vermeidung des sogenannten Degenerierungsproblems an und besitzt eine Komplexität von $O(N)$ (vgl. Abschnitt 6.2.4).

Die beste Schätzung zu jedem Zeitpunkt k wird durch den gewichteten Schwerpunkt aller Partikel (Erwartungswert) im Zustandsraum gebildet. Sie dient lediglich dem Vergleich verschiedener Parametrisierungen und ist keine rekursive Eingabe in den Algorithmus. Daher beeinflusst die Wahl der Schätzmethode nicht den Filterverlauf. Die

alternative Verwendung des Partikels mit maximalem Gewicht erwies sich als zu fluktuierend und wurde daher als ungeeignet eingestuft, um die Effekte der unterschiedlichen Parametrisierungen in den Experimenten aufzuzeigen.

Das umgesetzte Tracking-Verfahren weist eine starke Nähe zu [Canton-Ferrer et al., 2011] auf. Im Unterschied dazu wird weiterführend untersucht, ob die Integration von Wissen durch gegebene 3D-Modelle der statischen Objekte und ihren Verdeckungsvolumina ein erfolgreiches Tracking bei größeren und auch vollständigen Objektverdeckungen der Personen ermöglicht.

Im vorangegangenen Kapitel wurde der Rekonstruktionsalgorithmus von [Kuhn und Henrich, 2009] modifiziert, wodurch jedes Voxel des Überwachungsraums mit einem von vier exklusiven Voxelzuständen versehen wird. Diese Voxelmerkmale kodieren unterschiedliches Wissen zur Raumbelegung und den Verdeckungsvolumina. Die Voxel erhalten innerhalb der Likelihood-Berechnung des Partikelfilters entsprechend ihrer Zustände unterschiedliche Gewichte (vgl. Abschnitt 6.4). Zur Ermöglichung des Verfolgens von Personen durch Verdeckungsvolumina hindurch werden die *occluded*-Voxel positiv gewichtet. Tatsächlich kann jedoch keine Messung für sie vorliegen, weil sie vollständig verdeckt sind. Dementsprechend wird ein Verdeckungsvolumen als eine Art Pseudomessung behandelt.

Als weiterer Untersuchungsschwerpunkt werden die statischen Objekte in die Partikelpropagierung einbezogen. Objekte können nicht nur zu Objektverdeckungen führen, sie stellen auch physische Barrieren dar, die Einfluss auf das Bewegungsverhalten der Personen ausüben. Um geschätzte Zustände zu verhindern, die real nicht eingenommen werden können, kann die Filterbewegung im Zustandsraum durch Nebenbedingungen eingeschränkt werden. Die Durchführung eines Kollisionstests zwischen allen jeweils propagierten Partikeln und den statischen Objekten soll die Generierung valider Partikel herbeiführen, bei denen die Ellipsoide nicht (weit) in statische Objekte „hineinragen“ und eine Filterbewegung durch diese hindurch verhindert wird. Details hierzu werden in Abschnitt 6.5 beschrieben.

6.2 Partikelfilter

Inhalt dieses Abschnitts ist die zugrunde liegende Theorie des gewählten Partikelfilters. Der Partikelfilter stellt eine diskrete Implementierung des Bayes-Filters dar, zu welchem in Abschnitt 6.2.1 ein Überblick gegeben wird. In Abschnitt 6.2.2 erfolgt ein Abriss der historischen Entwicklung des Partikelfilters sowie eine Beschreibung seiner Eigenschaften. Der Partikelfilter beruht auf dem Prinzip des Sequential Importance Sampling, das in

Abschnitt 6.2.3 vorgestellt wird. Die Erweiterung dieses Prinzips um eine Resampling-Methode führt zu dem Sequential Sampling Importance Resampling, beschrieben in Abschnitt 6.2.4. Der konkret verwendete Partikelfilteralgorithmus dieser Dissertation, ein Bootstrap-Filter, wird in Abschnitt 6.2.5 erläutert.

6.2.1 Bayes-Filter

Das Ziel des Trackings von Objekten besteht darin, aufeinanderfolgende Objektzustände, wie beispielsweise Raumpositionen, aus Sensormessungen (genannt Beobachtungen) zu schätzen. Zur Lösung solch eines Schätzproblems wurde basierend auf dem Bayes-Theorem der rekursive **Bayes-Filter** entwickelt, welcher den Zustand eines dynamischen Systems probabilistisch aus verrauschten Beobachtungen schätzt [Arulampalam et al., 2002]. Bayes-Filter repräsentieren den Zustand eines Objekts zum Zeitpunkt k durch eine Zustandsvariable x_k. Die Unsicherheit des Zustands wird zu jedem Zeitpunkt von einer Wahrscheinlichkeitsdichtefunktion $p(x_k|Z_{1:k})$ über x_k repräsentiert (kurz: Dichte). Diese wird beispielsweise in der Robotik auch als **Aposteriori**-Dichte oder als **Belief** bezeichnet. Die Historie an Messungen ist gegeben mit $Z_{1:k} := (z_1, z_2, .., z_k)$. Nach jeder neuen Beobachtung z_k wird die Aposteriori-Dichte über den gesamten Zustandsraum X neu bestimmt.

Um ein exponentielles Wachstum der Berechnungskomplexität zur Bestimmung der Aposteriori-Dichte mit zunehmenden Sensormessungen über die Zeit zu verhindern, wird angenommen, dass das System einem Markov-Prozess unterliegt. Unter dieser Prämisse hängt die Zustandsschätzung zum Zeitpunkt k nur von dem vorherigen Zustand x_{k-1} ab. Die Zustände vor dem Zeitpunkt $k-1$ stellen keine weiteren Informationen für die Schätzung des Zustands x_k zur Verfügung. Ferner gilt die Annahme, dass jede Sensormessung unabhängig von allen vorangegangenen Messungen ist, sprich sie jeweils nur von dem aktuellen Zustand x_k abhängt.

Die Aposteriori-Dichte $p(x_k|Z_{1:k})$ sei definiert als eine Funktion $p : X \to \mathbb{R}$ und gilt für kontinuierliche Zustandsräume X (vgl. Formel (6.1)). Die Zustandsvariable x und die Beobachtungsvariable z sind auf einem kontinuierlichen Wertebereich definiert. Beobachtungen beginnen erst zum Zeitpunkt 1 und x_0 entspricht dem Initialzustand.

$$p(x_k|Z_{1:k}) = \eta_k p(z_k|x_k) \int_X p(x_k|x_{k-1}) p(x_{k-1}|Z_{1:k-1}) \, \mathrm{d}x_{k-1}, \quad k \geq 1 \tag{6.1}$$

Zur Durchführung einer rekursiven Schätzung wird die Dichte $p(x_{k-1}|Z_{1:k-1})$ des vorangegangenen Zeitpunkts $k-1$ benötigt, die auch als **Apriori**-Dichte bezeichnet wird. Existiert keine vorangegangene Schätzung, was typischerweise vor der ersten Beobach-

tung der Fall ist, so wird hierfür eine Initialdichte $p(x_0)$ eingesetzt. Diese kann eine anwendungsspezifische uni- oder multimodale Dichte sein oder, falls kein Vorwissen zur Verfügung steht, beispielsweise auch eine konstante Dichte, die über den gesamten Zustandsraum angenommen wird.

Eine Zustandsschätzung besteht aus zwei grundlegenden Schritten: der **Prädiktion** (engl. Prediction) und der **Korrektur** (engl. Update). Bei gegebener Apriori-Dichte zum Zeitpunkt $k-1$ wird für den Zeitpunkt k der Zustand des getrackten Objekts x_k vorhergesagt. Hierfür wird ein **Bewegungsmodell** $p(x_k|x_{k-1})$ des Systems benötigt, welches kodiert, wie sich der Systemzustand über die Zeit verändert. Das Bewegungsmodell gibt die bedingte Wahrscheinlichkeit dafür an, dass sich das System im Zustand x_k befindet, wenn es sich vorher im Zustand x_{k-1} befand. Das verwendete Bewegungsmodell (auch als Prozessmodell bezeichnet) hängt stark von den Informationen ab, die beim Schätzprozess zur Verfügung stehen. Es kann beispielsweise einfach aus einer linearen Extrapolation des letzten Bewegungsabschnitts im Zustandsraum bestehen oder auch Kontextwissen über die Bewegungsziele der Person im Raum beinhalten.

Nach der Prädiktion wird die prädizierte Zustandsschätzung durch das **Beobachtungsmodell** korrigiert, sobald eine Beobachtung z_k zur Verfügung steht. Das Beobachtungsmodell (auch als Messmodell bezeichnet) in Form der **Likelihood** $p(z_k|x_k)$ gibt an, wie wahrscheinlich die Messung z_k auftritt, wenn sich das Objekt im Zustand x_k befindet, beispielsweise in einer bestimmten Raumposition. Wie die Likelihood-Funktion konkret aussieht, hängt von den verwendeten Sensortypen und deren Fehlercharakteristik sowie vom Objektmodell (Parameter des Zustandsraums) ab.

Die Normalisierungskonstante η_k in Formel (6.1) garantiert die Aufsummierung der Wahrscheinlichkeiten über den gesamten Zustandsraum zu 1, wodurch die Bedingung $\int p(x_k|Z_{1:k})\mathrm{d}x_k = 1$ erfüllt wird. Anwendungsabhängig (z. B. bei mobilen Plattformen) können bei dem Bayes-Filter durch eine weitere Zufallsvariable auch Steueraktionen berücksichtigt werden, welche den Verlauf der Zustandsänderung beeinflussen. Dadurch können beispielsweise Fahrkommandos eines mobilen Roboters in die Schätzung integriert werden. Für die Betrachtungen der vorliegenden Dissertation ist dies jedoch nicht relevant.

Die Bayes-Filterung (auch genannt optimales oder stochastisches Filterungsproblem) ist zunächst nur ein abstraktes Konzept, das ein probabilistisches Framework für eine rekursive Zustandsschätzung zur Verfügung stellt. Erst durch die Konkretisierung der abstrakten Komponenten erhält man implementierbare Algorithmen. Aufgrund der hohen Dimensionalität der Beobachtungsvektoren bei Problemstellungen aus der Praxis können diese Dichten meist auch nicht empirisch erfasst werden.

Die Realisierungen des Bayes-Filters unterscheiden sich hauptsächlich durch verschiede-

ne Repräsentationen der Aposteriori-Dichte sowie anwendungsspezifische Bewegungs- und Beobachtungsmodelle. Eine analytische Lösung für beliebige Wahrscheinlichkeitsverteilungen ist nicht möglich. Deshalb wird sich in den Algorithmen entweder auf exakt lösbare Spezialfälle beschränkt, oder es werden mehr oder weniger genaue Approximationen der Dichten verwendet. Zwei bekannte Vertreter sind der **Kalman-Filter** und der **Partikelfilter**.

Der Kalman-Filter setzt voraus, dass der Zustand zu jedem beliebigen Zeitpunkt normalverteilt ist und als n-dimensionale Gaußverteilung repräsentiert werden kann. Die Zustandsänderung ist linear und a priori bekannt. Das Bewegungsmodell und das Beobachtungsmodell werden somit als lineare Funktionen mit normalverteiltem, mittelwertfreiem und unkorreliertem Rauschen angenommen. Die Herleitung der Kalman-Gleichungen aus der allgemeinen Bayes-Schätzung orientiert sich an [Thrun et al., 2005]. Beim Kalman-Filter wird zu jedem Zeitpunkt immer nur eine Hypothese zur Zustandsschätzung verfolgt. Er eignet sich für relativ genaue Sensoren mit einer hohen Aktualisierungsrate.

Für die Aufgabe des Personen-Trackings wurde der nachfolgend beschriebene Partikelfilter allerdings als geeigneter bewertet, da für das gegebene Anwendungsszenario erwartet wurde, dass die Annahmen des Kalman-Filters nicht ausreichend erfüllt werden können.

6.2.2 Eigenschaften des Partikelfilters

Partikelfilter sind eine nichtparametrische Form des Bayes-Filters und stammen von den **Sequentiellen Monte-Carlo-Methoden** (SMC) ab [Metropolis und Ulam, 1949]. Einen Überblick über generelle SMC-Methoden gibt [Doucet und Johansen, 2011]. In [MacCormick, 2012] wird ein kurzer Abriss zur Entwicklung des Partikelfilters gegeben, auch wenn es schwierig ist, den akkuraten geschichtlichen Ablauf zu rekonstruieren. Partikelfilter werden in einer Bandbreite von Anwendungen eingesetzt und die grundlegenden Algorithmen wurden unabhängig voneinander von Forschern verschiedener Disziplinen, wie z. B. der Physik, Statistik und Signalverarbeitung, entwickelt. Quellen, welche detaillierte Informationen zu Partikelfiltern und deren theoretischer Herleitung geben, sind beispielsweise [Gordon et al., 1993], [Doucet, 1998] und [Doucet et al., 2001]. Der Einsatz von Partikelfiltern für Tracking-Aufgaben wird unter anderem in [Arulampalam et al., 2002] und [Ristic et al., 2004] beschrieben.

Beim Partikelfilter, der eine diskrete Implementierung des Bayes-Filters darstellt, wird mit Hilfe einer Menge von Stichproben und assoziierten Gewichten versucht, die Aposteriori-Dichte zu approximieren und eine Schätzung des aktuellen Zustands zu berechnen. Die Stichprobe, genannt **Partikel** oder **Sample**, repräsentiert einen Punkt

im Zustandsraum. Die zu schätzenden Aposteriori-Dichten können multimodal und beliebig sein, was die Verfolgung mehrerer Hypothesen erlaubt. Jede Modalität der Aposteriori-Dichte kann dabei als Hypothese interpretiert werden. Genau genommen stellt aber auch jedes einzelne Partikel eine eigene Hypothese dar. Der Partikelfilter ermöglicht die Verwendung nichtlinearer Bewegungs- und Beobachtungsmodelle.

Die parallele Weiterverfolgung mehrerer Hypothesen kann für verschiedene Gegebenheiten sehr nützlich sein. So kann der Tracking-Erfolg bei Messungen mit niedriger Aktualisierungsrate erhöht werden. Gleiches gilt für ungenaue Messdaten, die größeren Störungen unterliegen und damit im Beobachtungsmodell Mehrdeutigkeiten hervorrufen. In [Isard und Blake, 1998] wird gezeigt, wie mit einer problemunabhängigen Implementierung des Partikelfilters, dem „Condensation-Algorithmus“, Objekte trotz starker Störmerkmale im Hintergrund weiter verfolgt werden können.

Einerseits hat der Partikelfilter eine lineare Laufzeit in der Anzahl an Partikeln. Andererseits sollte für eine vorgegebene Ortsauflösung/Genauigkeit die Partikelanzahl exponentiell mit der Dimension des Zustandsraums wachsen. Die Anzahl benötigter Partikel ist prozessabhängig und richtet sich nach der Komplexität der zu approximierenden Dichte sowie des Bewegungs- und Beobachtungsmodells. Vorteilhaft ist dabei, dass die Partikelanzahl variierend und auch adaptiv gestaltet werden kann, z. B. in Abhängigkeit der zur Verfügung stehenden Rechenressourcen. Ein weiterer Vorteil ist, dass Partikelfiltertechniken meist recht einfach implementiert werden können, was auch ein Grund für deren Popularität ist.

Die genannten Eigenschaften des Partikelfilters werden zur Lösung der betrachteten Aufgabenstellung als geeignet eingestuft, weshalb dieser zum Einsatz kommt. Die Auswahl beruht auf der prinzipiellen Anwendbarkeit des Algorithmus sowie auf seinem Status als häufig verwendeter Ansatz in anderen Arbeiten. Es gibt eine Vielzahl an Erweiterungen, jedoch liegt das Hauptaugenmerk in dieser Dissertation nicht auf einem Vergleich verschiedener Varianten des Partikelfilters, sondern auf der Integration von Wissen zu Verdeckungsvolumina in ein Tracking-Verfahren.

6.2.3 Sequential Importance Sampling

Der Algorithmus des **Sequential Importance Samplings** (SIS) ist eine Monte-Carlo-Methode, welche die Basis für die meisten SMC-Filter bildet. Das SIS ist unterschiedlich bekannt als „Bootstrap Filter“ [Gordon et al., 1993], „Survival of the Fittest“ [Kanazawa et al., 1995], „Condensation-Algorithmus“ [MacCormick und Blake, 1999], „Partikelfilter“ [Carpenter et al., 1999] oder auch „Interacting Particle Approximation“ [Del Moral und Miclo, 2000].

Das SIS ist eine Technik für die Implementierung eines rekursiven Bayes-Filters mit Monte-Carlo-Simulationen, bei dem die Aposteriori-Dichte sequentiell zum Zeitpunkt k aktualisiert wird, ohne die zuvor berechneten Aposteriori-Dichten modifizieren zu müssen. [Handschin und Mayne, 1969] gehören zu den Forschern, welche ursprünglich eine Methode von sequentiellem Bayes-Filtering basierend auf zufälligem Sampling umsetzten.

Der SIS-Algorithmus besteht aus einer rekursiven Propagierung von Gewichten und Stützpunkten (Partikeln) basierend auf den sequentiellen Messungen. Der SIS-Algorithmus stellt eine spezielle Umformung des **Importance Sampling** (IS) dar [Ristic et al., 2004], die im Folgenden näher beschrieben wird. Die Herleitung wurde dabei weitestgehend aus [Arulampalam et al., 2002] entnommen.

Sei $\{X_{0:k}^{(i)}, w_k^{(i)}\}_{i=1}^N$ ein zufälliges Maß, welches eine unbekannte bedingte Wahrscheinlichkeitsdichte, die Aposteriori-Dichte $p(X_{0:k}|Z_{1:k})$, charakterisiert. Dabei ist die Menge an Stützpunkten $\{X_{0:k}^{(i)}|i = 0, ..., N\}$ mit Gewichten $\{w_k^{(i)}|i = 1, .., N\}$ assoziiert. Ferner ist $X_{0:k} = \{x_j|j = 0, ..., k\}$ die Menge aller Zustände bis zum Zeitpunkt k. Die Gewichte sind normalisiert: $\sum_i w_k^{(i)} = 1$. Damit ergibt sich in Formel (6.2) für k eine diskrete gewichtete Approximation der wahren Aposteriori-Dichte $p(X_{0:k}|Z_{1:k})$.

$$p(X_{0:k}|Z_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(X_{0:k} - X_{0:k}^{(i)}) \tag{6.2}$$

Die Gewichte $w_k^{(i)}$ werden unter Verwendung des Prinzips des IS bestimmt, welches auf folgenden Überlegungen beruht [Doucet, 1998]: Von der Wahrscheinlichkeitsdichte $p(x)$ ist es schwierig, Samples zu ziehen, da sie typischerweise nicht zur Verfügung steht. Jedoch können Samples $x^{(i)} \sim q(x)$ mit $i = 1, .., N$ von einer frei wählbaren Ersatzdichte q generiert werden, die als **Importance-Dichte** bezeichnet wird und proportional zur Aposteriori-Diche ist, aber einfacher bestimmt werden kann. Damit ist eine gewichtete Approximation zu der Dichte p gegeben durch Formel (6.3). Für die Ersatzdichte q gilt, dass $q \equiv p$ optimal wäre.

$$p(x) \approx \sum_{i=1}^{N} w^{(i)} \delta(x - x^{(i)}) \tag{6.3}$$

Wenn die Samples $X_{0:k}^{(i)}$ mit Hilfe der Importance-Dichte $q(X_{0:k}|Z_{1:k})$ gezogen werden, dann lassen sich die Gewichte von Formel (6.2) angeben wie in Formel (6.4).

$$w_k^{(i)} \propto \frac{p(X_{0:k}^{(i)}|Z_{1:k})}{q(X_{0:k}^{(i)}|Z_{1:k})} \tag{6.4}$$

Bei Betrachtung des sequentiellen Falls kann man mit jeder Iteration Samples erhalten, welche die Dichte $p(X_{0:k-1}|Z_{1:k-1})$ repräsentieren und eine Approximation der Dichte $p(X_{0:k}|Z_{1:k})$ mit einer neuen Menge an Samples vornehmen. Wird die Importance-Dichte so gewählt, dass sie wie in Formel (6.5) zerlegt werden kann,

$$q(X_{0:k}|Z_{1:k}) = q(x_k|X_{0:k-1}, Z_{1:k})q(X_{0:k-1}|Z_{1:k-1}) \tag{6.5}$$

dann kann man N neue Partikel entsprechend Formel (6.6) auf Basis der Importance-Dichte erhalten.

$$X_{0:k}^{(i)} \sim q(X_{0:k}|Z_{1:k}) \tag{6.6}$$

Hierfür werden die existierenden Partikel in Formel (6.7)

$$X_{0:k-1}^{(i)} \sim q(X_{0:k-1}|Z_{1:k-1}) \tag{6.7}$$

mit dem neuen Zustand erweitert, wie in Formel (6.8) angegeben.

$$x_k^{(i)} \sim q(x_k|X_{0:k-1}, Z_{1:k}) \tag{6.8}$$

Um die Aktualisierung der Gewichte herzuleiten (vgl. [Arulampalam et al., 2002] für Details), wird die Aposteriori-Dichte $p(X_{0:k}|Z_{1:k})$ zunächst durch $p(X_{0:k-1}|Z_{1:k-1})$, die Likelihood-Funktion $p(z_k|x_k)$ sowie das Prozessmodell $p(x_k|x_{k-1})$ ausgedrückt, wie in Formel (6.9) definiert. Dies spiegelt die sogenannte sequentielle Bayes-Schätzung wider.

$$p(X_{0:k}|Z_{1:k}) \propto p(z_k|x_k)p(x_k|x_{k-1})p(X_{0:k-1}|Z_{1:k-1}) \tag{6.9}$$

Das Einsetzen der Terme aus den Formeln (6.5) und (6.9) in Formel (6.4) ergibt die Formel (6.10).

$$w_k^{(i)} \propto \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})p(X_{0:k-1}^{(i)}|Z_{1:k-1})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, Z_{1:k})q(X_{0:k-1}^{(i)}|Z_{1:k-1})} \tag{6.10}$$

Dies ist äquivalent zu Formel (6.11), welche die rekursive Gewichtsberechnung der Importance-Gewichte besser zum Ausdruck bringt:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|X_{0:k-1}^{(i)}, Z_{1:k})} \tag{6.11}$$

Geht man nun von dem häufigen Fall aus, bei dem nur eine Schätzung der gefilterten Aposteriori-Dichte $p(x_k|Z_{1:k})$ zu jedem Zeitpunkt k benötigt wird, dann wird hierfür die gefilterte Importance-Dichte herangezogen, welche nur abhängig von x_{k-1} und z_k

ist, so wie in Formel (6.12) angegeben.

$$q(x_k|X_{0:k-1}, Z_{1:k}) = q(x_k|x_{k-1}, z_k) \tag{6.12}$$

In diesem Fall muss für jedes Partikel nur $x_k^{(i)}$ gespeichert werden. Deshalb kann man den Pfad $X_{0:k-1}^{(i)}$ sowie die Historie der Beobachtungen $Z_{1:k-1}$ außen vor lassen. Die Gewichtsberechnung ändert sich dann zu Formel (6.13).

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)} \tag{6.13}$$

Die gefilterte Aposteriori-Dichte kann dementsprechend mit den Importance-Gewichten von Formel (6.13) approximiert werden, wie in Formel (6.14) angegeben.

$$p(x_k|Z_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{6.14}$$

Es kann gezeigt werden, dass die Approximation aus Formel (6.14) für $N \to \infty$ gegen die wahre Aposteriori-Dichte $p(x_k|Z_{1:k})$ konvergiert.

6.2.4 Sampling Importance Resampling

Bei den SIS-Algorithmen besteht das Problem der **Partikeldegenerierung**. Die Varianz der Partikelzustände wird über die Zeit mit den Iterationen immer größer, was zum Ergebnis hat, dass nur noch wenige Partikel Gewichte aufweisen, die signifikant größer als 0 sind. Bei dieser **Partikelineffizienz** sind Partikel mit Gewichten von nahezu 0 verschwendet, da sie nicht die Aposteriori-Dichte in den Regionen mit hoher Wahrscheinlichkeit approximieren. Befinden sich nur wenige Partikel in solchen Regionen, so wird die Approximation aufgrund der reduzierten Anzahl an Abtastungen unpräziser.

In [Doucet, 1998] wurde formal gezeigt, dass eine Degenierung von Partikeln bei SIS-Algorithmen praktisch nicht verhindert werden kann. Nur unter bestimmten Voraussetzungen wie einer unendlich hohen Partikelanzahl und der Verwendung spezieller Importance-Funktionen ist dies möglich. In der Theorie werden Importance-Funktionen diskutiert, welche die Varianz der Importance-Gewichte minimieren. Von [Zaritskii et al., 1975] wurde dahingehend die sogenannte optimale Importance-Funktion eingeführt, aber auch diese hat praktische Nachteile. Deshalb werden die SIS-Algorithmen um eine **Resampling**-Methode erweitert.

Nach [MacCormick, 2012] wurde ein Algorithmus von Rubin [Rubin, 1987], welcher ein Resampling enthält, unter dem Namen **Sampling Importance Resampling**

(SIR) populär. In [Gordon et al., 1993] und [Kitagawa, 1996] wurde das SIR unabhängig voneinander angewendet. Gordons Algorithmus ist bekannt unter dem Namen **Bootstrap-Filter**. Kitagawa, welcher den generischen Term **Monte-Carlo-Filter** verwendet, war anscheinend der erste Nicht-Physiker, der die zufälligen Samples als Partikel bezeichnete [Kitagawa, 1996].

Die Idee des Resamplings ist die Eliminierung von Partikeln mit schwachen normalisierten Importance-Gewichten zur Positionierung der Partikel in Bereichen hoher Zustandswahrscheinlichkeit. Bei einer Resampling-Prozedur werden Partikel mit einer Wahrscheinlichkeit proportional zu ihrem Gewicht gezogen und vervielfältigt. Dadurch werden Partikel geringen Gewichts mit hoher Wahrscheinlichkeit entfernt und Partikel mit hohem Gewicht vervielfältigt, sodass die Partikelanzahl N gleich bleibt. Es entsteht eine neue Partikelmenge, wobei jedes Partikel ein durchschnittliches Gewicht von $1/N$ erhält. In Abb. 6.1 sind die allgemeinen Schritte eines SIR-Algorithmus dargestellt.

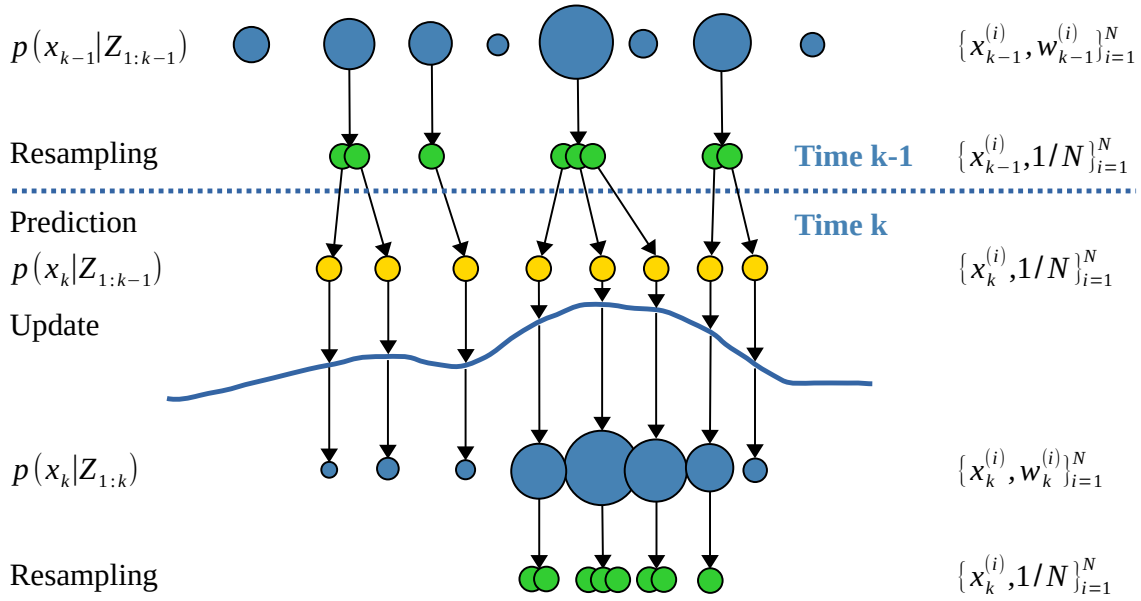


Abb. 6.1: Schritte eines SIR-Algorithmus (von oben nach unten). Die Partikel werden als Kreise symbolisiert. Gegeben ist eine Partikelmenge mit Importance-Gewichten zum Zeitpunkt $k-1$ (blau, oberste Zeile). Die Partikeldurchmesser sind proportional zu den Importance-Gewichten dargestellt. Im nachfolgenden Resampling werden Partikel geringer Importance mit hoher Wahrscheinlichkeit entfernt und Partikel mit hoher Importance vervielfältigt. Es entsteht eine neue Menge von Partikeln gleichen Gewichts $1/N$ (grün). Für $k=1$ würde diese Partikelmenge von einer initialen Dichte gezogen werden. Zum Zeitpunkt k erfolgt eine Prädiktion der Partikel (Prediction). Die Partikel werden dabei durch Ziehung von einem probabilistischen Bewegungsmodell in einen neuen Zustand überführt (gelb). Anschließend erhalten die Partikel ein neues Gewicht (Update) durch die Likelihood-Funktion, wenn die Messung zum Zeitpunkt k zur Verfügung steht. Im Ergebnis besitzen die Partikel wieder unterschiedliche Gewichte proportional zu ihrer Importance (blau) und es kommt zu einer zyklischen Wiederholung der bereits beschriebenen Schritte.

Obwohl das Degenerierungsproblem durch das Resampling verringert werden kann, entstehen neue Probleme, die nicht beim ursprünglichen SIS-Algorithmus existieren. Die Partikel interagieren, wodurch die simulierten Trajektorien nicht mehr länger statistisch unabhängig sind. Mithin gibt es einen Verlust in der Partikeldiversität, da Trajektorien mit hoher Importance statistisch mehrfach ausgewählt werden. Um den Diversitätsverlust zu reduzieren, wurden viele Ad-hoc-Prozeduren vorgestellt. So werden beispielsweise bei [Gordon et al., 1993] die Trajektorien künstlich gestört. Weiterhin begrenzt das Resampling die Parallelisierbarkeit des Algorithmus. Jedoch muss festgehalten werden, dass das Resampling, trotz genannter Nachteile, sowohl große praktische Effekte als auch theoretische Vorteile mit sich bringt [Doucet und Johansen, 2011] und dementsprechend häufig eingesetzt wird.

Für den Zeitpunkt der Durchführung des Resamplings gibt es verschiedene Strategien. Eine Möglichkeit besteht darin, nur zu Resampeln, wenn eine Degenerierung detektiert wird. Als Maß dafür kann die **Effective Sample Size** $\hat{N}_{\text{eff}}$ verwendet werden, welche von den normalisierten Gewichten geschätzt wird [Arulampalam et al., 2002], [Kong et al., 1994], [Doucet, 1998] und [Doucet et al., 2001]. Immer dann, wenn $\hat{N}_{\text{eff}}$ unter eine feste Schwelle fällt, wird eine Resampling-Prozedur gestartet. In vielen Algorithmen, wie auch dem Bootstrap-Filter, wird allerdings in jeder Iteration resampelt. Vier häufig verwendete Resampling-Schemata werden in [Hol et al., 2006] dargestellt.

Bei dem populären Condensation-Algorithmus [Isard und Blake, 1998] wird das Resampling durch Erzeugung einer Zufallszahl und Bisektion auf den kumulativen Gewichten der Partikel erzeugt. Dieser Resampling-Algorithmus hat eine Komplexität von $O(N \cdot \log N)$. Zudem kann er zu einer Reduktion der Partikeldiversität führen, da hierbei auch Partikel mit größerem Gewicht aus dem Resampling fallen können. Eine reduzierte Diversität kann zu einem Hypothesenverlust führen, da die Partikel beim Resampling zufällig verschwinden und nicht wieder regeneriert werden können, wenn ein Track einmal alle Partikel verloren hat.

In dem wie folgt aufgeführten Bootstrap-Filter wird das Resampling auf eine einfachere Art bewerkstelligt, welche eine reduzierte Komplexität von $O(N)$ besitzt und garantiert, dass jedes Partikel mit einem Gewicht, welches größer ist als der Durchschnitt $1/N$, mindestens einmal gewählt wird.

6.2.5 Bootstrap-Filter

Der in [Gordon et al., 1993] vorgestellte Bootstrap-Filter ist eine sehr häufig verwendete Umsetzung des SIR. Die dazu in dieser Dissertation vorgenommene Implementierung entspricht den Algorithmen 6.1 und 6.2. Aus [Choset et al., 2006] wurden die Pseudocodes

und deren Beschreibung weitestgehend entnommen. Die einzelnen Schritte lassen sich auch in Abb. 6.1 wiederfinden.

Die Aposteriori-Dichte wird von einer Menge M von N Samples repräsentiert. Jedes Sample i zu einem Zeitpunkt j besteht aus einem Paar $(x_j^{(i)}, w_j^{(i)})$, das einen Zustandsvektor $x_j^{(i)}$ des zugrundeliegenden Systems beinhaltet sowie einen Gewichtsfaktor $w_j^{(i)}$. Letzterer wird verwendet, um die „Importance“ des entsprechenden Partikels zu speichern.

Als Eingabe für den Algorithmus wird die bestehende Sequenz von Messungen $Z_{1:k}$ und eine Menge M von N Samples $(x_0^{(i)}, w_0^{(i)})$ entsprechend der initialen Apriori-Dichte $p(x_0)$ verwendet. Auf die gegebene Partikelmenge wird im Schritt der Prädiktion das Bewegungsmodell angewandt (Zeile 4 in Algorithmus 6.1), welches aus einem Drift und einer Diffusion (Rauschkomponente) besteht. Diese Überführung der Partikel durch das Bewegungsmodell $p(x_j|x_{j-1}^{(i)})$ in einen neuen Zustand entspricht dem Sampling von einer Importance-Dichte. Anschließend wird im Update-Schritt mit der neuen Messung z_j rekursiv das Gewicht $w_j^{(i)}$ eines Partikels $(x_j^{(i)}, w_j^{(i)})$ als die Likelihood $p(z_j|x_j^{(i)})$ dieser Beobachtung berechnet, gegeben das System in Zustand $x_j^{(i)}$ (Zeile 9 in Algorithmus 6.1). Damit gilt: $w_j^{(i)} \propto w_{j-1}^{(i)} p(z_j|x_j^{(i)})$, da $w_{j-1}^{(i)} = 1/N$ nach dem Resampling und der Prädiktion.

Algorithmus 6.1 Sequential Importance Sampling mit Resampling (SIR)

1: **procedure** SIR($Z_{1:k}$, N samples $(x_0^{(i)}, w_0^{(i)})$ from $p(x_0)$)
2: **for** $j \leftarrow 1$ to k **do** ▷ for each time step
3: **for** $i \leftarrow 0$ to $N-1$ **do** ▷ for each sample of M
4: compute a new state x by sampling according to $p(x|x^{(i)})$ ▷ drift and
5: diffuse particle
6: $x^{(i)} \leftarrow x$
7: **end for**
8: $\eta \leftarrow 0$
9: **for** $i \leftarrow 0$ to $N-1$ **do** ▷ for each sample of M
10: $w^{(i)} = p(z_j|x^{(i)})$ ▷ get importance weight from likelihood function
11: $\eta = \eta + w^{(i)}$ ▷ add importance weight to accumulative weight
12: **end for**
13: **for** $i \leftarrow 0$ to $N-1$ **do** ▷ for each sample of M
14: $w^{(i)} = \eta^{-1} \cdot w^{(i)}$ ▷ normalize importance weight
15: **end for**
16: M =RESAMPLE(M) ▷ generate a new sample set
17: **end for**
18: **return** $p(x_k|Z_{1:k})$ ▷ return estimation of aposteriori pdf at time k
19: **end procedure**

Algorithmus 6.2 Resampling

```
1: procedure RESAMPLE(M)
2:     M' ← ∅                                          ▷ new sample set is empty
3:     △ ← RAND((0; 1/N])                              ▷ select a random value from interval
4:     c ← w_0                                         ▷ add first importance weight to accumulative weight
5:     i ← 0                                           ▷ assign index of first sample
6:     for r ← 0 to N − 1 do                           ▷ iterate in fix steps over raster
7:         u ← △ + r · 1/N                             ▷ set offset △ to next step
8:         while u > c do                              ▷ value of next step is bigger than accumulative weight
9:             i ← i + 1                               ▷ go to next sample weight
10:            c ← c + w_i                             ▷ add new weight to accumulative weight
11:        end while
12:        M' ← M' ∪ {(x_i, 1/N)}                      ▷ add sample to new sample set
13:    end for
14:    return M'                                       ▷ return new sample set
15: end procedure
```

Die gewichtete Partikelmenge beschreibt jeweils die aktualisierte Schätzung. Nach Berechnung der Importance-Gewichte wird in Zeile 16 von Algorithmus 6.1 die sogenannte Resampling-Prozedur aufgerufen (Algorithmus 6.2).

Beim Resampling werden N Samples aus der Eingabemenge M gezogen, sodass jedes Sample mit einer Wahrscheinlichkeit proportional zu seinem Gewicht $w^{(i)}$ ausgewählt wird. Entsprechend überleben Samples mit größerem Gewicht mit höherer Wahrscheinlichkeit als Samples mit geringerem Gewicht. Dazu wird in der Prozedur ein festes Raster im Rastermaß $1/N$ über die Aneinanderreihung der Gewichte gelegt, die dafür nicht sortiert sein müssen (Zeile 6 in Algorithmus 6.2). Das Raster wird dabei mit Hilfe eines zufälligen Startwerts $\triangle$ aus dem Intervall $(0; N^{-1}]$ verschoben (Zeile 7 in Algorithmus 6.2). Wann immer ein Rasterpunkt in ein Gewichtsintervall eines Samples fällt, wird das zugehörige Sample zur Menge der neuen Samples M' hinzugefügt (Zeile 12 in Algorithmus 6.2). Liegen mehrere Rasterpunkte in einem Gewichtsintervall, so wird ein Sample vervielfältigt. Das Gewicht jedes neuen Samples wird auf $1/N$ gesetzt, um die Vervielfältigung auszugleichen. Als Ergebnis des Algorithmus 6.2 wird eine neue Menge M' an N Samples zurückgeliefert.

Am Ende erfolgt die Ausgabe der Aposteriori-Dichte $p(x_k|Z_{1:k})$ über den Zustand der Person zum Zeitpunkt k, repräsentiert von der aktualisierten Menge M (Zeile 18 in Algorithmus 6.1).

6.3 Tracking-Komponenten

Nachfolgend werden einzelne Komponenten beschrieben, die Teil des Tracking-Verfahrens sind und spezifische Eigenschaften des Anwendungsszenarios berücksichtigen. In Abschnitt 6.3.1 werden der Zustandsraum, in dem getrackt wird, sowie das eingesetzte Bewegungsmodell erläutert. Die Vorgehensweise für das gleichzeitige Verfolgen mehrerer Personen wird in Abschnitt 6.3.2 vorgestellt, wobei auch auf den Umgang mit dem Datenassoziationsproblem eingegangen wird. Die Track-Initialisierung und -Terminierung wird in Abschnitt 6.3.3 beschrieben.

6.3.1 Zustandsraum und Bewegungsmodell

Zur Personenverfolgung wird ein Ellipsoidmodell unbekannten Zustands $\mathbf{x}_k$ verwendet, dessen Parameter zu jedem Zeitpunkt k geschätzt werden. Das Ellipsoidmodell (vgl. Abb. 6.2) sei definiert über die Variablen x_k, y_k, z_k, die das Ellipsoidzentrum angeben, sowie l_k, m_k, n_k, welche für die Längen der Ellipsoidachsen stehen, die an den Achsen des Voxelraums ausgerichtet sind. Die drei Achsen jedes Ellipsoids können verschieden sein, weshalb es sich um triaxiale Ellipsoide handelt.

Die Parameter des Ellipsoidmodells bilden neben drei weiteren Parametern, die noch beschrieben werden, die Dimensionen des Zustandsraums, in welchem mit dem Partikelfilter getrackt wird. Jedes Partikel repräsentiert dabei einen Punkt im Zustandsraum. Nach dem Schritt des Resamplings (vgl. Abschnitt 6.2) werden die Partikel der neuen Partikelmenge von dem Bewegungsmodell in einen neuen Zustand überführt, d. h. neu parametrisiert, auch Prädiktion oder Propagierung genannt.

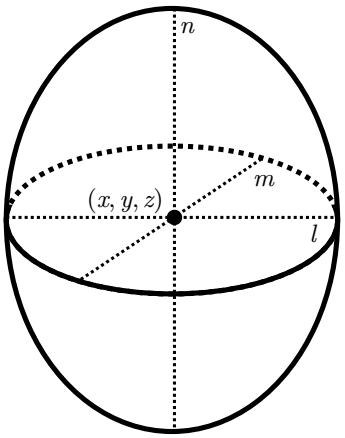


Abb. 6.2: Als Objektmodell kommt ein triaxiales Ellipsoid zum Einsatz, dessen Achsen l, m, n unterschiedliche Längen aufweisen können. Das Ellipsoidzentrum ist gegeben durch x, y, z.

Die Verwendung eines möglichst exakten Bewegungsmodells ist oft der Schlüssel für gute Tracking-Ergebnisse. Allerdings ist das Bewegungsmodell auch an die Repräsentation des Objektmodells gebunden und damit nicht beliebig wählbar. Bei Skelettmodellen beispielsweise ist die Form und Größe der Körperteile meistens fix. Die Bewegung wird über die Anordnung der Körperteile zueinander durch die Artikulierung der Gelenkwinkel beschrieben (Posen). Damit können auch kompliziertere Bewegungen ausgedrückt werden. Es ist naheliegend, dass die Parameter des Ellipsoidmodells zur Kodierung des Zentrums und der Ausdehnung keine Abbildung detaillierter Bewegungen zulässt. Zudem wird von den Achsenlängen sowohl die Form als auch die Bewegung der Person gleichermaßen kodiert. Die Orientierung bzw. Rotation des Ellipsoids wird nicht betrachtet, da sie den Zustandsraum um drei Dimensionen vergrößern würde und angenommen wird, dass die zusätzlichen Rotationsparameter keinen großen Gewinn an zusätzlicher Approximationsgüte erbringen würden. Da die zu erwartende Vielfalt und Komplexität an Personenbewegungen im Anwendungsszenario sehr hoch ist und nur wenigen Einschränkungen unterliegen soll, besteht das Ziel der Bewegungsvorhersage in dieser Dissertation nicht darin, die Bewegungsabläufe exakt einzufangen. Vielmehr soll die Personenverfolgung auch während größerer Verdeckungen aufrechterhalten und eine 3D-Lokalisierung der Person(en) ermöglicht werden. Aus diesem Grund werden, wie auch in vielen anderen Arbeiten der Literatur, einige Variablen des Zustandsraums als konstant angenommen und auftretende Änderungen durch ein überlagertes mittelwertfreies normalverteiltes Rauschen $\mathcal{N}$ modelliert.

Implementierungsdetails zu dem in dieser Dissertation eingesetzten Bewegungsmodell sind in Abschnitt 7.1 beschrieben. Nach dem Resampling (vgl. Algorithmus 6.1) wird der neue Zustand $\mathbf{x}_k^{(j)}$ jedes Partikels j mit Anwendung der Formeln (6.15) und (6.16) propagiert. Neben den Variablen des Ellipsoidmodells bilden die ersten Ableitungen der Positionsvektoren des Ellipsoidzentrums $\dot{x}_k, \dot{y}_k, \dot{z}_k$ drei weitere Dimensionen des Zustandsraums. Für das Bewegungsmodell haben verschiedene Tests eine im Mittel gute Bewegungsverfolgung gezeigt, wenn die Geschwindigkeitsvektoren mit in den Zustandsraum aufgenommen werden. Damit ergibt sich der Zustand $\mathbf{x}_k = \left(x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k, l_k, m_k, n_k\right) \in \mathbb{R}^9$. Als Randbedingung für das Tracking wird ein konstantes Ellipsoidvolumen angenommen. Weiterhin werden aber auch die Längen der Ellipsoidachsen begrenzt, um Maße, die nicht mehr dem Menschen entsprechen können, zu vermeiden. Ein Beispiel hierfür wäre ein 3 m hohes und gleichzeitig sehr schmales Ellipsoid. Die Positionsvariablen werden durch die Grenzen des Überwachungsraums (den definierten Voxelraum) eingeschränkt.

$$\mathbf{x}_k := F \cdot \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{6.15}$$

$$\mathbf{x}_k := \begin{pmatrix} 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta T & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \\ \dot{z}_{k-1} \\ l_{k-1} \\ m_{k-1} \\ n_{k-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathcal{N}_{\sigma_{\dot{x}}^2} \\ \mathcal{N}_{\sigma_{\dot{y}}^2} \\ \mathcal{N}_{\sigma_{\dot{z}}^2} \\ \mathcal{N}_{\sigma_l^2} \\ \mathcal{N}_{\sigma_m^2} \\ \mathcal{N}_{\sigma_n^2} \end{pmatrix} \tag{6.16}$$

Die Verwendung eines Bewegungsmodells mit konstanter Geschwindigkeit, wie in den Zeilen 4–6 der Formel (6.16), wird in der Literatur als **Constant Velocity Model** bezeichnet und für das Tracking in Zustandsräumen mit geringer Dimensionalität, z. B. bei punktförmigen Objekten eingesetzt. Die Modellierung von Geschwindigkeitsänderungen wie Abbrems- und Beschleunigungsvorgängen oder von Richtungswechseln kann dabei durch ein additives normalverteiltes Rauschen $\mathcal{N}_{\sigma_{\dot{x}}^2}, \mathcal{N}_{\sigma_{\dot{y}}^2}, \mathcal{N}_{\sigma_{\dot{z}}^2}$ erreicht werden.

Dabei wird die Position nicht verrauscht (vgl. Zeilen 1–3 in Formel (6.16)), um eine Überlagerung bei der Variation von Position und Geschwindigkeit zu vermeiden. Es gilt: $\mathcal{N}_{\sigma_x^2} = \mathcal{N}_{\sigma_y^2} = \mathcal{N}_{\sigma_z^2} = 0$. Beim Resampling des Partikelfilters (im Vergleich zur Anwendung beim Kalman-Filter) führt dies zur Entstehung von Partikelkopien, welche die gleiche Position enthalten, sich aber in ihren Geschwindigkeiten und Achsenlängen unterscheiden. Der Effekt redundanter Ellipsoidpositionen wird allerdings als vernachlässigbar eingestuft.

Für die Achsenlängen l_k, m_k, n_k des Zustandsraums wurde neben dem Hinzufügen additiven Rauschens, wie in den Zeilen 7–9 der Formel (6.16) eingetragen, auch die Verwendung der jeweils ersten Ableitung der Achsenlängen (Achsenlängengeschwindigkeiten) im Zustandsraum ausgetestet. Bei größeren Werten für die Achsenlängengeschwindigkeiten trat als negativer Effekt ein zu schnelles Abdriften in unwahrscheinlichere Bereiche des Zustandsraums auf (Ellipsoide, die quer zur Messung liegen). Da für typische Bewegungsabfolgen, wie beispielsweise Laufen, die grobe Form des Menschen über weite Teile konstant ist und im Wesentlichen nur in eng begrenzten Abschnitten bei einem Wechsel der Bewegungsart eine größere Änderung der Form erfolgt, überwiegt der Nachteil der schlechteren Abtastung des größeren Zustandsraums gegenüber dem möglichen Vorteil an wenigen Stellen. Daher wurden die Achsenlängengeschwindigkeiten nicht in den Zustandsraum aufgenommen.

6.3.2 Mehrpersonen-Tracking und Datenassoziationsproblem

Häufig besteht der Wunsch, mehrere Objekte bzw. Personen gleichzeitig zu tracken, was in der Literatur auch als **Multi Target Tracking** bezeichnet wird. Dies erfordert jedoch eine gemeinsame Lösung für das Datenassoziations- und Zustandsschätzproblem. Bei der Verwendung mehrerer Filter muss eine Zuordnung der wahrscheinlichsten Messung für ein bestimmtes Objekt und einen Zustand bestimmt werden [Yilmaz et al., 2006]. Dieses Korrespondenzproblem sollte im Grunde vor Anwendung der Filter gelöst werden und stellt eine besondere Herausforderung dar, wenn Objekte zu nah beieinander liegen, sich ihre Wege kreuzen oder sie sich gegenseitig verdecken. Messungen mit geringem Abstand zueinander, die auch miteinander verschmelzen oder sich wieder aufteilen können (Merge- und Split-Szenarien) sind oft nicht eindeutig den verursachenden Objekten zuordenbar.

In der Literatur finden sich verschiedene Datenassoziationstechniken [Oh et al., 2009]. In heuristischen Methoden wird die Zuordnung von Messungen zu Objekten durch einfache Metriken vorgenommen. Ein typischer Vertreter davon ist das „Global-Nearest-Neighbour-Verfahren“ [Konstantinova et al., 2003]. Dabei wird zu jedem Zeitpunkt genau eine Kombination von Zuordnungen (Hypothese) verfolgt. Mehrere Hypothesen werden bei der „Probabilistic Data Association“ (PDA) und ihrer ebenso bekannten Erweiterung in Form der Methode des „Joint Probabilistic Data Association Filtering“ (JPDAF) erzeugt [Bar-Shalom et al., 1988]. Zu den Maximum-A-Posteriori-Methoden gehört das **Multi Hypothesis Tracking (MHT)** [Reid, 1979], welches aus vorhandenen Hypothesen rekursiv weiterführende Hypothesen über die Zeit entwickelt. Diese erhalten eine Gewichtung entsprechend ihrer Auftrittswahrscheinlichkeit. Nur die k besten Hypothesen werden anschließend weiterverfolgt. Das MHT bietet gegenüber dem JPDAF den Vorteil einer automatisierten Initialisierung und Terminierung von Trajektorienabschnitten, genannt **Tracklets**, unabhängig von der Anzahl der beobachteten Objekte.

Für ein robustes Tracking besteht häufig der Bedarf des Einbeziehens von mehr Informationen, damit die Trajektorienverläufe erfolgreich die tatsächlichen Objektbewegungen abbilden. Verfahren, die mehrere Hypothesen parallel verfolgen, können bei der Erweiterung um zusätzliche Kombinationsmöglichkeiten schnell an ihre Grenzen gelangen: beispielsweise bei der Integration von Fällen, in denen eine Messung mehrere Objekte nach einem Merge repräsentieren kann oder bei denen Messungen unvollständig sind oder gänzlich fehlen. Datenassoziationstechniken mit kombinatorischen Vorgehensweisen führen dann schnell zu einer Erhöhung der Berechnungsdauer und des Speicherbedarfs, was zu einer Einschränkung der Echtzeitfähigkeit und Praxistauglichkeit führen kann. Als Alternative bieten sich Verfahren des **Multi Level Trackings** an. Sie lassen eine Einbeziehung von Umwelt- und komplexeren Bewegungsmodellen sowie Lernkomponen-

ten zu und versuchen, den Aufwand in Grenzen zu halten. Auch im Hinblick auf den Umgang mit Verdeckungen können diese Verfahren interessant sein.

Multi-Level-Verfahren bestehen auf unterster Stufe, der „Low-Level-Ebene“, aus einer eher einfach gehaltenen Tracking-Komponente zur effizienten Erzeugung zusammenhängender Tracklets. Die Datenassoziation kann beispielsweise rein distanzbasiert erfolgen [Huang et al., 2008, Henriques et al., 2011] oder den Einsatz von Kalmanfiltern beinhalten [Perera et al., 2006]. Zur Begrenzung des Berechnungsaufwands bietet sich die Verwendung niedrigdimensionaler Merkmale wie Position und Geschwindigkeit einer punktförmigen Messung an. In einer oder mehreren nachgeschalteten Entscheidungskomponente(n) der „High-Level-Ebene“ werden zusammenhängende Trajektorien aus den Tracklets gewonnen, was als Stitching oder **Linking** bezeichnet wird. Hierbei können höherdimensionale Merkmale, wie z. B. Farbhistogramme, ausgewertet werden, um das Zuordnungsproblem auf höherer Ebene zu lösen. In den meisten Fällen wird dafür eine Maximum-A-Posteriori-Schätzung vorgenommen, aber auch andere Techniken wie Bayes'sche Netze [Jorge et al., 2004] oder MHT-Frameworks [Makris und Prieur, 2014, Singh et al., 2008] finden Anwendung. Weitere Quellen sowie eine detailliertere Gegenüberstellung verschiedener Multi-Level-Verfahren dieser Verfahrensklasse lassen sich [Baeuerlein, 2014] entnehmen.

In [Baeuerlein, 2014] wird überdies ein eigener Multi-Level-Tracking-Ansatz beschrieben, bei dem Wissen zu gegenseitigen Objektverdeckungen dynamischer Objekte (Merge-Ereignisse) sowie zu Objektverdeckungen, die durch statische Objekte verursacht werden, einbezogen wird. Szenen mit mehreren Objekten werden mit einer Kamera aus größerer Entfernung aufgezeichnet. In jedem Bild werden zu Beginn Rechtecke um die Regionen von Interesse gelegt, wobei deren Mittelpunkte die Messungen darstellen. Diese werden in der Low-Level-Ebene verarbeitet, in der ein MHT-Framework zur Tracklet-Generierung verwendet wird. Getrackt wird mit Kalmanfiltern. Verschmelzen Messungen miteinander (Merge-Ereignisse), so gehen damit meist Verdeckungen eines oder mehrerer dynamischer Objekte einher. Eine Detektion sich überschneidender Rechtecke wird vorgenommen, um solche Ereignisse zu erkennen. Ist dies der Fall, so erfolgt eine Terminierung der zugehörigen Tracklets, um zu verhindern, dass nachfolgende Messungen nur einem der vorangegangenen Tracklets zugeordnet werden und eine Verdeckung unerkannt bleibt. Stattdessen wird ein neues Tracklet initialisiert, das in der nachgelagerten High-Level-Ebene mit beiden vorangegangenen Tracklets verbunden werden könnte.

Zur Lösung des Zuordnungsproblems in der High-Level-Ebene wird der Ansatz aus [Henriques et al., 2011] zugrunde gelegt, bei dem die Tracklets als Graphen betrachtet werden. Jeder Tracklet-Graph wird um einen Initial- und Terminalknoten erweitert. Beim Verlinken der Tracklets wird in [Baeuerlein, 2014] eine Merge- und Split-Wahrscheinlichkeit

für die jeweils betrachteten Tracklets berechnet. Dafür werden die Farbhistogramme der Objekte in den Rechtecken ausgewertet sowie Abstände der letzten und ersten Messungen von Tracklets zu den Verdeckungsbereichen statischer Objekte, die dafür modelliert wurden. Zur Lösung der Zuordnungswahrscheinlichkeitsmatrix wird der „Auction-Algorithmus“ herangezogen [Bertsekas und Castanon, 1989]. Im Ergebnis entstehen eine oder mehrere Tracklet-Sequenzen, die geschätzte Gesamttrajektorien von Objekten beschreiben. Der Ansatz bringt jedoch die Einschränkung mit sich, dass nur von zwei Objekten bzw. Messungen bei einer Verschmelzung ausgegangen wird.

Die Experimente in [Baeuerlein, 2014] zeigen ein prinzipielles Funktionieren des gewählten Ansatzes. Die modellierten Verdeckungen helfen beim Verlinken der Tracklets. Dennoch zeigte sich insbesondere auch die Bedeutung der zur Verfügung stehenden Appearance-Merkmale. Sind Objekte zu ähnlich, so verschlechtert sich tendenziell die Qualität der Tracklet-Verlinkungen, weil nicht alle Mehrdeutigkeiten aufgelöst werden können.

In dieser Dissertation wird aus den nachfolgend genannten Gründen von einer Übertragung des Ansatzes aus [Baeuerlein, 2014] abgesehen: Das Tracking mit zahlreichen Partikelfilterinstanzen würde im Vergleich zum Tracking mit einer entsprechenden Anzahl an Kalmanfiltern bei der gewählten Likelihood-Berechnung (vgl. Abschnitt 6.4) einen erheblich größeren Aufwand mit sich bringen. Um diesen abzufangen, müssten Partikelfilter mit einer sehr begrenzten Partikelanzahl verwendet werden oder ein Clustering von Partikeln erfolgen, um mehrere Objekte mit einem Filter zu tracken (oder Ähnliches). Da jedes Partikel des Partikelfilters bereits eine Hypothese darstellt (auch der gewichtete Schwerpunkt), jedoch bei der Datenassoziation nur eine einzelne verwendet werden sollte, ist es möglich, eine ungünstige Partikelhypothese dafür auszuwählen. Zudem können Partikel eines Filters auf Messungen liegen, die anderen Partikelfiltern zugeordnet sind. Dies macht die Anwendung schwieriger. Weiterhin ist die Verlinkung von Tracklets, die durch Verdeckungen unterbrochen wurden in [Baeuerlein, 2014] zwar prinzipiell möglich, dennoch setzt die Suche zusammengehöriger Tracklets eine zeitliche Nähe und eine Fortführung des Bewegungsmodells voraus (hier: Constant Velocity Model). Auch wenn die Wirkung des Bewegungsmodells bei hoher Verdeckungswahrscheinlichkeit abgemildert und mit einer Bewegungsunschärfe belegt wird, so bildet die Distanz zwischen der Vorhersage einer Bewegung und einer Messung die Grundlage für eine berechnete Aufenthaltswahrscheinlichkeit eines Objekts. In dieser Dissertation darf die Verdeckungsdauer sehr ausgeprägt sein, wodurch nach einigen Zeitschritten keine sinnvolle Prädiktion aus einem Bewegungsmodell mehr getroffen werden kann. Zudem würden Langzeitverdeckungen ein recht langes Aufrechterhalten bestimmter Zuordnungshypothesen erfordern und könnte den Raum der Hypothesen gegebenenfalls explodieren lassen.

Anstelle einer Datenassoziationstechnik wird die Lösung aus [Canton-Ferrer et al., 2011] für den Umgang mit Zuordnungsproblemen eingesetzt. Jedes Objekt wird mit einem separaten Partikelfilter getrackt, was einen linear steigenden Aufwand bei gleich bleibender Partikelanzahl je Filter bedeutet. Die Filter können aber prinzipiell bei der Likelihood-Berechnung und Anwendung eines Bewegungsmodells sämtliche Voxel des Überwachungsraums in Erwägung ziehen, wodurch sie nicht vollständig unabhängig voneinander agieren. Die Voxeldaten (*unknown*-Voxel) werden nicht segmentiert, um „Einzelmessungen“ zu erzeugen, welche bei der Anwendung von Datenassoziationstechniken benötigt werden würden. Stattdessen wird zur Verhinderung des Verfolgens eines Objekts durch mehrere Filter eine sogenannte **Blocking-Methode** eingesetzt, welche die Interaktionen der Filter berücksichtigt.

Vorschläge für solche Methoden finden sich in [Khan et al., 2003]. Die Idee dabei ist, dass jeder Filter eine Exklusionszone besitzt. Partikel eines Filters, die mit ihren assoziierten Ellipsoiden in eine oder mehrere solcher Zonen hineinfallen, werden bei der Likelihood-Gewichtung entsprechend bestraft. Bei Anwendung von Formel (6.17) wird für jedes Gewicht $w_k^{(i)}$ eines Partikelfilters ein neues Gewicht $\tilde{w}_k^{(i)}$ berechnet. E ist dabei die Menge aller Filter und $f(w_k^{(i)})$ liefert den Filter des betrachteten Partikelgewichts $w_k^{(i)}$. Der Abstand zwischen dem Ellipsoidzentrum des Partikels i und dem Zentrum des gewichteten Schwerpunkts von Filter e (beste Schätzung zum Zeitpunkt $k-1$) wird durch $d(i,e)$ repräsentiert. Über das Produkt fließen für einen Partikel i die Abstände zu allen anderen bestehenden Filtern aus E ein.

$$\tilde{w}_k^{(i)} := \frac{w_k^{(i)}}{\sum_{i=1}^{N} w_k^{(i)}} \prod_{\substack{e=1 \\ e \neq f(w_k^{(i)})}}^{E} \left(1 - \exp^{-0{,}001 \cdot d^2(i,e)}\right) \tag{6.17}$$

Das Ziel der Anwendung einer solchen Blocking-Methode besteht darin, zu verhindern, dass ein Filter, welcher zuerst ein Objekt trackt und damit einen entsprechenden Voxelbereich „belegt“ von einem anderen Filter verdrängt wird. In [Canton-Ferrer et al., 2011] wird davon ausgegangen, dass die Personen immer eine Verbindung zum Boden haben, wodurch jede Exklusionszone nur bezüglich der Ebene des Überwachungsraums definiert ist, die den Boden repräsentiert. Da in dieser Dissertation Bewegungen, die Personen ausführen können, beliebig sein dürfen und beispielsweise auch das Klettern auf einen Tisch erfolgen kann, wird auch die dritte Dimension des Überwachungsraums in die Berechnung einbezogen. Die Blocking-Methode aus Formel (6.17), die etwas von der Methode aus [Canton-Ferrer et al., 2011] abweicht, wird im nachfolgenden Kapitel 7 untersucht und in Kapitel 8 diskutiert.

6.3.3 Track-Initialisierung und Track-Terminierung

Für den Beginn jeder Objektverfolgung wird ein Mechanismus der Objektdetektion benötigt, der zur Lokalisierung unbekannter Objekte im Überwachungsraum dient und für die Track-Initialisierung verwendet wird. Die erfassten Merkmale bei der Objektdetektion können naturgemäß gleich oder ähnlich derer sein, die in der Likelihood-Funktion des Beobachtungsmodells zur Gewichtung der Modellparametrisierungen eingesetzt werden.

Ziel einer Track-Initialisierung ist typischerweise das Zuweisen von Anfangswerten an die veränderlichen Zustandsvariablen sowie an die festen Modellparameter (falls vorhanden). Ebenso können anwendungsabhängig Appearance-Merkmale gewonnen werden, die als Referenz dienen und innerhalb der Likelihood-Funktion Verwendung finden. In dieser Dissertation werden die Ellipsoide mit konstantem Volumen und Durchschnittswerten für die Achsenlängen initialisiert, weil die getrackten Personen eine ähnliche Größe besitzen und sich die Ellipsoide mit ihrer geringen Parameteranzahl zügig in die Messungen einpassen. Bei komplizierteren Objektmodellen hingegen können die Initialwerte stark die Gesamtperformanz des Systems beeinflussen und sollten deshalb möglichst gut bestimmt werden. Bei Echtzeit-Anwendungen des Trackings muss meist auf eine manuelle Parameterbestimmung verzichtet werden, wodurch eine automatisierte **Modellakquise** benötigt wird. Beispielsweise könnten Parameter wie die Personengröße, die Beinlängen oder die Kleidungsfarbe automatisch ermittelt werden. Diese bleiben zwar über den Tracking-Zeitraum hinweg konstant, sie können sich jedoch individuell deutlich unterscheiden und bieten deshalb das Potential einer besseren Differenzierbarkeit der Objekte durch das Beobachtungsmodell, wenn sie möglichst genau bestimmt werden. Dies wiederum kann sich positiv auf die Tracking-Güte und Auflösung von Mehrdeutigkeiten bei Datenassoziationsproblemen auswirken.

In [Canton-Ferrer et al., 2011] werden bei der Objektdetektion für die Track-Initialisierung Personen und Nicht-Personen klassifiziert. Dafür wird zu jedem Zeitpunkt nach Erzeugung einer Visuellen Hülle eine 3D-Komponentenanalyse durchgeführt, wodurch die Visuelle Hülle als eine Menge von Voxelsegmenten repräsentiert wird. Für jedes Segment wird eine Anzahl an Merkmalen berechnet, wie z. B. die Höhe und die Bounding-Box-Ausmaße, die als Eingabe für die Klassifikation dienen. Für den Klassifikator wurden in [Canton-Ferrer et al., 2011] verschiedene Techniken, wie z. B. Mixture of Gaussians oder Neuronale Netze untersucht. Am geeignetsten für das Problem erwiesen sich dabei Entscheidungsbäume. Mithilfe des Klassifikators soll vermieden werden, dass beispielsweise Objekte wie Möbel fälschlicherweise getrackt werden. Auch sollen störende Segmente, die durch Verdeckungen oder Schatteneffekte entstehen, eliminiert werden. In dieser Dissertation werden nach Systemstart keine weiteren Gegenstände in den Raum

eingebracht. Solch eine Klassifikation ist deshalb primär nicht notwendig, könnte jedoch als Erweiterung integriert werden.

In dieser Dissertation werden die Voxeldaten zu jedem Zeitpunkt k durch eine Nachbarschaftsanalyse segmentiert und die resultierenden, zusammenhängenden Segmente für die Track-Initialisierung (Erzeugung neuer Partikelfilter) bezüglich ihres Volumens ausgewertet. Diese Vorgehensweise ist beim Vorliegen von 3D-Rekonstruktionsdaten, die Volumina beschreiben, intuitiv. Segmente, die ein deutlich kleineres Volumen besitzen als ein Mensch, können als Artefakt oder leere Verdeckungsvolumina betrachtet und ignoriert werden. Zu den verbleibenden Segmenten werden die Abstände zu den bereits existierenden Filtern berechnet, um zu erkennen, ob ein Objekt, welches von solch einem Segment repräsentiert wird, bereits getrackt wird. Ist dies nicht der Fall, so wird ein neuer Filter für das entsprechende Segment initialisiert.

Neben der Initialisierung neuer Tracks spielt auch die **Terminierung** bestehender Tracks eine Rolle. Sobald eine Person den Überwachungsraum verlässt, sollte der Track beendet werden. Ebenso, wenn das Objekt innerhalb des Überwachungsraums verloren geht, also der Filter nicht mehr auf einer Messung liegt. Wird das Objekt dann an anderer Stelle erneut detektiert, kann eine **Reinitialisierung** getriggert werden. Gründe für den Verlust eines Objekts innerhalb des Überwachungsraums gibt es verschiedene. Neben einem potentiellen Detektionsverlust, wenn das Objekt (vollständig) verdeckt ist, kann es zu einem Merge von Einzelmessungen (Segmenten) kommen, wenn Objekte zu nah beieinander sind, meist gefolgt von einem Split-Ereignis. Nicht immer ist es möglich, in solchen Fällen sämtliche Objekte erfolgreich weiter zu tracken, auch bei Verwendung der bereits vorgestellten Blocking-Methode. Detektierte Merkmale beim Background Subtraction können unzureichend sein, beispielsweise bei gegenseitigen Objektverdeckungen oder geringer Unterscheidbarkeit von Vordergrund und Hintergrund.

Für eine Reinitialisierung wird angenommen, dass die maximale Anzahl an Personen, die sich zu einem Zeitpunkt im Überwachungsraum befinden, bekannt ist. Gesetzt den Fall mehrere Filter liegen nah beieinander auf einer Messung und eine größere Messung (Segment) ohne zugehörigen Filter tritt in unmittelbarer Nähe auf, so wird dies als Split-Ereignis gewertet. In Folge wird einer der bestehenden Filter terminiert und ein neuer Filter für das „unbelegte“ Voxelsegment initialisiert. Um festzustellen, ob eine Person den Überwachungsraum verlässt oder innerhalb des Überwachungsraums verloren geht, kann das maximale Gewicht des Filters herangezogen werden. Liegt dieses unterhalb eines definierten Schwellenwerts, so wird eine Terminierung eingeleitet. Dieser Schwellenwert sollte so gewählt werden, dass ein Filter nicht terminiert, wenn sich die Person teilweise oder vollständig in einem Verdeckungsvolumen befindet. Ist es nicht möglich, für den gesamten Überwachungsraum einen einheitlichen Schwellenwert zu

verwenden (z. B. bei lokal unterschiedlichen Gewichten für die *occluded*-Voxel), so könnte alternativ die Zusammensetzung der Voxelzustände innerhalb des Partikelellipsoids mit maximalem Gewicht für die Terminierung ausgewertet werden. Das Terminieren könnte dann vorgenommen werden, wenn die Summe der *unknown*- und *occluded*-Voxelhäufigkeiten unter einem definierten Schwellenwert liegt. Für die Auswertung wird das Ellipsoidvolumen wieder als konstant angenommen. Liegen die betrachteten Ellipsoide auf der Person, so sollten sie recht gut mit Voxeln der Zustände *unknown* und *occluded* gefüllt sein. Ist jedoch ein größerer Teil des Elliposids mit Voxeln der Zustände *empty* und/oder *filled* belegt, darf davon ausgegangen werden, dass sich der Filter nicht auf dem Objekt und nicht innerhalb eines größeren Verdeckungsvolumens befinden kann. Eine Terminierung kann entsprechend vorgenommen werden.

6.4 Likelihood-Funktion

Für die Gewichtung eines Partikels j mit der Likelihood-Funktion $p(z_k|x_k^{(j)})$ werden aus der Messung bzw. Beobachtung zu jedem Zeitpunkt k Merkmale extrahiert, mit Hilfe derer die Übereinstimmung eines parametrisierten Ellipsoidmodells (Hypothese) mit der Beobachtung bewertet werden kann.

Bei 3D-Rekonstruktionsdaten bietet es sich an, den Grad der geometrischen Einpassung jeder Ellipsoidhypothese auf Basis der gemessenen Raumbelegung (belegte Voxel) zu bestimmen. Definiert sei eine Funktion getVoxelInEllipsoid : $\mathbb{R}^3 \times \mathbb{R}^3 \rightarrow V = \{v_1, v_2, .., v_n\}$ wie in Formel (6.18) angegeben, die für ein Partikel j mit seinen Ellipsoidparametern (Position des Zentrums und Achsenlängen) die Menge an Voxeln M_k^j zurückliefert, die sich zum Zeitpunkt k im Inneren des Ellipsoids befindet. Es gilt $M_k^j \subseteq V$: die Voxelmenge M_k^j ist eine Teilmenge des Voxelraums V. Voxel, die ein Ellipsoid mit seinem Rand schneidet und nicht vollständig enthält, gehören aus konservativen Gründen ebenfalls zur Menge M_k^j.

$$M_k^j = \text{getVoxelInEllipsoid}(x_k^{(j)}) \tag{6.18}$$

Als Randbedingung soll die gewählte Parametrisierung einer Ellipsoidhypothese immer mit einem konstanten Ellipsoidvolumen einhergehen. Dies ist eine legitime Annahme, da sich das Volumen einer Person in einem kurzen Zeitabschnitt nicht merklich verändert. Die Erfüllung dieser Bedingung wird innerhalb des Resampling-Schritts (vgl. Abschnitt 6.2.5) gewährleistet.

Basierend auf allen Voxelzuständen der Menge M_k^j wird die zugehörige Ellipsoidhypothese $\mathbf{x}_k^j$ gewichtet. Die Autoren in [Canton-Ferrer et al., 2011] verwenden hierfür die binäre

Belegungsinformation der Voxel. Ein Voxel ist „belegt“, wenn er zur Visuellen Hülle gehört. Andernfalls ist er „frei“. In der Likelihood-Funktion erhält jedes belegte Voxel aus der Menge M_k^j ein positives Gewicht, während die freien Voxel ignoriert werden. Voxel, die von a priori gegebenen statischen Objekten wie beispielsweise einem Tisch belegt sind, oder die sich in einem leeren Verdeckungsvolumen befinden, werden in [Canton-Ferrer et al., 2011] als frei angenommen. Sie sind in einem Standardalgorithmus der Visuellen Hülle nicht Teil der Rekonstruktion.

In dieser Dissertation stehen weitere Informationen zur Verfügung, die im Folgenden beschrieben werden. Zu jedem Zeitpunkt k werden die Zustände aller n Voxel des Überwachungsraums $V = \{v_i\}$ mit $i = 1, .., n$ beobachtet. Bei der Rekonstruktion nach Algorithmus 5.3 erhält jedes Voxel v_i einen von vier möglichen Zuständen durch die Funktion $\text{voxelValue}_k : \mathcal{V} \rightarrow \{\textit{filled}, \textit{occluded}, \textit{unknown}, \textit{empty}\}$, vergleiche Formel (5.11). Die Bedeutung dieser Zustände für die Belegung des Überwachungsraums wird nun kurz wiederholt.

a) Durch die Voxelisierung der modellierten Objekte (gegeben in Form von Dreiecksnetzen) wird eine Voxelmenge bestimmt, für die gilt: $\text{voxelValue}_k(v_i) = \textit{filled}$. Diese Voxel repräsentieren die bekannten physisch belegten Volumina der statischen Objekte des Überwachungsraums.

b) Mithilfe berechneter Tiefenbilder von den 3D-Modellen wird eine Tiefenhülle zu den statischen Objekten generiert. Für Voxel, die in allen Kameras verdeckt sind und gleichzeitig nicht zu den Voxeln des Zustands *filled* gehören, gilt: $\text{voxelValue}_k(v_i) = \textit{occluded}$. Diese Voxel sind initial, d. h. ohne die Anwesenheit dynamischer Objekte im Überwachungsraum, leer.

c) Bei der Online-Rekonstruktion der Visuellen Hülle werden die Voxel bestimmt, die das Volumen der sichtbaren Teile der dynamischen Objekte bzw. Personen approximieren. Für diese Voxel gilt: $\text{voxelValue}_k(v_i) = \textit{unknown}$. Nicht alle diese Voxel müssen tatsächlich von dynamischen Objekten okkupiert sein. Aus der Rekonstruktion lassen sich jedoch keine näheren Informationen darüber gewinnen, welche dieser Voxel zu leeren Artefakten oder zu leeren Verdeckungsvolumina gehören, die von dynamischen Objekten verursacht sind.

d) Für alle verbleibenden Voxel gilt: $\text{voxelValue}_k(v_i) = \textit{empty}$. Unter den gegebenen Annahmen zu Objekten (vgl. Abschnitt 2.4) und einer idealen Verarbeitung der Kameradaten (inklusive Background Subtraction) sind diese Voxel auch tatsächlich leer.

Die Menge M_k^j bestehend aus allen Voxeln innerhalb eines Ellipsoids lässt sich abhängig von den zugeordneten Voxelzuständen in die disjunkten Untermengen $\Upsilon_k^j, \Psi_k^j, \Phi_k^j$ und

E_k^j aufteilen, wobei Formel (6.19) gilt:

$$M_k^j = \Upsilon_k^j \,\dot\cup\, \Psi_k^j \,\dot\cup\, \Phi_k^j \,\dot\cup\, \mathrm{E}_k^j \tag{6.19}$$

Die Untermengen ergeben sich nach Formel (6.20):

$$\begin{aligned}
\Upsilon_k^j &= \{v_i \in M_k^j | \mathrm{voxelValue}_k(v_i) = \textit{unknown}\} \\
\Psi_k^j &= \{v_i \in M_k^j | \mathrm{voxelValue}_k(v_i) = \textit{occluded}\} \\
\Phi_k^j &= \{v_i \in M_k^j | \mathrm{voxelValue}_k(v_i) = \textit{filled}\} \\
\mathrm{E}_k^j &= \{v_i \in M_k^j | \mathrm{voxelValue}_k(v_i) = \textit{empty}\}
\end{aligned} \tag{6.20}$$

Jedem Voxel $v_i \in M_k^j$ wird abhängig von seinem Zustand ein Gewicht zugewiesen. Dazu werden die Parameter $\upsilon, \psi, \phi, \epsilon \in \mathbb{R}$ verwendet. Mit Hilfe der einzelnen Gewichte aller Voxel kann das gesamte Gewicht $w_k^{(j)}$ einer Partikelhypothese $x_k^{(j)}$ bestimmt werden. Jedes Partikelgewicht wird durch die Summe S aller Partikel- bzw. Ellipsoidgewichte aus Formel (6.21) normiert, womit sich ein Wertebereich für die Likelihood-Funktion von $[0, 1]$ in Formel (6.22) ergibt. Abhängig von den Werten der Gewichtsparameter könnten auch negative Gewichte entstehen, worauf an späterer Stelle noch näher eingegangen wird. Diese werden in Formel (6.22) auf 0 gesetzt.

$$S = \sum_j w_k^j \tag{6.21}$$

$$p(\mathbf{z}_k | \mathbf{x}_k^{(j)}) \approx \max\left(\frac{w_k^j}{S}, 0\right) \tag{6.22}$$

Im Anschluss an die Gewichtung aller Partikel eines Filters durch die Likelihood-Funktion wird das SIR-Sampling ausgeführt, das in Abschnitt 6.2.5 vorgestellt wurde. Nachfolgend wird eine Bewegungsprädiktion für alle Partikel vorgenommen (vgl. Abschnitt 6.3.1).

Im Weiteren wird erörtert, welche Möglichkeiten zur Gewichtung der Voxelzustände innerhalb der Likelihood-Funktion sinnvoll genutzt werden können. Der Einfluss der jeweiligen Gewichtung wird dabei qualitativ an einem Beispiel veranschaulicht und diskutiert, unabhängig von den anderen Verfahrensschritten des Resamplings und der Bewegungsprädiktion. Experimente zu den Überlegungen sind Gegenstand von Kapitel 7.

Als Beispiel dient die schematische Darstellung aus Abb. 6.3, welche als zweidimensionale Aufsicht eines 3D-Raums verstanden werden kann. Auf eine Voxeldiskretisierung wird

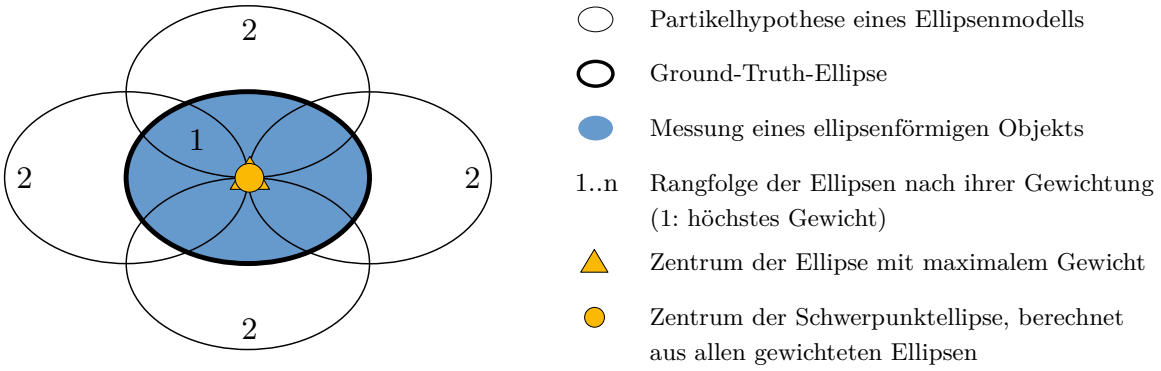


Abb. 6.3: 2D-Beispiel mit Legende zur Veranschaulichung der Gewichtung von Voxelzuständen in der Likelihood-Funktion bei der Verwendung eines ellipsenförmigen Objektmodells.

dabei verzichtet, um die Darstellung einfach zu halten. Zu sehen sind fünf Ellipsen, die an unterschiedlichen Stellen platziert sind. Jede Ellipse repräsentiert dabei ein 3D-Ellipsoid, das mit einem Partikel assoziiert ist. Die Ellipse mit stärkerem schwarzen Rand repräsentiert das Ground-Truth-Ellipsoid, welches den gesuchten Zustand des verfolgten Objekts widerspiegelt. Das getrackte Objekt hat in dem Beispiel die Form eines Ellipsoids. Eine zur Veranschaulichung gewählte gleichmäßige Abtastung des Raums um das gesuchte Objekt wird durch die gewählte Anordnung der vier äußeren Ellipsen symbolisiert. Weiterhin sei die Messung der Raumbelegung gegeben (blauer Bereich), die in Abb. 6.3 die Ground-Truth-Ellipse vollständig ausfüllt. Es wird davon ausgegangen, dass alle dargestellten Ellipsen bzw. Ellipsoide die gleichen Achsenlängen besitzen und sich nur in ihrer Position unterscheiden. Dabei sollen die vier äußeren Ellipsen in gleichem Maße die Ground-Truth-Ellipse schneiden, was in Abb. 6.3 auch dem gleichen Anteil an der gemessenen Raumbelegung (blau) entspricht.

Von den Voxeln, die innerhalb einer zugehörigen Ellipsoidhypothese liegen, erhalten in Formel (6.23) nur Voxel des Zustands *unknown* ein positives Gewicht (υ). Dies entspricht in Abb. 6.3 der Multiplikation des Gewichts υ mit dem Inhalt der Fläche, die sich aus dem Verschnitt der Ellipse mit dem Messbereich ergibt. Je größer demnach die gemeinsame Schnittfläche ist (in 3D: das gemeinsame Schnittvolumen und damit die Anzahl der Voxel mit Zustand *unknown*), umso größer ist auch das zugewiesene Gewicht. Das Gewicht wird in Formel (6.23) und den nachfolgenden Formeln mit der Voxelanzahl $|M_k^j|$, die zum Ellipsoid gehört, normiert. Damit wird einer ungleichen Kardinalität der Voxelmengen verschiedener Ellipsoide, die – trotz der Nebenbedingung eines konstanten Ellipsoidvolumens – aufgrund von Diskretisierungsfehlern entstehen kann, entgegengewirkt.

$$w_k^j = \frac{\upsilon \cdot |\Upsilon_k^j|}{|M_k^j|} \tag{6.23}$$

Vergleicht man in Abb. 6.3 alle Ellipsen bezüglich ihrer Schnittfläche mit der Messung, so ist erkennbar, dass die Ground-Truth-Ellipse das höchste Gewicht erhalten muss, da sie die Messung vollständig enthält. Sie bekommt den Rang 1 zugewiesen. Die Schnittflächen der anderen vier Ellipsen mit der Messung sind gleich groß, weshalb die zugehörigen Partikel auch das gleiche Gewicht erhalten. Sie bekommen alle den Rang 2 zugewiesen. Für die Schätzung des Partikelfilters wird häufig entweder die Hypothese des größten maximalen Gewichts gewählt (gelbes Dreieck) oder es wird der Schwerpunkt aller gewichteter Hypothesen gebildet (gelber Punkt). Der Übersicht halber werden nur die Zentren dieser beiden Ellipsen als Schätzung durch die gelben Symbole (Kreis und Dreieck in Abb. 6.3) visualisiert. Das Schwerpunktellipsoid selbst ist nicht dargestellt, es besitzt jedoch die gleiche Form wie die eingezeichneten Ellipsen. In Abb. 6.3 stimmt die Schwerpunktschätzung mit dem Ellipsoid maximalen Gewichts überein und entspricht der Ground-Truth-Ellipse. Dies liegt an der gleichen Gewichtung der außenliegenden Ellipsen, da deren Schnittflächen mit der Messung gleich groß sind.

6.4.1 Gewichtung von Verdeckungsvolumina

Anhand von Abb. 6.4 soll nun die Gewichtung der verdeckten *occluded*-Voxel diskutiert werden. Die Motivation zu deren positiver Gewichtung liegt darin, dass sie zur Laufzeit von Personen oder anderen Objekten belegt werden können. Ohne die Berücksichtigung verdeckter Volumina könnte ein Filter das zu trackende Objekt verlieren, insbesondere wenn es sich vollständig in einem Verdeckungsvolumen befindet. Möchte man Objekte auch dann weiterverfolgen, wenn diese aus dem Sichtbereich verschwunden sind, so besteht eine Möglichkeit darin, Verdeckungen als Messung zu behandeln und ebenfalls in die Gewichtung der Partikelhypothesen einzubeziehen (Pseudomessung).

Zur Veranschaulichung des Einflusses der Gewichtung verdeckter Voxel auf die Schätzung in Abb. 6.4 werden in jeder Zeile drei verschiedene Verdeckungssituationen dargestellt. Der verdeckte Bereich wird von dem gestrichelten Rechteck repräsentiert. In der ersten Spalte wird das ellipsenförmige Objekt zu etwa einem Drittel verdeckt. Das bedeutet ein Drittel der Messung „fehlt“. In der zweiten Spalte werden ca. zwei Drittel des Objekts verdeckt und in der dritten Spalte wird das Objekt vollständig verdeckt. Die Ground-Truth-Ellipse befindet sich innerhalb des gestrichelten Rechtecks. In Zeile 1 von Abb. 6.4 wird die Gewichtung nach Formel (6.23) dargestellt. Es steht kein Wissen zu den Verdeckungsvolumina zur Verfügung. Der fehlende Teil der Messung trägt nicht zur Gewichtung bei. Betrachtet man Abb. 6.4(a), so sieht man im Vergleich zu Abb. 6.3 eine Änderung der Gewichte. Die Ellipsen, die den verdeckten Objektteil enthalten, bekommen ein geringeres Gewicht als die linke äußere Ellipse, die nicht von der Verdeckung betroffen ist. Die Schwerpunktellipse liegt nicht mehr auf der Ground-

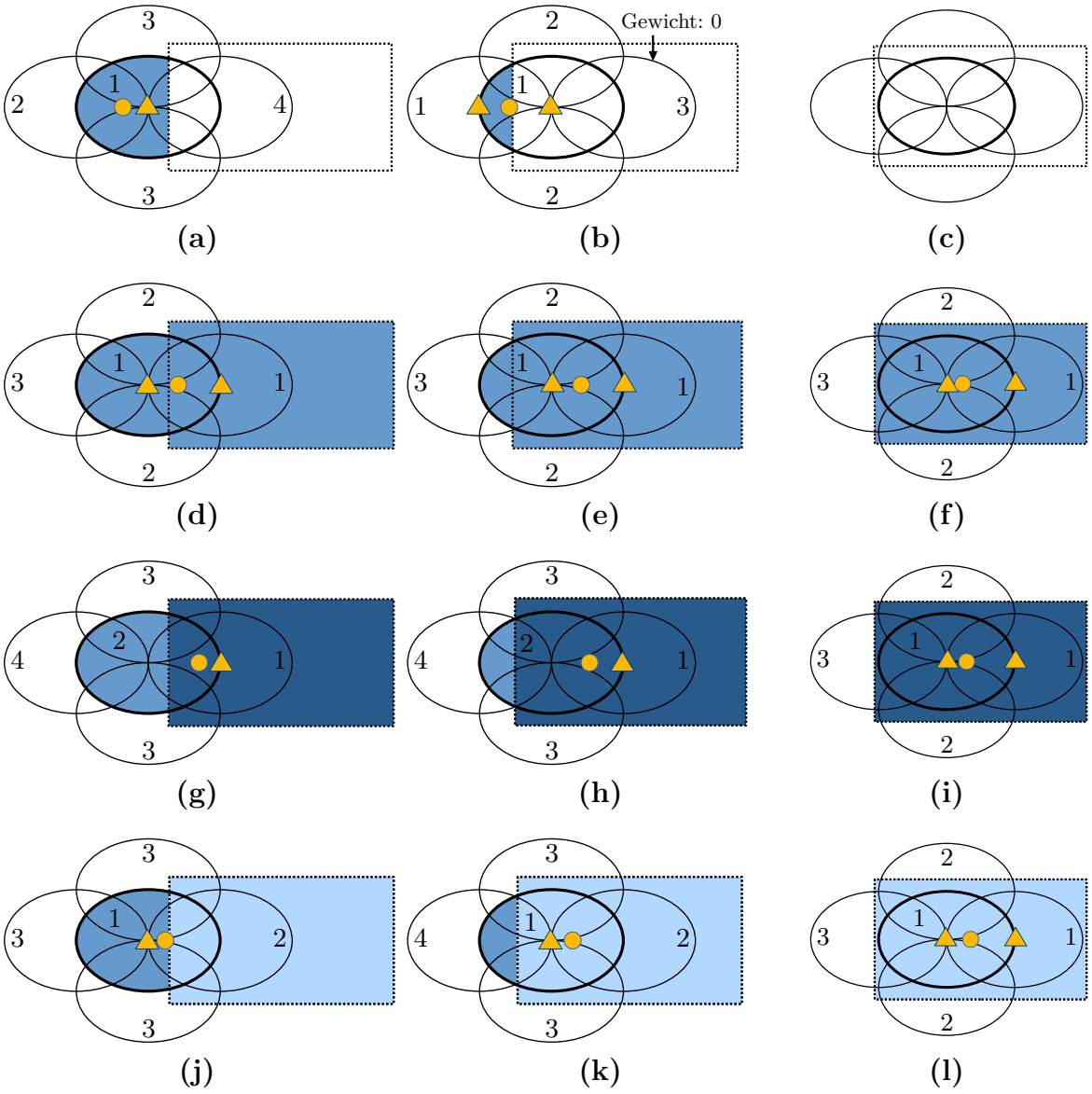


Abb. 6.4: 2D-Beispiel zur Darstellung des Einflusses der Gewichtung von Verdeckungsvolumina in der Likelihood-Funktion. Dargestellt sind unterschiedliche Grade der Verdeckung eines ellipsenförmigen Objekts (stärkerer schwarzer Rand) in den Spalten. Zeile 1: Nur die Messung (blau) geht in die Gewichtung ein. Zeile 2: Der verdeckte Bereich wird mit dem gleichen positiven Faktor gewichtet wie die Messung ($\psi = \upsilon$). Zeile 3: Der verdeckte Bereich wird stärker gewichtet als die Messung ($\psi > \upsilon$). Zeile 4: Der verdeckte Bereich wird geringer gewichtet als die Messung ($\psi < \upsilon$).

Truth-Ellipse. In Abb. 6.4(b) erhalten die mittlere Ellipse und die linke äußere Ellipse das gleiche Gewicht, woraus sich zwei Maxima für die Schätzung ergeben. Das Zentrum der Schwerpunktellipse hat sich im Vergleich zu (a) weiter von der Ground-Truth-Ellipse entfernt. In (c) befindet sich das Objekt vollständig in der Verdeckung, weshalb keine Messung mehr auftritt. Der Filter verliert in diesem Fall das Objekt und würde ohne weitere Maßnahmen, wie z. B. eine Fortführung der Bewegungsprädiktion, terminieren.

Im dargestellten Fall von Zeile 2 in Abb. 6.4 erhält der verdeckte Bereich auch eine Gewichtung, so wie in Formel (6.24) angegeben. Der Gewichtungsparameter υ für die Voxel des Zustands *unknown* und der Gewichtungsparameter ψ für die Voxel des Zustands *occluded* stimmen dabei überein ($\psi = \upsilon$). In den Abbildungen von Zeile 2 ist deshalb der verdeckte Bereich im gleichen Blauton gehalten wie der Messbereich.

$$w_k^j = \frac{\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|}{|M_k^j|} \quad \text{mit} \quad \psi = \upsilon \tag{6.24}$$

Betrachtet man Abb. 6.4(d), so ist zu sehen, dass die mittlere und die rechte Ellipse das gleiche maximale Gewicht erhalten. Obwohl die Ground-Truth-Ellipse in der Mitte noch zu zwei Drittel aus der Verdeckung schaut, muss sie sich den Rang bei der Gewichtung mit der rechten Ellipse teilen, obwohl diese fast vollständig verdeckt ist. Auch in den Abbildungen 6.4(e) und (f) bleibt die gleiche Gewichtung der mittleren und der rechten Ellipse erhalten, da jeweils beide dieser Ellipsen vollständig gefüllt sind und bei der Gewichtung nicht zwischen Messung und Verdeckung unterschieden wird. In Abb. 6.4(f) ist das Objekt vollständig verdeckt und auch die linke äußere Ellipsenhypothese befindet sich zur Hälfte im verdeckten Bereich. Deshalb verschiebt sich die geschätzte Schwerpunktellipse im Vergleich zu den Abbildungen 6.4(d) und (e) wieder in Richtung Ground Truth. Dies hängt mit der speziellen Platzierung der Ellipsen bezüglich der Verdeckung zusammen. Wäre die Verdeckung so groß, dass alle Ellipsen vollständig in der Verdeckung liegen würden, so würde die Schwerpunktellipse aufgrund der symmetrischen Anordnung der Ellipsenhypothesen wieder auf der Ground-Truth-Ellipse liegen. Bei dem gewählten Beispiel ergibt sich damit eine gute Schwerpunktschätzung. Dennoch ist die Aussagekraft von Schätzungen innerhalb und um Verdeckungen herum begrenzt, was man daran erkennt, dass sich mehrere Maxima bzw. insgesamt hohe Gewichte ausbilden können.

In Zeile 3 von Abb. 6.4 wird der Fall gezeigt, bei dem die Verdeckungen entsprechend Formel (6.25) stärker gewichtet werden als die Messungen.

$$w_k^j = \frac{\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|}{|M_k^j|} \quad \text{mit} \quad \psi > \upsilon \tag{6.25}$$

Zur Verdeutlichung dieses Sachverhalts ist der verdeckte Bereich in einem dunkleren Blauton dargestellt als der Messbereich. Für die Gewichtsparameter der Voxelzustände bedeutet dies, dass die Voxel vom Zustand *unknown* ein geringeres Gewicht erhalten als die Voxel vom Zustand *occluded* ($\psi > \upsilon$). Bei Abb. 6.4(g) ist zu erkennen, dass die rechte Ellipse, die am meisten verdeckt ist, das stärkste Gewicht erhält und damit alleinig die maximale Schätzung repräsentiert. Zusätzlich zieht sie auch den geschätzten Schwerpunkt stark auf sich. Damit verschlechtert sich die Situation gegenüber Abb. 6.4(d),

denn obwohl zwei Drittel des Objekts messbar sind, wird nicht die Ground-Truth-Ellipse am stärksten gewichtet, die die vorhandene Messung vollständig enthält, sondern die komplett verdeckte Ellipse auf der rechten Seite. Dies gilt auch für die Darstellung in Abb. 6.4(h). Bei dem Fall in Abb. 6.4(i), bei dem keine Messung mehr existiert, entspricht die Schätzung dem Ergebnis aus Abb. 6.4(f). Die mittlere und die rechte Ellipse erhalten dabei das gleiche und größte Gewicht.

In Zeile 4 von Abb. 6.4 werden Voxel des Zustands *occluded* geringer gewichtet als Voxel des Zustands *unknown* ($\psi < \upsilon$), wie in Formel (6.26) vermerkt.

$$w_k^j = \frac{\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|}{|M_k^j|} \quad \text{mit} \quad \psi < \upsilon \tag{6.26}$$

Die Messung wird demnach gegenüber der Verdeckung präferiert. Dies wird in Zeile 4 durch den helleren Blauton der Verdeckung verdeutlicht. In den Darstellungen der Abb. 6.4(j) und (k) stimmt die Ellipse maximalen Gewichts mit der Ground-Truth-Ellipse überein und auch die berechnete Schwerpunktellipse liegt nur etwas daneben. In Abb. 6.4(l) hingegen fehlt wieder die Messung, wodurch zwei Ellipsen das maximale Gewicht erhalten, wie auch in den Abbildungen 6.4(f) und (i).

In Abb. 6.5 werden Gewichtsfunktionen zu den Formeln (6.23) bis (6.26) visualisiert. Dabei ist das Verhältnis der Kardinalitäten der Voxelmengen des Zustands *occluded* ($|\Psi_k^j|$) und des Zustands *unknown* ($|\Upsilon_k^j|$) zur Kardinalität der Menge aller Voxel eines Ellipsoids $|M_k^j|$ jeweils auf zwei Achsen aufgetragen. Der Anteil der übrigen Voxel (mit den Zuständen *empty* und *filled*) ergibt sich zu: $1 - |\Psi_k^j|/|M_k^j| - |\Upsilon_k^j/|M_k^j|$. Die Gewichtsfunktion von Abb. 6.5(a) entspricht der Formel (6.23). Nur Voxel des Zustands *unknown* erhalten ein Gewicht. In Abb. 6.5(b) wird die Messung und die Verdeckung nach Formel (6.24) identisch gewichtet. In der Abb. 6.5(c) erhält die Verdeckung ein größeres Gewicht als die Messung (vgl. Formel (6.25)) und in der Abb. 6.5(d) ein kleineres Gewicht (vgl. Formel (6.26)). Bei (c) und (d) wird der Faktor $1/e$ zur Gewichtung einer der beiden Voxelanteile verwendet. Hierbei könnte auch ein anderer Faktor eingesetzt werden. Anmerkung: In den Abbildungen 6.5 und 6.6 wird die Variable s gleichbedeutend zum Gewicht w und die Variable t gleichbedeutend für den Zeitpunkt k verwendet.

Die symmetrische Platzierung der Ellipsen um die Ground-Truth-Ellipse in Abb. 6.4 erleichtert den qualitativen Vergleich der beschriebenen Gewichtungen. Diese Anordnung berücksichtigt jedoch nicht den Resampling-Schritt, der jeweils im Anschluss an eine Gewichtung aller Partikel durchgeführt wird. Dort wo hohe Gewichte entstehen, werden dabei proportional viele neue Partikel erzeugt, und dort wo geringe Gewichte entstehen, entsprechend weniger Partikel. Anschließend werden die Partikel mit dem Bewegungs-

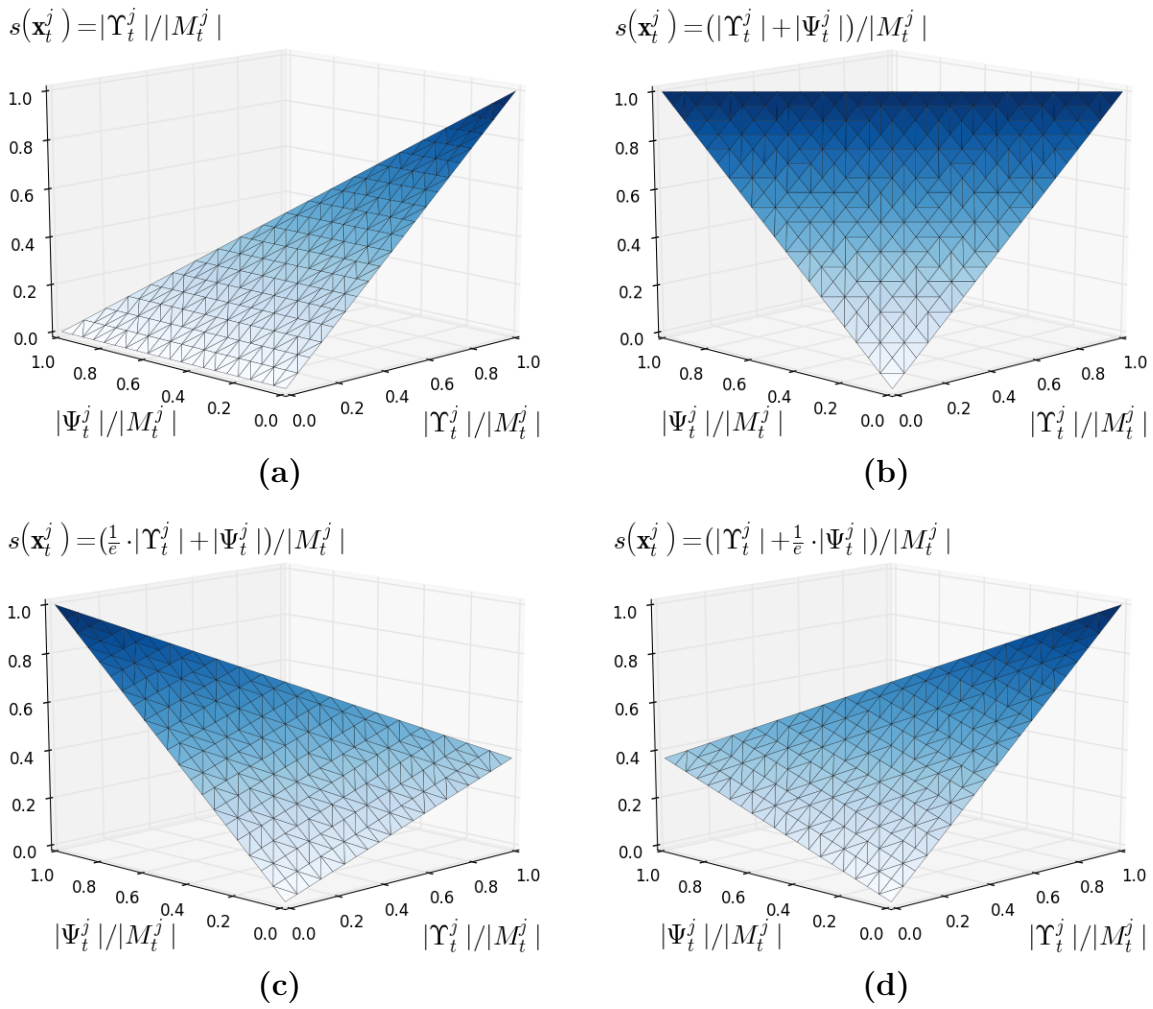


Abb. 6.5: Gewichtungsfunktionen zu den Formeln (6.23) bis (6.26) in (a) bis (d).

modell aus Abschnitt 6.3.1 neu positioniert. Dadurch entsteht eine Filterbewegung. Betrachtet man die vermutete Filterbewegung für die gegebenen Gewichtungsvarianten in Abb. 6.4, so ist Folgendes festzuhalten:

Fall $\psi = \upsilon$:

In Zeile 2 von Abb. 6.4 ist zu sehen, dass die Ellipsen, welche die Verdeckung schneiden, die höchsten Gewichte erhalten. Das Resampling und das Bewegungsmodell würden dafür sorgen, dass sich nach einigen Schritten mehr Partikel innerhalb der Verdeckung befinden als außerhalb. Die resultierende Filterbewegung wäre eine Diffusion in die Verdeckung, was durch die Eigenbewegung des zu verfolgenden Objekts und der damit verschwindenden Messung außerhalb der Verdeckung noch verstärkt werden würde. In Abb. 6.4 ist die Verdeckung größer als das messbare Objekt. Damit ist zu erwarten, dass der Filter der Person nicht wieder aus der Verdeckung folgen kann, weil dafür

zu viele Partikel innerhalb der Verdeckung ein hohes Gewicht erhalten. Für eine Objektverfolgung müssten Partikel, die auf der Messung liegen, über mehrere Zeitschritte höhere Gewichte erhalten als Partikel in der Verdeckung. Dies dürfte jedoch bei einer identischen Gewichtung von Messung und Verdeckung schwierig werden, weil letztere größer ist. Denn nur Partikel, welche die Messung exakt treffen, würden das gleiche Gewicht liefern wie eine Vielzahl an Ellipsen, die in der größeren Verdeckung liegen. Alternativ müsste die Filterprädiktion zu einer größeren Verschiebung der Partikel weg von der Verdeckung führen, was bei dem verwendeten Bewegungsmodell nicht auftritt.

Fall $\psi > \upsilon$:

Bei dieser Gewichtung, die in Zeile 3 von Abb. 6.4 visualisiert ist, wird die Situation des vorhergehenden Falls noch verschärft. Durch das höhere Gewicht, das die Verdeckung gegenüber der Messung erhält, dürfte der Filter noch schneller in den verdeckten Bereich diffundieren. Je höher die Verdeckung im Vergleich zur Messung gewichtet wird und je größer die Verdeckung im Vergleich zum Objekt ist, desto unwahrscheinlicher wird der Filter dem Objekt aus der Verdeckung folgen können.

Fall $\psi < \upsilon$:

Wird die Verdeckung geringer gewichtet als die Messung (vgl. Zeile 4 von Abb. 6.4), dann werden die Partikel länger auf der Messung (und damit auf dem Objekt) gehalten, weil die Partikel dort häufiger resampelt werden. Dies ist umso stärker ausgeprägt, je geringer das Gewicht der Verdeckung gegenüber der Messung ausfällt. Verschwindet die Messung wie im Beispiel der Abb. 6.4 (bei Betrachtung der Spalten von links nach rechts), so gibt es einen Zeitpunkt, zu welchem die Mehrheit der Partikel, die in die Verdeckung gestreut werden, ein höheres Gewicht erhalten. Sobald dann eine neue Messung an der Verdeckung auftritt, weil sich die Person aus der Verdeckung bewegt, werden Partikel die dorthin gestreut werden durch das höhere Gewicht der Messung stärker resampelt, sofern das Objekt schon weit genug aus der Verdeckung herausschaut. In diesem Fall sollte es gelingen, der Person aus der Verdeckung zu folgen.

Anhand des gewählten Beispiels in Abb. 6.4 lässt sich nachvollziehen, dass die Objektverfolgung auch durch Verdeckungen möglich ist, wenn diese in die Gewichtung einbezogen werden. Nach Beurteilung der sich ergebenden Filterbewegungen für die unterschiedlichen Gewichtungen ist die Variante zu bevorzugen, bei der die Verdeckung geringer gewichtet wird als die Messung ($\psi < \upsilon$). Dadurch erhält der Filter die Möglichkeit, die Verdeckung zusammen mit der Person auch wieder zu verlassen. Andernfalls würde der Filter in der Verdeckung verbleiben.

Wie groß der Abstand zwischen den Gewichten von Messung und Verdeckung sein muss, hängt von dem spezifischen Tracking-Szenario ab, insbesondere von den Eigenschaften des Verdeckungsvolumens. Die Tendenzen lassen sich jedoch klar beschreiben: Je ähnlicher die Gewichte von Verdeckung und Messung sind, umso weniger wird die Messung gegenüber der Verdeckung von dem Filter bevorzugt und umso mehr Partikel entstehen an und in der Verdeckung. Umgekehrt steigert man die Konzentration der Partikel auf der Messung, wenn man die Verdeckung deutlich niedriger gewichtet. Hierbei ist allerdings zu beachten, dass damit eine Angleichung der Gewichtung der *occluded*-Voxel mit den übrigen Voxeln der Zustände *filled* und *empty* einhergeht, die bei der bisherigen Betrachtung ein Gewicht von 0 erhalten haben. Damit können Partikel auch weniger deutlich in die Verdeckung streuen, was wiederum nicht unbedingt wünschenswert ist, wenn man davon ausgeht, dass sich ein gesuchtes Objekt wahrscheinlicher in der Verdeckung befindet als im leeren Raum und in Bereichen, die von statischen Objekten belegt werden.

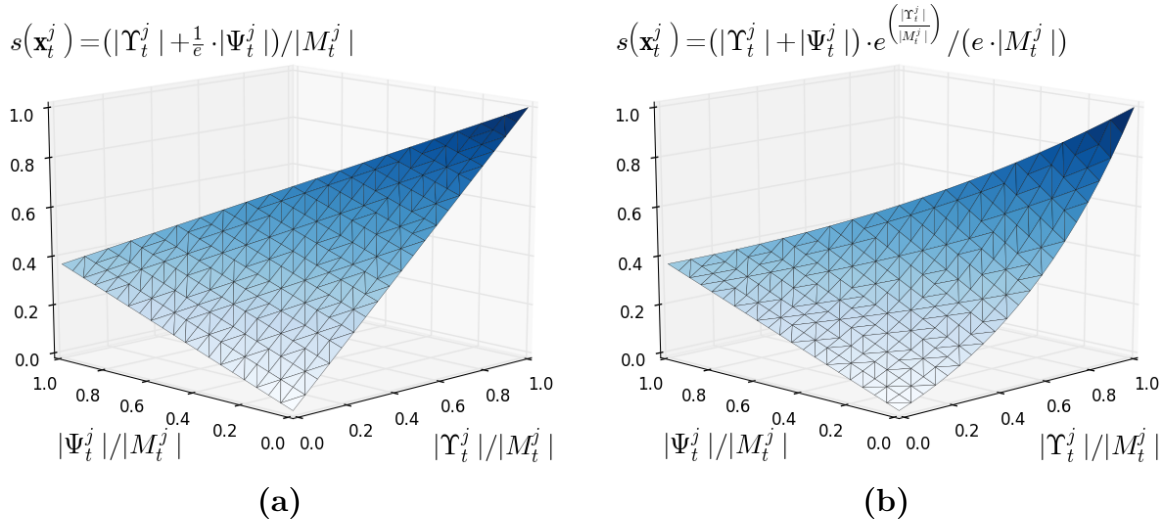


Abb. 6.6: Gewichtungsfunktionen zu den Formeln (6.26) und (6.27) in (a) und (b)

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{e^{\frac{|\Upsilon_k^j|}{|M_k^j|}}}{e} \tag{6.27}$$

Die additive Gewichtung von Abb. 6.6(a), die resultierend aus den bisherigen Überlegungen gewählt wird, wird nun in Formel (6.27) mit einem nichtlinearen Term multipliziert, durch welchen der Anteil der Messung gegenüber dem der Verdeckung exponentiell verstärkt wird. Eine Funktion dazu ist in Abb. 6.6(b) dargestellt. Für diese gilt zwar: $\psi = \upsilon$. Durch den nichtlinearen Term ergibt sich jedoch implizit eine Reduk-

tion des Gewichts für die *occluded*-Voxel, weshalb die Funktion mit der Funktion in Abb. 6.6(a) vergleichbar ist. Der Effekt dieser Gewichtung wird in Kapitel 7 untersucht. Die Gewichtungsvariante selbst wird im weiteren Verlauf dieser Dissertation auch als „nichtlineare Verstärkung“ (engl. Gain) bezeichnet.

6.4.2 Bestrafung der Voxelzustände *empty* und *filled*

Bei den bisher vorgestellten Likelihood-Funktionen erhalten Voxel der Zustände *empty* und *filled* kein Gewicht, weil davon ausgegangen wird, dass sich Personen und andere dynamische Objekte nicht in diesen Voxeln befinden. Die zwei Voxelzustände werden dabei semantisch gleichbehandelt und in ihrer Gewichtung nicht voneinander unterschieden (Gewicht von 0). Der Gedanke liegt nahe, diese Voxel explizit zu bestrafen. Für die Anzahl der Voxel aus beiden Mengen gegeben als $(|M_k^j| - |\Upsilon_k^j| - |\Psi_k^j|)$ könnte entsprechend Formel (6.28) eine negative Gewichtung mit dem Faktor c in die Likelihood-Funktion einfließen.

$$\begin{aligned} w_k^j &= \frac{\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j| - c \cdot (|M_k^j| - |\Upsilon_k^j| - |\Psi_k^j|)}{|M_k^j|} \\ &= \frac{(\upsilon + c) \cdot |\Upsilon_k^j| + (\psi + c) \cdot |\Psi_k^j| - c \cdot |M_k^j|}{|M_k^j|} \end{aligned} \tag{6.28}$$

Aufgrund der linearen Abhängigkeit der Menge aller Voxel vom Zustand *empty* und *filled* von den anderen beiden Voxelmengen der Zustände *unknown* und *occluded* lässt sich durch die Formel (6.28) jedoch keine Gewichtung erreichen, die nicht auch mit den vorherig dargestellten Likelihood-Funktionen umsetzbar wäre (gegebenenfalls unter der Verwendung eines zusätzlichen Offsets). Denn für das Ergebnis ist die Relation der Gewichte der einzelnen Partikel zueinander entscheidend und nicht der absolute Betrag eines Gewichts.

Es besteht jedoch die Möglichkeit, zwischen den beiden Zuständen *empty* und *filled* zu differenzieren. Geht man davon aus, dass sich in Voxeln beider Zustände bei einer idealen Verarbeitungskette kein gesuchtes Objekt befinden kann, so gibt es keinen Grund, weshalb einer dieser Zustände ein höheres Gewicht erhalten sollte als der andere. Berücksichtigt man jedoch, dass die reale Bildverarbeitung (insbesondere das Background Subtraction) vermutlich fehleranfälliger ist als die (einmalige) Modellierung der statischen Objekte, so kann unter den getroffenen Annahmen (Abschnitt 2.4) mit höherer Sicherheit ausgeschlossen werden, dass sich ein dynamisches Objekt in einem Voxel des Zustands *filled* befindet als in einem Voxel des Zustands *empty*.

Dementsprechend bestünde eine Möglichkeit darin, die Voxel des Zustands *empty* mit dem Faktor ϵ nach Formel (6.29) positiv zu gewichten, unter der Bedingung $0 < \epsilon < \psi < \upsilon \leq 1$, wodurch sich implizit für die verbleibenden Voxel des Zustands *filled* ein geringeres Gewicht (0) ergibt.

$$w_k^j = \frac{\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j| + \epsilon \cdot |\mathrm{E}_k^j|}{|M_k^j|} \tag{6.29}$$

Alternativ dazu könnten aber auch die Voxel des Zustands *filled* nach Formel (6.30) mit einem Faktor θ negativ gewichtet werden. Damit erhalten die verbleibenden Voxel des Zustands *empty* implizit ein höheres Gewicht (0).

$$w_k^j = \frac{\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j| - \theta \cdot |\Theta_k^j|}{|M_k^j|} \tag{6.30}$$

Problematisch an dieser Gewichtung ist allerdings, dass die Partikelgewichte durch den negativen Bestrafungsterm auch kleiner 0 werden können. Um ein Partikelgewicht zwischen [0,1] zu erzeugen, müsste neben einer Normierung auch ein Verschiebung negativer Gewichte in den positiven Bereich vorgenommen werden, sofern man negative Gewichte nicht einfach auf 0 setzen möchte. Bei einer Verschiebung sämtlicher Partikelgewichte mit einem Offset, der dem Betrag des kleinsten negativen Partikelgewichts entspricht, werden die bereits positiven Gewichte zu größeren Werten hin verschoben. Je weiter diese allerdings vom Nullpunkt entfernt liegen, umso geringer fallen die Unterschiede dieser Gewichte bei anschließender Normierung aus. Die Partikel des Partikelfilters approximieren damit weniger die gewünschte Wahrscheinlichkeitsverteilung. Vielmehr streuen sie breiter um den zu schätzenden Zustand, da ihre Gewichte durch die Offset-Verschiebung zu ähnlich werden.

Abhilfe verschafft hierbei eine Bestrafung der *filled*-Voxel, die multiplikativ mit dem additiven Term verknüpft wird. Solch eine Gewichtung wird in Formel 6.31 mithilfe einer Polynomfunktion erreicht. In Formel 6.32 ist zusätzlich die nichtlineare Verstärkung aus Abb. 6.6(b) Teil der Funktion. Die multiplikative Verknüpfung des Bestrafungsterms kann als „Verundung“ interpretiert werden. Ein hohes Gesamtgewicht des additiven Terms kann nur dann in Gänze zu einem hohen Gewicht führen, wenn der bestrafende Term ebenfalls einen hohen Teilwert liefert. Dies ist dann der Fall, wenn sich keine oder nur wenige Voxel des Zustands *filled* innerhalb des Ellipsoids befinden. Mit zunehmender Anzahl an *filled*-Voxeln wird ein Ellipsoid stärker bestraft und kann damit auch ein Gewicht von 0 hervorrufen. Wie stark die Bestrafung dabei tatsächlich ausfällt, wird über den Exponenten a geregelt.

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{\left(\frac{|\Phi_k^j|}{|M_k^j|} - s\right)^a}{(-s)^a} \quad \text{mit} \quad s = 1,\ 1 \leq a,\ a \in \mathbb{N} \tag{6.31}$$

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{e^{\frac{|\Upsilon_k^j|}{|M_k^j|}}}{e} \cdot \frac{\left(\frac{|\Phi_k^j|}{|M_k^j|} - s\right)^a}{(-s)^a} \quad \text{mit} \quad s = 1,\ 1 \leq a,\ a \in \mathbb{N} \tag{6.32}$$

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{e^{\frac{|\Upsilon_k^j|}{|M_k^j|}}}{e} \cdot \frac{\left(\frac{|\Phi_k^j|}{|M_k^j|} - s\right)^a}{(-s)^a} \cdot \frac{\left(\frac{|\mathrm{E}_k^j|}{|M_k^j|} - s\right)^b}{(-s)^b} \tag{6.33}$$
$$\text{mit} \quad s = 1,\ 1 \leq a,\ a \in \mathbb{N},\ 1 \leq b,\ b \in \mathbb{N}$$

Auf die gleiche Art und Weise kann letztendlich auch die Menge der *empty*-Voxel bestraft werden. Deshalb wird in Formel 6.33 die Funktion noch um einen weiteren Term erweitert. Mit dem Steuerparameter b kann dabei die Stärke der Bestrafung der *empty*-Voxel festgelegt werden. Aus den semantischen Vorüberlegungen könnte die Bedingung $b \leq a$ noch in die Formel mit aufgenommen werden. In den Experimenten des Kapitels 7 werden jedoch auch Parametrisierungen untersucht, für welche diese Bedingung nicht gilt.

6.4.3 Einfluss der Größe eines Verdeckungsvolumens

Wie die Bewegung eines Filters in oder an Verdeckungen konkret ausfällt, hängt nicht nur von der Gewichtungsfunktion und der Objektbewegung selbst ab, sondern auch von der Anordnung der Verdeckungsvolumina im Raum, ihrer Form sowie ihrem Volumen.

In Abbildung 6.7 wird der Einfluss der Verdeckungsgröße wieder an einem zweidimensionalen Beispiel erläutert. Die Gewichtsfaktoren für die Verdeckung (Rechteck) und die Messung (Ground-Truth-Ellipse) sind gleich groß ($\psi = \upsilon$), was an dem einheitlichen Blauton zu erkennen ist. Bei den drei dargestellten Verdeckungssituationen sind jeweils neun Ellipsenhypothesen eingezeichnet, wobei sich die Ground-Truth-Ellipse wieder in der Mitte befindet. Die Verdeckung von Abb. 6.7(a) ist größer als die von (b). Vergleicht man die berechneten Schwerpunktellipsen (gelbe Kreise, vgl. Abb. 6.3), so liegt erstere etwas weiter in der Verdeckung. Aufgrund der Größe dieser Verdeckung erhalten die rechte obere und untere Ellipse ein größeres Gewicht und beeinflussen entsprechend die

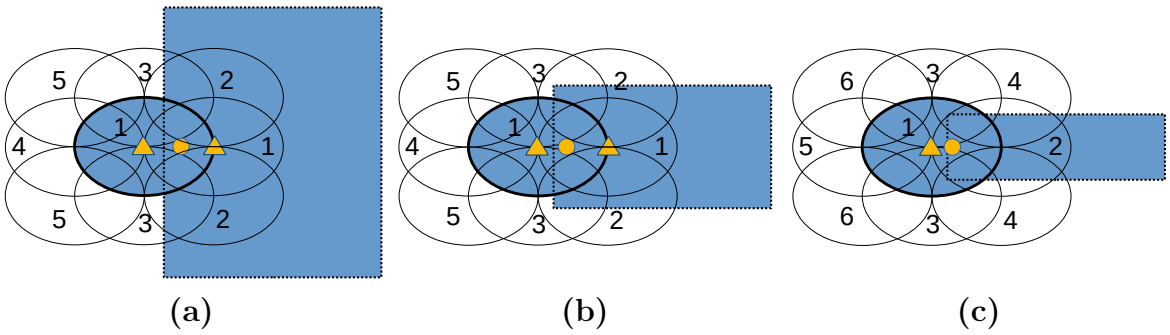


Abb. 6.7: Beispiel für den Einfluss der Größe eines Verdeckungsvolumens auf die Gewichtung der Ellipsenhypothesen, bei gleicher Gewichtung von Verdeckung (Rechteck) und Messung (blaue Ellipse). Je kleiner die Verdeckung ist, betrachtet von (a) nach (c), desto geringer sind die Abweichungen zwischen den Schätzungen (Ellipsen deren Zentren durch die gelben Symbole repräsentiert werden) und der Ground-Truth-Ellipse (Ellipse mit stärkerem schwarzen Rand).

Filterschätzung sowie das Resampling, welches im nachfolgenden Schritt durchgeführt wird.

Für die Filterbewegung würde dies im Effekt bedeuten, dass sich der Filter bei Abb. 6.7(a) schneller in die Verdeckung bewegt als bei (b), da mehr Partikel die Verdeckung in größerem Ausmaß schneiden als die Messung und damit diese Partikel häufiger resampelt werden, aufgrund eines höheren Gewichts. Auf der rechten Seite von Abb. 6.7(c) ist die Verdeckung so klein, dass eine Ellipsenhypothese nicht vollständig in die Verdeckung passt. Deshalb liefert die Verdeckung einen geringeren Beitrag zu den Gewichten der Ellipsenhypothesen. Im Ergebnis existiert bei (c) gegenüber (a) und (b) nur eine Ellipse maximalen Gewichts und die Schwerpunktellipse befindet sich näher an der Ground-Truth-Ellipse. Das gewählte Beispiel soll zeigen, dass Verdeckungen bei konstanter Gewichtung vermutlich umso stärker die Filterbewegung beeinflussen, je größer sie sind. Dies sollte bei der Parametrisierung beachtet werden. Verdeckungsbereiche bzw. verdeckte Volumina, die deutlich kleiner sind als das Objekt selbst, stellen generell ein kleineres Problem dar, da dann ein Teil des Objekts immer detektiert werden kann (sofern die Objektdetektion gelingt) und das Objekt für den Filter damit typischerweise nicht vollständig verloren geht.

6.4.4 Einfluss der Form eines Verdeckungsvolumens

Neben der Verdeckungsgröße ist auch die Form der Verdeckung entscheidend für das Filterverhalten. Ein Beispiel dazu wird in Abb. 6.8 veranschaulicht. Hierbei bewegt sich ein ellipsoidförmiges Objekt durch eine Verdeckung. Eingenommene Positionen sind durch die Ellipsen mit stärkerem schwarzen Rand (A, B und C) gekennzeichnet.

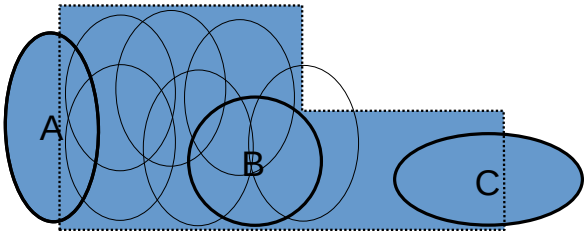


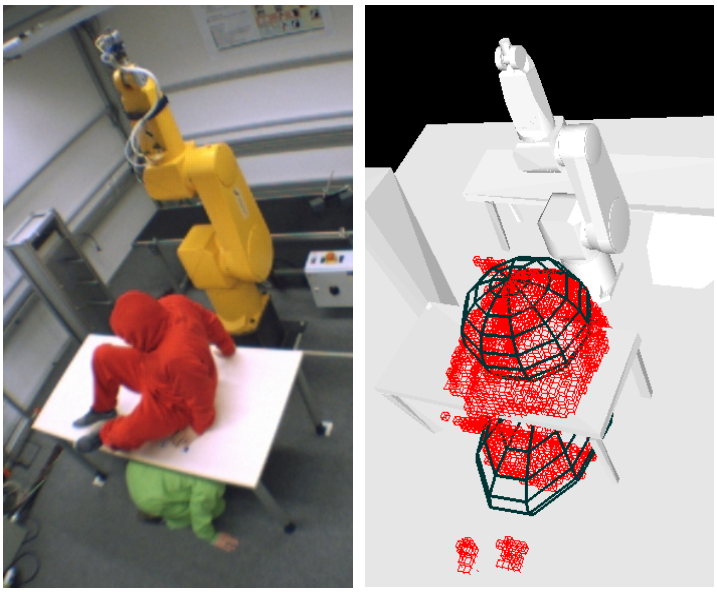
Abb. 6.8: Beispiel für den Einfluss der Form einer Verdeckung auf die Filterbewegung. Ein Objekt (Ellipse mit stärkerem schwarzen Rand) bewegt sich von Position A zur Position C. Aufgrund der Form der Verdeckung werden Partikel im linken Bereich hoch gewichtet und beim Resampling verstärkt, weshalb die Diffusion durch den Zustandsraum und damit durch die Verdeckung möglicherweise nicht ausreicht, um dem Objekt (schnell genug) bis zur Position C zu folgen. Das Objekt könnte sich dabei aus der Verdeckung entfernen, während der Filter darin verbleibt.

Aufgrund der Form der Verdeckung kann es passieren, dass die Streuung der Ellipsen (repräsentiert von den restlichen Ellipsen) nicht ausreicht, um von der aufrechten Position A schnell genug in die liegende Position C überzugehen. Dadurch könnte sich das Objekt bereits aus der Verdeckung herausbewegt haben, während sich der Filter immer noch im linken Bereich der Verdeckung befindet, wo besonders viele Partikel ein hohes Gewicht erhalten und stärker resampelt werden, so dass diese nicht in einen anderen Bereich des Zustandsraums diffundieren. In einer konkreten Anwendung des Verfahrens muss sich die Verdeckungsform nicht problematisch auf das Tracking auswirken, dennoch sollte der mögliche Einfluss berücksichtigt werden, wenn der Filter nicht das gewünschte Verhalten zeigt.

6.5 Partikelprädiktion mit Kollisionstest

Jedes Voxel des Zustands *filled* stellt eine räumliche Belegung und damit eine Barriere dar, die nicht von anderen Objekten durchdrungen werden kann. In den Formeln (6.29) bis (6.33) werden *filled*-Voxel innerhalb der Partikelellipsoide bei der Likelihood-Gewichtung bestraft, da sich eine Person nicht gleichzeitig mit einem statischen Objekt eine Raumbelegung teilen kann.

Abhängig von der gewählten Likelihood-Funktion und deren Parametrisierung könnte diese Bestrafung so aussehen, dass stringent Nullgewichte für sämtliche Partikel, deren Ellipsoide *filled*-Voxel enthalten, vergeben werden. Da ein Ellipsoidmodell menschliche Formen nur sehr grob approximieren kann, muss eine gewisse Toleranz gegenüber *filled*-Voxeln gewährleistet werden, um Tracking-Verluste in bestimmten Situationen zu vermeiden. Näheres dazu wird in Kapitel 7 beschrieben. Mit einer milderen Bestrafung der *filled*-Voxel geht jedoch einher, dass sich ein Filter prinzipiell auch durch statische



(a) **(b)**

Abb. 6.9: Situation eines Zweipersonen-Trackings (a) und die zugehörige Filterschätzungen (b). Die rekonstruierten *unknown*-Voxel der Beobachtung werden gezeigt (rot) sowie die gewichteten Schwerpunktellipsoide der Zustandsschätzungen (schwarz).

Objekte hindurch bewegen kann. Abb. 6.9 zeigt eine Tracking-Situation, in der sich eine Person auf dem Tisch befindet und eine andere darunter. Das Verdeckungsvolumen unter dem Tisch wird weniger stark gewichtet als die Messung über dem Tisch, weshalb sich der Filter von der grünen Person auf die rote Person durch den Tisch hindurch bewegen kann. Der Einsatz einer Blocking-Methode kann dies zwar weitestgehend verhindern, dennoch könnten auch Artefakte oder leere Verdeckungsvolumina, die in der Nähe statischer Objekte entstehen, einen ähnlichen Effekt herbeiführen. Wenn dann kein anderer blockierender Filter in der Nähe ist, so übt die Blocking-Methode in diesen Fällen keinen oder nur einen geringen Effekt aus. Deshalb könnte es nützlich sein, die Undurchdringlichkeit realer physischer Barrieren auf das Tracking-System abzubilden, um die Filterbewegung im Zustandsraum entsprechend einzuschränken. Dazu steht Wissen in Form der modellierten 3D-Objekte zur Verfügung, das hierfür genutzt werden kann.

Eine mögliche Umsetzung zur Einschränkung der Filterbewegungen stellt der wie folgt beschriebene Kollisionstest dar. Bei diesem wird im Schritt der Partikelprädiktion (Anwendung des Bewegungsmodells) überprüft, ob ein neuer Partikelzustand eine Kollision mit einem statischen Objekt verursacht. Hierbei wäre eine stringente Konsequenz (analog zu einer Nullgewichtung der Ellipsoide mit *filled*-Voxeln) das Ersetzen sämtlicher Partikel, deren Ellipsoide einen Schnitt mit den Modellen der statischen Objekte verursachen. Das Ergebnis wäre jedoch wieder ein vorzeitiges Terminieren des Trackers in

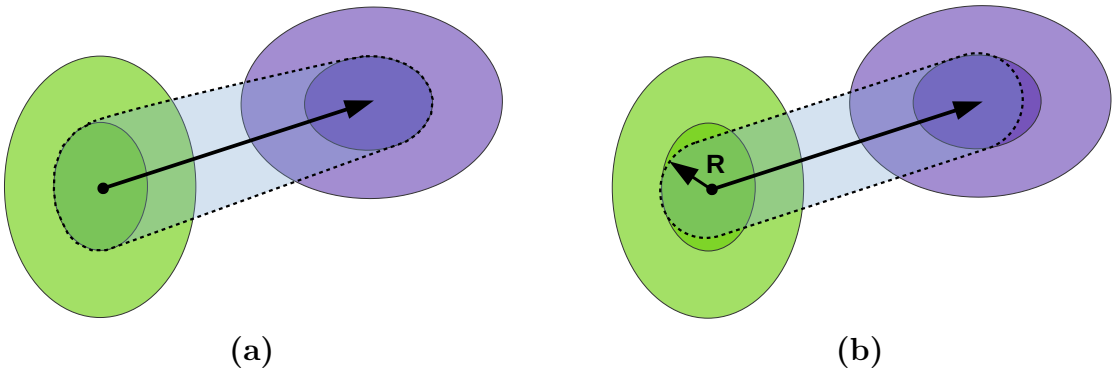


Abb. 6.10: Das dargestellte Sweeping-Volumen wird bei der Partikelprädiktion auf Kollision mit den statischen Objekten des Überwachungsraums hin getestet. Zu sehen ist in (a) ein Sweeping-Volumen (gepunktet), das von den Ellipsoid-Kernen (etwas dunkler eingefärbt) eines Partikels im prädizierten Zustand $x_k^{(j)}$ (violett) und im vorangegangenen Zustand $x_{k-1}^{(j)}$ nach dem Resampling (grün) aufgespannt wird. Aus Effizienzgründen erfolgt eine Approximation des Sweeping-Volumens durch eine Kapsel mit dem Radius R, dargestellt in (b).

bestimmten Situationen, beispielsweise wenn sich die Person unter den Tisch begibt und die Tischbeine zu einer Kollision mit den Partikelellipsoiden führen.

Zur Verhinderung solch eines Filterverhaltens wird ein Kollisionstest benötigt, der nachgiebiger ist, aber trotzdem vollständige „Filterdrifts“ durch die statischen Objekte verhindert. Hierfür wird in dieser Dissertation folgender Ansatz untersucht: Ein Sweeping-Volumen wird aufgespannt von den Ellipsoidkernen eines Partikels j im Zustand $x_{k-1}^{(j)}$ nach dem Resampling sowie seinem vorhergesagten Zustand $x_k^{(j)}$, dargestellt in Abb. 6.10(a). Diese Kerne sind kleine Versionen ihrer Eltern mit den gleichen Proportionen und dem selben Zentrum. Kommt es zu einer Kollision des Sweeping-Volumens mit einem statischen Objekt, so wird der betroffene Partikel verworfen und durch einen kollisionsfreien Partikel nach einer lokalen Suche ersetzt. Zur Reduktion des Zeitaufwands solch eines Kollisionstests wird das ideale Sweeping-Volumen durch eine Kapsel approximiert, genannt „R-Zylinder“, die den Radius R besitzt und an den Ellipsoidzentren ausgerichtet ist, wie in Abb. 6.10(b) gezeigt. Der Radius R sollte kleiner sein als die minimale Länge der sechs Achsen der beiden Elternellipsoide. Aus Effizienzgründen werden die statischen Hindernisse ebenso als Kapseln und andere geeignete Primitive modelliert. Der Einfluss des Kollisionstests wird in den Experimenten des Kapitels 7 untersucht.

6.6 Zusammenfassung

In diesem Kapitel wurde der Fokus auf eine tracking-basierte Zustandsschätzung zur Personenlokalisierung im 3D-Raum gelegt, wobei statische Verdeckungsvolumina und

Barrieren im Überwachungsraum im Besonderen betrachtet wurden. Nach einer Beschreibung des abstrakten Bayes-Filters zur rekursiven Schätzung einer Aposteriori-Dichte wurde auf das Prinzip des Sequential Importance Samplings (SIS) eingegangen, welches durch eine spezielle Umformung des Importance Samplings (IS) erreicht werden kann. Mit dem SIS erhält man einen generellen rekursiven Monte-Carlo-Filter als diskrete Approximation des Bayes-Filters, der für Online-Anwendungen geeignet ist und aus dem der Partikelfilter hervorgeht. Für diese Dissertation wurde als Partikelfilter konkret der Bootstrap-Filter implementiert. Dieser nimmt zusätzlich zum SIS ein Resampling im Update-Schritt vor, um eine Partikeldegenerierung zu vermeiden, was auch als Sequential Importance Resampling (SIR) bezeichnet wird. Dabei wird im Anschluss an eine Likelihood-Gewichtung aus den Partikeln eine neue Partikelmenge erzeugt, um möglichst viele Partikel auf den Zustandsbereichen hoher Wahrscheinlichkeit zu konzentrieren, damit eine gute Approximation der Aposteriori-Dichte erreicht werden kann. Der implementierte Algorithmus aus [Choset et al., 2006] bietet den Vorteil, dass der gesamte Resampling-Prozess in $O(N)$ Schritten ausgeführt werden kann.

Für das Tracking wird als Objektmodell ein triaxiales Ellipsoid verwendet, für dessen Parametrisierung als Randbedingung ein konstantes Ellipsoidvolumen angenommen wird. Der neundimensionale Zustandsvektor wird durch die am Voxelraum ausgerichteten Ellipsoidachsen, die 3D-Position des Ellipsoidzentrums sowie die Positionsgeschwindigkeit beschrieben. Da die für eine gleichbleibende Ortsauflösung/Genauigkeit benötigte Partikelanzahl exponentiell mit der Anzahl an Dimensionen steigt, sind wenige Dimensionen des Zustandsraums und die damit einhergehende geringere Partikelanzahl vorteilhaft für das Tracking mit dem Partikelfilter. Nachteilig hierbei ist jedoch, dass ein Erfassen detaillierter Bewegungsabläufe, wie es mit höherdimensionalen Objektmodellen wie z. B. Skelettmodellen möglich ist, nicht erfolgen kann. Das gewählte Objekt- und Bewegungsmodell dient vorwiegend einer 3D-Personenlokalisierung.

Das Tracking von mehreren Personen erfolgt mit separaten Partikelfiltern. Zur Initialisierung wird zu jedem betrachteten Zeitpunkt k eine 3D-Segmentierung aller *unknown*-Voxel vorgenommen. Die Voxelsegmente werden anschließend nach der Größe ihrer Volumina sortiert. Zu allen Voxelsegmenten, die eine minimale Größe überschreiten, werden deren Euklidische Abstände zu den gewichteten Schwerpunktellipsoiden bereits existierender Filter berechnet. Sind alle Abstände für solch ein betrachtetes Segment ausreichend groß, so wird davon ausgegangen, dass noch keine Filterassoziation existiert, was zu einer Initialisierung eines neuen Filters führt. Unabhängig davon, ob ein neuer Filter benötigt wird, muss mit jeder neuen Beobachtung auch eine Zuordnung der Einzelmessungen (beispielsweise in Form von Segmenten oder Voxeln) zu den bestehenden Filtern vorgenommen werden. Dieses Datenassoziationsproblem wird mit einer Blocking-Methode gelöst, die innerhalb der Likelihood-Funktion Anwendung

findet. Sie soll vermeiden, dass ein Filter, der bereits ein Objekt trackt durch einen anderen Filter verdrängt wird. Eine Filterterminierung erfolgt, sobald das maximale Partikelgewicht eines Filters unter einem definierten Schwellenwert liegt, oder alternativ, wenn das Ellipsoid maximalen Gewichts eine zu geringe Menge an *occluded*- und *unknown*-Voxeln enthält. In diesen Fällen wird davon ausgegangen, dass die Person den Überwachungsraum verlassen oder der Filter die Person verloren hat.

Die Likelihood-Funktion dient der Bewertung wie gut die aufgetretene Messung z_k einen propagierten Zustand $x_k^{(j)}$ erklärt und liefert hierfür ein entsprechendes Gewicht. Die Wahl der extrahierten Merkmale sowie deren Verarbeitung in der Likelihood-Funktion sind entscheidend für die Güte der Partikelgewichtung, um eine entsprechend gute Approximation der Aposteriori-Dichte zu erhalten. In dieser Dissertation wird die Volumeneinpassung der Partikelellipsoide in die voxelbasierten Rekonstruktionsdaten bewertet. Dabei erhalten auch die verdeckten *occluded*-Voxel, welche sensorisch nicht erfasst werden können, eine Gewichtung. Anhand zweidimensionaler Beispiele wurde dargelegt, welchen Einfluss diese Vorgehensweise auf das Filterverhalten sowie die Schätzergebnisse ausübt. Ermöglicht wird damit die Objektverfolgung durch die gegebenen Verdeckungsvolumina hindurch. Damit ein Filter einem dynamischen Objekt auch dann aus einem Verdeckungsvolumen heraus folgen kann, wenn dieses zu einer vollständigen Objektverdeckung führt, müssen die *occluded*-Voxel geringer gewichtet werden als die *unknown*-Voxel. Für die Likelihood-Funktion wird weiterhin eine nichtlineare Verstärkung der *unknown*-Voxel mit zunehmender Häufigkeit vorgeschlagen, damit diese gegenüber der *occluded*-Voxel stärker präferiert werden, wenn sie in höherer Anzahl vorliegen. Weiterhin können Terme zur Bestrafung der *empty*- und *filled*-Voxel in die Likelihood-Funktion integriert werden, um Filterbewegungen auf statische Objekte oder leere Volumina stärker einzuschränken.

Jedes Voxel des Zustands *filled* stellt eine räumliche Belegung und damit eine Barriere dar, die nicht von anderen Objekten durchdrungen werden kann. Die Bestrafung dieser Voxel innerhalb der Likelihood-Funktion soll ein häufiges Resampeln von Ellipsoiden, die in statische Objekte hineinragen, verhindern, indem sie ein geringes Gewicht erhalten. Alternativ oder zusätzlich kann ein Kollisionstest durchgeführt werden, der Partikel im Prädiktionsschritt auf Kollisionen mit den gegebenen statischen Objekten testet und diese gegebenenfalls ersetzt. Der implementierte Kollisionstest verwendet R-Zylinder zur Approximation des Testvolumens und toleriert Kollisionen bis zu einem gewissen Grad. Der Vorteil dieses Kollisionstests gegenüber der Bestrafung von *filled*-Voxeln ist, dass ein vollständiges Durchdringen von Filtern durch statische Objekte verhindert werden kann, ohne Partikel, die ein Stück weit in statische Objekte hineinragen, zu bestrafen. Ob das erwartete beschriebene Filterverhalten tatsächlich erreicht wird, ist Gegenstand

der experimentellen Untersuchungen des nachfolgenden Kapitels 7. Weiterhin werden die vorgeschlagenen Varianten der Likelihood-Funktion dieses Kapitels untersucht.

Kapitel 7

Experimente

In diesem Kapitel wird untersucht, welchen Einfluss die Integration zusätzlichen Wissens in den Tracking-Prozess auf die Tracking-Ergebnisse ausübt. Das Wissen bezieht sich auf die bekannten statischen Objekte des Überwachungsraums mit ihren Verdeckungsvolumina und wird in Form verschiedener Voxelzustände sowie von 3D-Modellen zur Verfügung gestellt, die zu den statischen Objekten erzeugt wurden.

Zu Beginn werden in den Abschnitten 7.1 und 7.2 Details zur Implementierung und zur Durchführung der Experimente erläutert. In den Abschnitten 7.3 bis 7.5 werden Experimente zur Likelihood-Gewichtung der vier Voxelzustände *empty* (Parameter $\epsilon \in \mathbb{R}$), *filled* (Parameter $\phi \in \mathbb{R}$), *unknown* (Parameter $\upsilon \in \mathbb{R}$) und *occluded* (Parameter $\psi \in \mathbb{R}$) durchgeführt. Eine Gesamtevaluierung des Tracking-Ansatzes erfolgt in Abschnitt 7.6. Hierbei werden verschiedenste Parametrisierungen vergleichend betrachtet und in Zusammenhang mit den vorhergehenden Ergebnissen diskutiert und bewertet. Weiterhin wird auch der Kollisionstest mit den R-Zylindern aus Abschnitt 6.5 untersucht, welcher zum Ziel hat, die Diffusion von Partikelfiltern durch physische Barrieren zu unterbinden. In Abschnitt 7.7 wird das Kapitel zusammengefasst.

7.1 Implementierungsdetails

Die gewählte Voxelraumauflösung liegt bei $126 \times 110 \times 73$ Voxel (in Abb. 5.7 wird der Voxelraum gezeigt), wobei die Seitenlänge jedes Voxels ungefähr 3,4 cm entspricht. Die Auflösung der Kamerabilder beträgt über die gesamte Arbeitskette hinweg 640×480 Pixel. Die Bilder wurden mit einer Framerate von 30 fps (frames per second) aufgezeichnet. Für die Rekonstruktion der Visuellen Hülle wurden die Silhouettenbilder von sieben Kameras verwendet, die mit einem Background-Subtraction-Verfahren erzeugt wurden. Konkret kam dafür ein Codebook-Verfahren der OpenCV zum Einsatz [Pavlenko, 2012]. Die Verarbeitung erfolgte offline mit zuvor aufgezeichneten Videosequenzen. In allen Experimenten wurden die Partikelfilter mit je 500 Partikeln initialisiert, um eine Vergleichbarkeit der Ergebnisse sicherzustellen. In mehreren Tests erwies sich diese Anzahl bei Betrachtung des Verhaltens des gewichteten Schwerpunktellipsoids als geeignet. Die Partikelanzahl ist nicht adaptiv, jedoch kann sie beim Einsatz des Kollisionstests (vgl.

Abschnitt 6.5) in der Nähe statischer Verdeckungsvolumina aufgrund eines notwendigen Abbruchkriteriums bei der Partikelprädiktion temporär verringert sein. Führt im Falle der Kollision eines Partikels die lokale Suche nach einem kollisionsfreien Ersatzpartikel nach einer begrenzten Anzahl an Versuchen nicht zu einem positiven Ergebnis, so wird der Partikel für diesen Zeitschritt verworfen. Im nachfolgenden Resampling-Schritt wird die ursprüngliche Partikelanzahl wieder hergestellt.

Zur Initialisierung eines Partikelfilters wird der Voxelraum in zusammenhängende 3D-Komponenten segmentiert. Für alle Komponenten mit einer minimalen Voxelanzahl (2000 für den verwendeten Voxelraum) wird überprüft, ob bereits ein Partikelfilter für dieses Segment existiert (d. h. ob die aktuelle Schätzung das Segment schneidet). Falls nicht, wird ein neuer Filter auf dem Segment initialisiert, indem für jedes Partikel zufällig ein Voxel des Segments als Ellipsoidmittelpunkt ausgewählt wird. Die Längen der Ellipsoidachsen l und m werden gleichverteilt aus einem Bereich (in etwa 0,3 bis 0,6 m) gezogen und die Länge für n entsprechend des gewählten konstanten Ellipsoidvolumens als Nebenbedingung ermittelt. Die Geschwindigkeiten werden aus einer Normalverteilung mit Mittelwert 0 gezogen.

Für das verwendete Ellipsoidmodell wird als Bounding Box ein konstantes Volumen von $\frac{1}{3}\,\mathrm{m}^3$ angenommen. Weiterhin werden die Seitenlängen dieser Bounding Box auf Werte zwischen 0,2 m und 2 m begrenzt, was plausibel für zu trackende Personen erscheint. Das Volumen der Ellipsoide ergibt sich dann zu: $V_{\text{Ellipsoid}} = \frac{\pi}{6} \cdot l \cdot m \cdot n = \frac{\pi}{6} \cdot \frac{1}{3} = 0{,}174\,\mathrm{m}^3$, wobei l, m und n die Achsenlängen der Ellipsoide sowie der sie umschließenden Bounding Box sind. Dementsprechend umfasst das Ellipsoidvolumen ca. 175 l. Auch wenn das durchschnittliche Volumen eines 1,7 m großen Menschen grob als 75 l angenommen werden kann, so werden noch Zuschläge benötigt für die Berücksichtigung der Kleidung sowie von Ungenauigkeiten bezüglich der geometrischen Approximation, die bei Anwendung des Volumenverschnitts bei der Rekonstruktion der Visuellen Hülle entstehen. Ein deutlich geringeres Volumen als 175 l führte zu stärkeren Schwingungen des Schwerpunktellipsoids zwischen den Frames, weil verschiedenste Partikelellipsoide sich vollständig auf dem rekonstruierten Bereich platzieren konnten und dementsprechend zu ähnlich hohen Gewichten führten durch die sich kein stabiler Schwerpunkt der besten Schätzung herausbildete. Bei zu groß gewählten Ellipsoidvolumina erhielten ebenfalls zu viele Partikel ein ähnliches Gewicht, weil die Messung von verschiedenen Ellipsoiden vollständig eingeschlossen werden konnte. Dies ist ebenfalls zu vermeiden.

Für die Rauschkomponente des Bewegungsmodells (vgl. Abschnitt 6.3.1) werden univariate Normalverteilungen für die einzelnen Längen l und m in x- und y-Richtung verwendet. Die verbleibende Länge n in z-Richtung ist aufgrund der Randbedingung des konstanten Volumens von den l- und m-Längen abhängig. Dies ist zwar prinzi-

piell symmetrisch für jede Achse, aber die Werte von l und m werden als erstes aus univariaten Normalverteilungen gezogen.

Zur Berechnung der Zustandsvektoren wurden folgende Standardabweichungen in der Implementierung gewählt: 4 cm für die Geschwindigkeiten in alle drei Richtungen und 6 cm für die drei Achsenlängen der Ellipsoide. Die Positionsvariablen wurden nicht verrauscht. Die Geschwindigkeit wird als konstant angenommen. Änderungen werden durch das Rauschen modelliert. Für die Ellipsoidlängen gilt das gleiche. Zur Einordnung der genannten Werte sollte beachtet werden, dass diese sich auf die Varianz von Frame zu Frame beziehen und die Videosequenzen mit 30 fps aufgezeichnet wurden, sodass der Faktor 30 für die Betrachtung im Sekundenbereich relevant ist. Bei der Ausführung von Bewegungen mit größeren Geschwindigkeiten (beispielsweise Sprints) müssten die Werte entsprechend angepasst oder adaptiv gestaltet werden.

Das Personen-Tracking erfolgt in einem Überwachungsraum, der räumlich begrenzt ist. Die Grenzen müssen im Verfahren behandelt werden, was folgendermaßen umgesetzt wurde: Für außerhalb liegende Bereiche werden virtuelle Voxel angenommen, die ebenfalls einen Voxelzustand besitzen, damit Ellipsoide prinzipiell auch über die Grenzen des Voxelraums hinausragen können. Dies ermöglicht eine Voxelgewichtung auf die gleiche Art wie innerhalb des Voxelraums. Die Voxelzustände müssen anwendungsabhängig so gewählt werden, dass der Filter das gewünschte Verhalten zeigt, wenn er an den Rand des Überwachungsraums gelangt bzw. diesen überschreitet. Ein Filter soll beim vollständigen Verlassen des Überwachungsraums terminieren, d. h. wenn sich keine Partikel mehr innerhalb des Überwachungsraums befinden. Meist erfolgt jedoch bereits vorher eine Terminierung, wenn die Partikelgewichte innerhalb des Überwachungsraums zu gering gewichtet werden. Das Verlassen des Überwachungsraums ist bei der betrachteten Roboterarbeitszelle nur über eine der vier Seiten möglich, da sich an den anderen drei Seiten Wände befinden. Werden die virtuellen Voxel als *empty* angenommen und immer mit 0 gewichtet oder bestraft, so kann das gewünschte Filterverhalten implementiert werden, weil die Vervielfältigung solcher Partikel beim Resampling mindestens nach mehreren Zeitschritten aussetzt. Für die virtuellen Voxel müssen dabei keine Daten abgespeichert werden. In den Experimenten werden virtuelle Voxel in den Ergebnisdiagrammen gesondert als *outside*-Voxel aufgeführt und einzeln gezählt, auch wenn sie semantisch und rechnerisch wie *empty*-Voxel behandelt werden.

Der Rechner für die Experimente hatte die folgenden Eckdaten: CPU AMD A10-7850K, 16 GB RAM, Ubuntu 16.04. In dieser Umgebung hatte der Partikelfilter mit 500 Partikeln und der gewählten Voxelraumauflösung von $126 \times 110 \times 73$ Voxel für einen Filterzyklus (Prädiktion, Update, Resampling) eine durchschnittliche Laufzeit von ca. 40 ms (ohne Kollisionen). Wurde der Kollisionstest aktiviert, so ergab sich im Mittel eine

Laufzeit von ca. 100 ms. Eine Parallelisierung wurde dabei nicht vorgenommen, wodurch noch Optimierungspotential sowie eine Perspektive der Echtzeitfähigkeit besteht.

7.2 Experimentspezifische Details

Die Partikelzustände wurden unter Einsatz eines deterministischen Zufallsgenerators gewürfelt. Ein solcher besitzt die Eigenschaft, die gleiche Folge von Pseudozufallszahlen zu liefern, wenn er mit dem gleichen Startwert, genannt **Seed**, initialisiert wird. Dies wurde sich zunutze gemacht, um eine direkte Vergleichbarkeit der Experimente mit verschiedenen Parametrisierungen zu ermöglichen, indem der Zufallsgenerator zu Beginn jedes Experiments mit demselben Startwert initialisiert wurde. Die Durchführung jedes Einzelexperiments erfolgte insgesamt viermal mit vier konstanten, aber unterschiedlichen Seeds. Damit sollten zufällig auftretende Effekte (Ausreißer) erkannt werden. Auch die Eingabedaten aus den Vorverarbeitungsschritten der Bildverarbeitung und 3D-Rekonstruktion, die dem Tracking zugrunde gelegt werden, waren für die Experimente identisch, um die Vergleichbarkeit zu garantieren.

Zur Lösung des Datenassoziationsproblems, das beim Tracking mehrerer Personen entsteht, wurde eine Blocking-Methode ähnlich zu [Canton-Ferrer et al., 2011] eingesetzt (vgl. Formel (6.17)).

Für die Experimente wurden zwei längere Videosequenzen generiert, in denen sich eine Person bzw. zwei Personen im Überwachungsraum bewegen. Diese wurden jeweils in Teilsequenzen unterteilt, für die das Tracking separat ausgewertet wird. Zwei ausgewählte Teilsequenzen A und B wurden für detaillierte Analysen verwendet. Zur Gesamtevaluierung wurden auch die anderen Teilsequenzen untersucht. Da in dieser Dissertation Verdeckungen im Fokus stehen, wurden vier Verdeckungsvolumina (vier Mengen bestehend aus *occluded*-Voxeln) betrachtet, die zu unterschiedlichen Objektverdeckungen der Personen führen. In Abb. 7.1(a) ist das reale Verdeckungsvolumen unterhalb des Tischs dargestellt. Dieses wurde für bestimmte Experimente um die grünen Voxel in (b) oder (c) synthetisch erweitert. In (d) ist ein hinzugefügtes Verdeckungsvolumen an anderer Stelle im Raum zu sehen. Durch (a) und (b) kommt es in den betrachteten Sequenzen zu partiellen Objektverdeckungen der Personen. Für (c) und (d) ergeben sich vollständige Objektverdeckungen. Die grünen Voxel wurden dabei ungeachtet ihres ursprünglichen Voxelzustands auf *occluded* gesetzt und damit als verdeckt behandelt.

Zur Bewertung der Tracking-Parametrisierungen wurde je Frame die beste Zustandsschätzung herangezogen. Diese wird von dem Schwerpunktellipsoid repräsentiert, das aus den gewichteten Ellipsoiden aller Partikel zu einem Zeitpunkt berechnet wird.

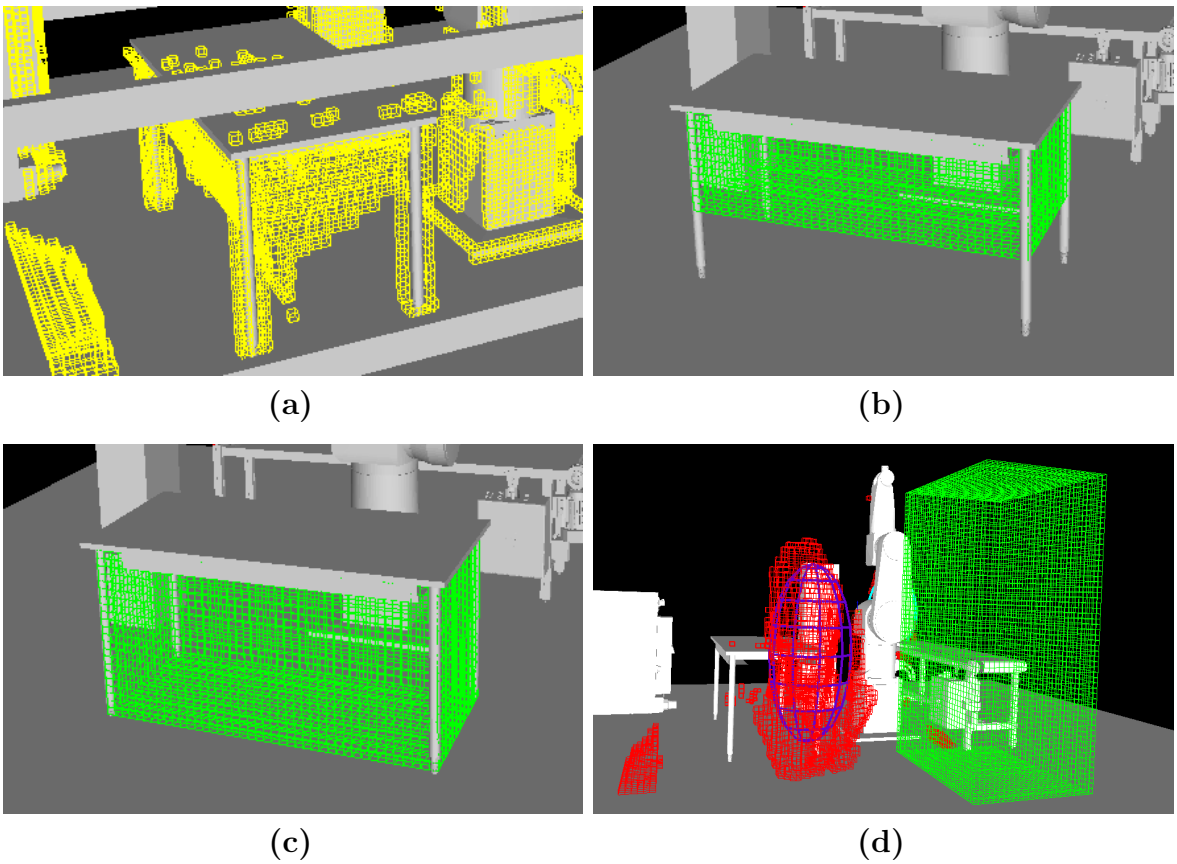


Abb. 7.1: Reale und synthetische Verdeckungsvolumina in der Roboterarbeitszelle. Gelbe *occluded*-Voxel visualisieren das reale Verdeckungsvolumen unterhalb des Tischs (a). Grüne Voxel zeigen synthetisch hinzugefügte *occluded*-Voxel in (b),(c) und (d). Durch die Verdeckungsvolumina in (a) und (b) kommt es in den betrachteten Videosequenzen zu partiellen Objektverdeckungen der Personen. Die Verdeckungsvolumina in (c) und (d) führen zu vollständigen Objektverdeckungen. Rote *unknown*-Voxel zeigen die Visuelle Hülle.

Die Verwendung des Ellipsoids maximalen Gewichts zeigte in Voruntersuchungen starke Schwankungen zwischen den Frames. Die Schwerpunktellipsoide waren hierfür stabiler, auch wenn sie eine kleine Latenz bei der Bewegungsverfolgung mit sich bringen können. Zur Bewertung des Filterverhaltens standen keine Ground-Truth-Daten zur Verfügung, da es insbesondere für Realweltszenarios mit Verdeckungsvolumina schwierig ist, diese in Form von 3D-Daten zu generieren. Stattdessen wurden die Häufigkeiten aller Voxelzustände, die innerhalb der Schwerpunktellipsoide liegen, für jeden Frame erfasst und grafisch in Diagrammen über alle Frames einer Teilsequenz dargestellt. Die Ergebnisse aller betrachteten vier Seeds wurden jeweils in das gleiche Diagramm eingetragen (vgl. Abb. 7.3). Weiterhin wurden Screenshots der im 3D-Viewer visualisierten Rekonstruktionsdaten und Schwerpunktellipsoide für jeden Frame abgespeichert und subjektiv ausgewertet. Zum Teil wurden auch die Ellipsoide der einzelnen Partikel abgebildet, um die Streuung des Filters zu zeigen, hierbei jedoch nur jedes 25. Ellipsoid aus Gründen der Übersichtlichkeit.

7.3 Gewichtung von Verdeckungsvolumina in der Likelihood-Funktion

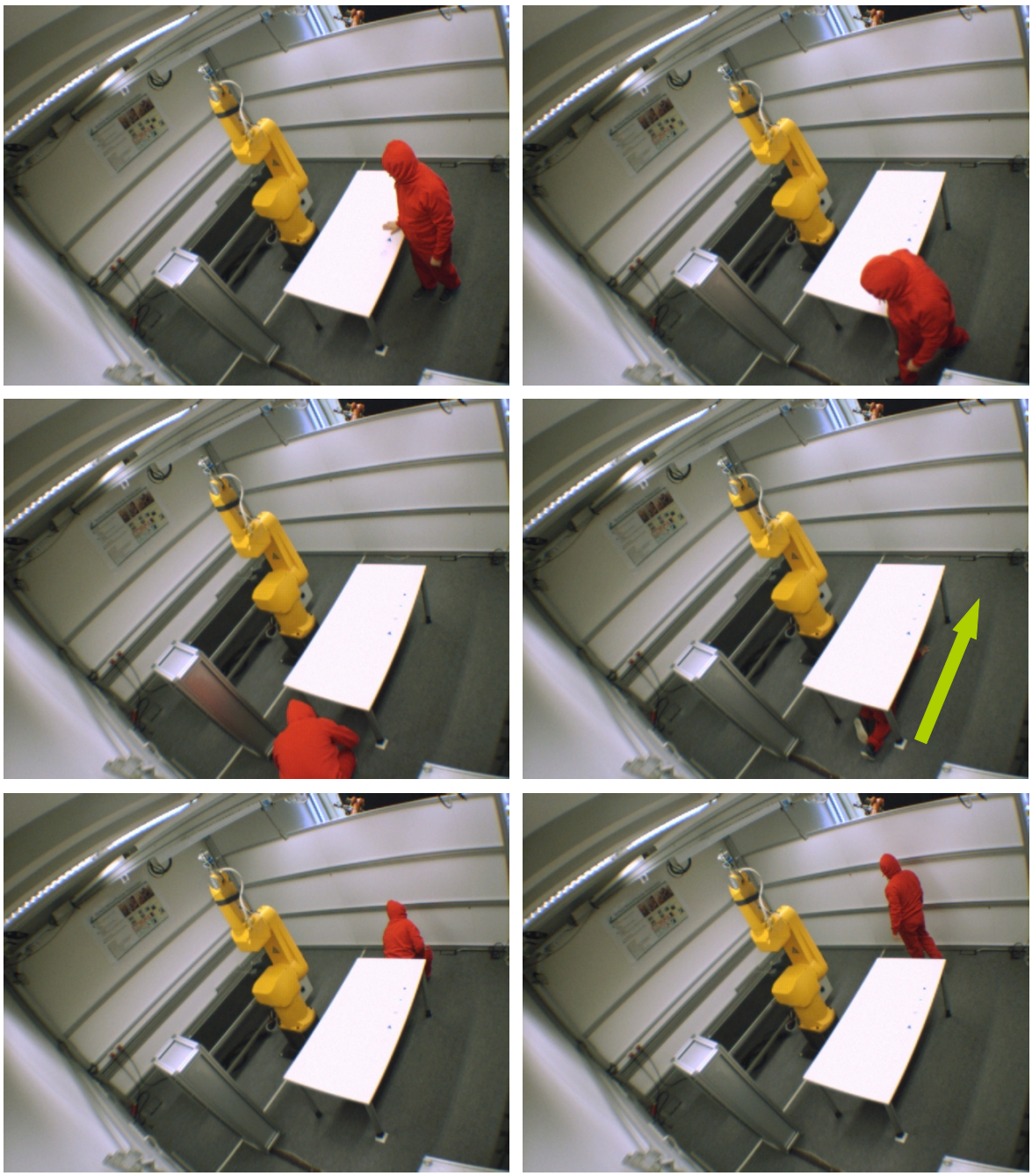


Abb. 7.2: Bilder der Teilsequenz A (Frame 1500 bis Frame 1699), die zur Bewertung der Gewichtung von *occluded*-Voxeln eingesetzt wird. Eine Person bewegt sich unter dem Tisch entlang und erfährt dabei eine partielle Objektverdeckung. Gezeigt werden die Frames 1510, 1536, 1555, 1583, 1629 und 1648 (von links oben bis rechts unten).

Die gegebene Voxelmenge Ψ aller *occluded*-Voxel repräsentiert die Verdeckungsvolumina der statischen Objekte des Überwachungsraums. Ihnen wird für die Likelihood-Gewichtung jeweils ein Gewicht ψ zugewiesen, das im Folgenden untersucht werden

soll. Dazu erfolgt die Likelihood-Gewichtung jedes Partikels j zum Zeitpunkt k nach Formel 7.1 entsprechend der gegebenen Voxelzustände innerhalb des Partikelellipsoids.

$$w_k^j = \frac{\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|}{|M_k^j|} \tag{7.1}$$

Die Werte von ψ werden in Zehntelschritten im Bereich von $[0, 1]$ variiert. Die Gewichte der anderen drei Parameter werden dabei konstant gehalten: $\epsilon = 0$ (*empty*-Voxel der Menge E), $\phi = 0$ (*filled*-Voxel der Menge Φ) und $\upsilon = 1$ (*unknown*-Voxel der Menge Υ). Demnach werden leere Voxel sowie Voxel, die statischen Objekten zugeordnet sind, nicht gewichtet. Die *unknown*-Voxel, für die angenommen werden muss, dass sie zu Personen gehören, erhalten in sämtlichen Experimenten ein Gewicht von 1.

Für die Untersuchungen dieses Abschnitts wird die Teilsequenz A (Abb. 7.2) eingesetzt. Die Person tritt darin zunächst frontal an den Tisch heran, läuft anschließend ein Stück um diesen herum und begibt sich danach unter den Tisch, wo sie eine Objektverdeckung erfährt. Auf der gegenüberliegenden Seite des Tischs kommt die Person wieder zum Vorschein, richtet sich auf und betritt den hinteren Bereich der Roboterarbeitszelle.

7.3.1 Partielle Objektverdeckung

Als Erstes wird das Verdeckungsvolumen aus Abb. 7.1(a) betrachtet, das zu einer partiellen Objektverdeckung der Person führt, wenn sich diese unter den Tisch begibt. In Abb. 7.3(a) ist das Ergebnis der Parametrisierung mit $\psi = \epsilon = \phi = 0$ zu sehen. Die *occluded*-Voxel werden dabei nicht gewichtet, wie in [Canton-Ferrer et al., 2011]. Gibt man den *occluded*-Voxeln ein Gewicht größer null, so werden die verdeckten Voxel gegenüber den *empty*- und *filled*-Voxeln von dem Partikelfilter „bevorzugt“, was erwünscht ist, da sich in diesen Voxeln Teile von Personen befinden könnten. Im Vergleich zu Abb. 7.3(a) lässt sich in (b) bis (k) im Mittel eine Erhöhung der Anzahl der *occluded*-Voxel in den Schwerpunktellipsoiden mit Zunahme des Werts von ψ erkennen. Gleichzeitig verringern sich die Häufigkeiten der *filled*- und *empty*-Voxel. Dies bildet folgenden gewünschten Effekt ab: Unter der Annahme, dass sich in den *filled*- und *empty*-Voxeln keine Person befinden kann, soll das gewichtete Schwerpunktellipsoid auch möglichst wenige dieser Voxel beinhalten.

Am besten ist dieser positive Effekt in Abb. 7.3(c) für $\psi = 0{,}2$ erkennbar: Die Anzahl der *empty*-Voxel fällt gegenüber $\psi = 0{,}0$ und $\psi = 0{,}1$ geringer aus, gleichzeitig erhöht sich die Häufigkeit der *occluded*-Voxel, während die Anzahl der *unknown*-Voxel in etwa gleich bleibt. Erhöht man ψ auf 0,3 und höher, so ergibt sich ein unerwünschter Effekt: Der Anteil der *unknown*-Voxel sinkt, weil der Einfluss der *occluded*-Voxel zu groß wird.

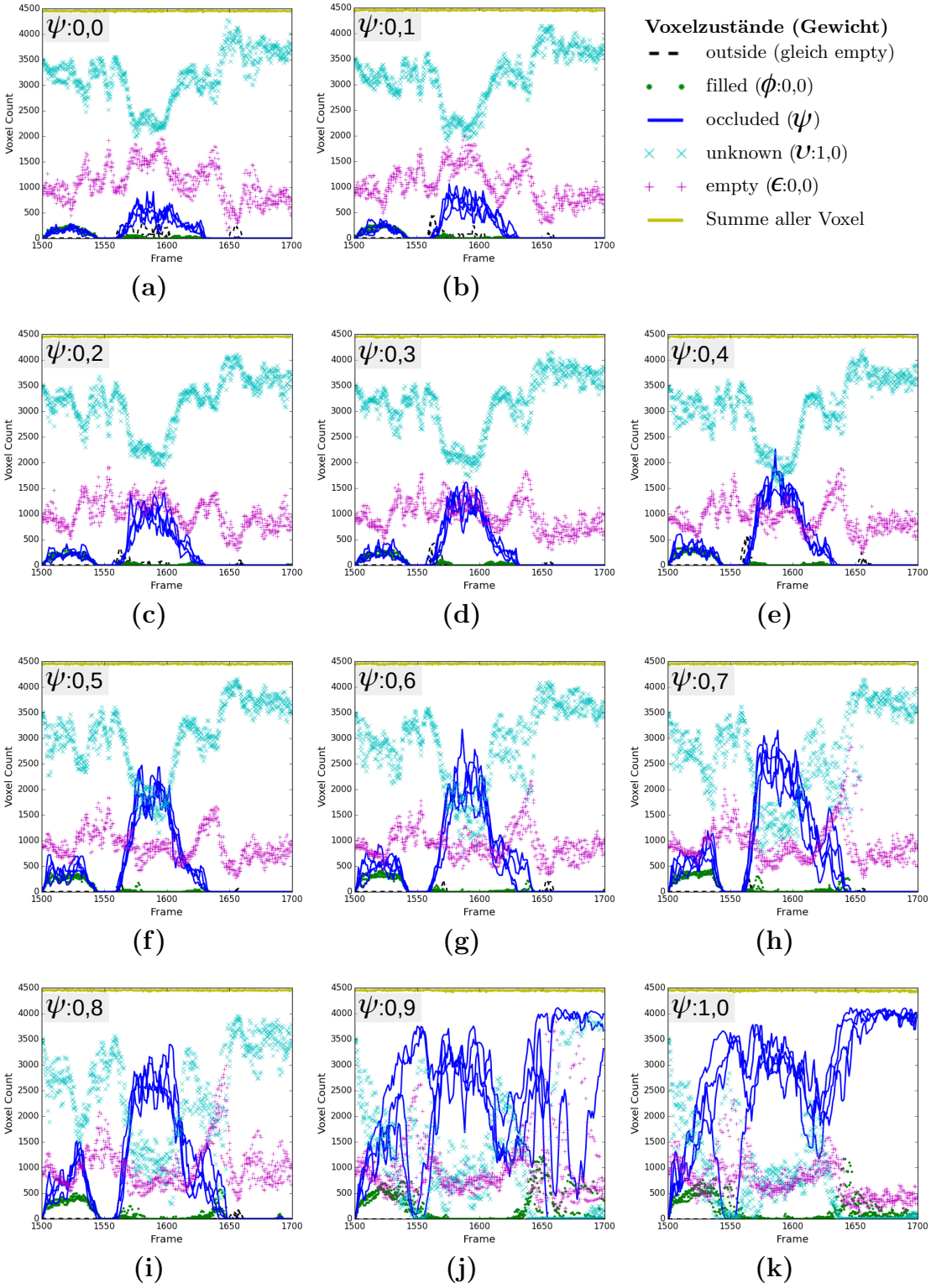


Abb. 7.3: Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel wird schrittweise erhöht von 0,0 (a) bis 1,0 (k). Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

Bei einer Gewichtung mit $\psi = 0{,}9$ in Abb. 7.3(j) enthält das Schwerpunktellipsoid nur noch wenige *unknown*-Voxel. Der Filter verliert ab Frame 1542 die Person und positioniert sich in dem Verdeckungsvolumen unterhalb des Tischs, wo er bis zum Ende der Sequenz verbleibt (Seed 1 und Seed 3). Für Seed 2 und Seed 4 ragt das Schwerpunktellipsoid weit in den Tisch und das Verdeckungsvolumen hinein, schafft es aber noch, der Person um den Tisch herum zu folgen (und anschließend unter den Tisch). Dennoch gelingt es dem Filter nur für Seed 4 auch aus dem Verdeckungsvolumen wieder heraus zu kommen und sich entsprechend außerhalb auf der Person zu lokalisieren. Die Gewichtserhöhung auf $\psi = 1,0$ verstärkt die beschriebenen Effekte weiter, was aus Abb. 7.3(k) hervorgeht. Der Filter positioniert sich in allen vier Durchgängen in dem Verdeckungsvolumen, ohne der Person zu folgen.

In Abb. 7.4 sind virtuelle Ansichten der rekonstruierten Arbeitszelle für die Gewichtung der *occluded*-Voxel mit 0,9 zu sehen. Die Schwerpunktellipsoide sind violett eingezeichnet, für den Seed 1 (linke Spalte) und den Seed 4 (rechte Spalte). Die *unknown*-Voxel sind rot dargestellt. In den Bildern (a) bis (d) wird Frame 1530 aus zwei Perspektiven gezeigt. Es lässt sich erkennen, dass die Schwerpunktellipsoide nicht gut auf dem Torso der Person liegen. Bei Seed 1 in (a) und (c) befindet sich das Ellipsoid weiter unter dem Tisch und hat eine weniger vertikal ausgerichtete Form angenommen als bei Seed 4 in (b) und (d). Die Bilder in (e) und (f) zeigen Frame 1645, in welchem sich die Person wieder aufgerichtet hat, nachdem sie unter dem Tisch war. Das Schwerpunktellipsoid befindet sich in dem Frame noch unter dem Tisch (und teilweise im Tisch). Obwohl beide Schätzungen einander ähnlich sind, gelingt es nur einem Filter (Seed 4) sich final auf der Person zu repositionieren (nicht dargestellt). Diese Ergebnisse zeigen den negativen Effekt der entsteht, wenn die Gewichtung der *occluded*-Voxel ähnlich hoch ausfällt wie die der *unknown*-Voxel. Der Filter wird dann in das Verdeckungsvolumen des Tischs „gezogen“, obwohl genügend *unknown*-Voxel außerhalb zur Verfügung stehen. Selbst wenn der Filter die Person nicht verliert und ihr folgt (Seed 4), so weicht der geschätzte Zustand stark vom realen Zustand ab, wie aus den Einzelbildern von Abb. 7.4 hervorgeht.

Synthetisches Verdeckungsvolumen

Die synthetische Erweiterung des realen Verdeckungsvolumens um die Voxel aus Abb. 7.1(b) führt zu einer größeren partiellen Objektverdeckung der Person in Teilsequenz A. Die Person bleibt dennoch unterhalb des Tischs in Bodennähe unverdeckt, wodurch in diesem Bereich weiterhin *unknown*-Voxel rekonstruiert werden. Die Ergebnisse der Untersuchung sind in Abb. 7.5 zu finden. Für $\psi = 0{,}0$ positionieren sich die Partikelellipsoide vorwiegend auf diesen *unknown*-Voxeln (während sich die Person unter dem Tisch befindet), da die *occluded*-Voxel kein Gewicht liefern. Für $\psi > 0$ bewegt sich

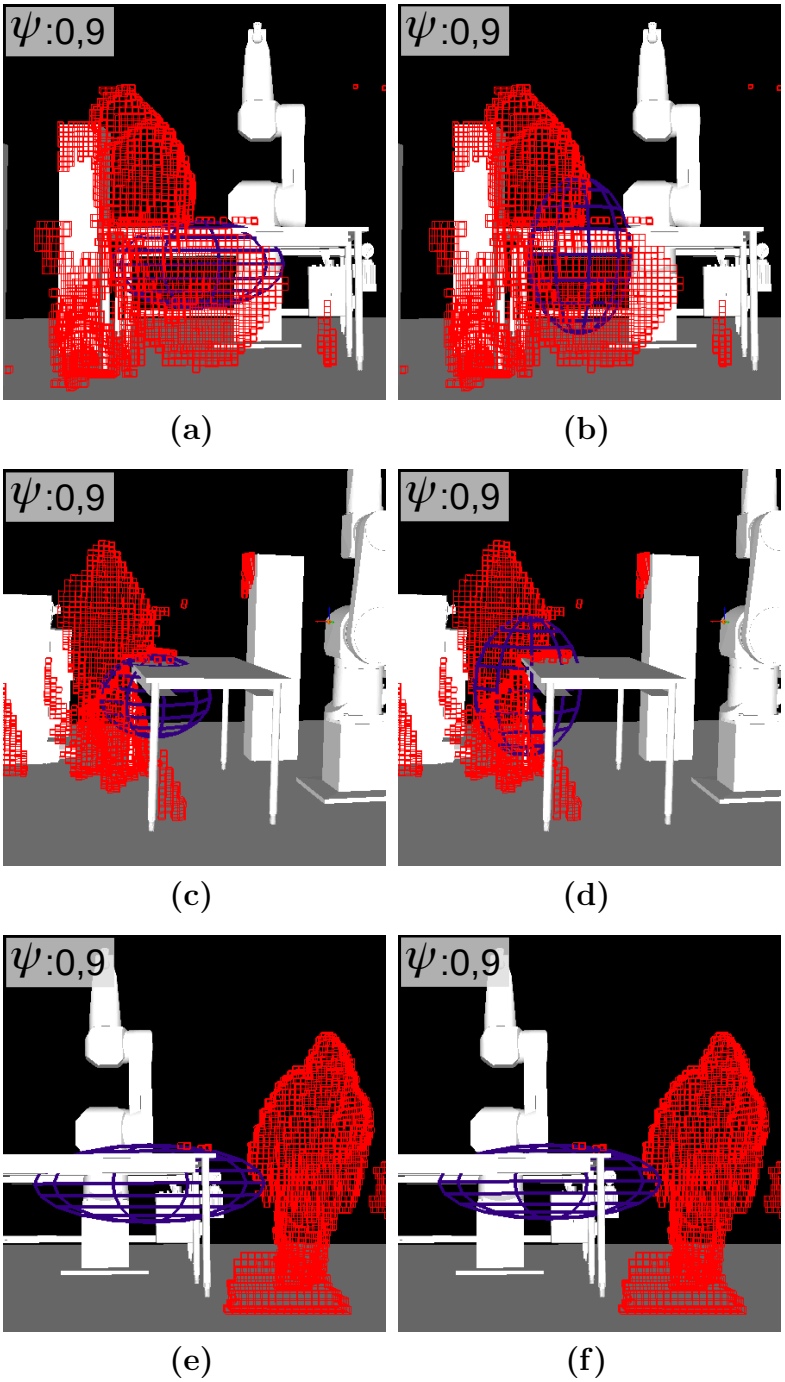


Abb. 7.4: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}9$. Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (violett), sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 1 (linke Spalte) und Seed 4 (rechte Spalte). Dargestellt sind Frame 1530 von Teilsequenz A in (a) bis (d), bei dem die Person gerade um den Tisch läuft, sowie Frame 1645 in (e) und (f), in dem die Person das reale Verdeckungsvolumen unterhalb des Tischs verlassen hat.

der Filter weiter in das Verdeckungsvolumen hinein, und zwar umso stärker, je größer das gewählte Gewicht von ψ ist. Dementsprechend steigen die Häufigkeiten der *occluded*-Voxel innerhalb der Diagramme von Abb. 7.5 mit Erhöhung von ψ an. Gleichfalls sinken

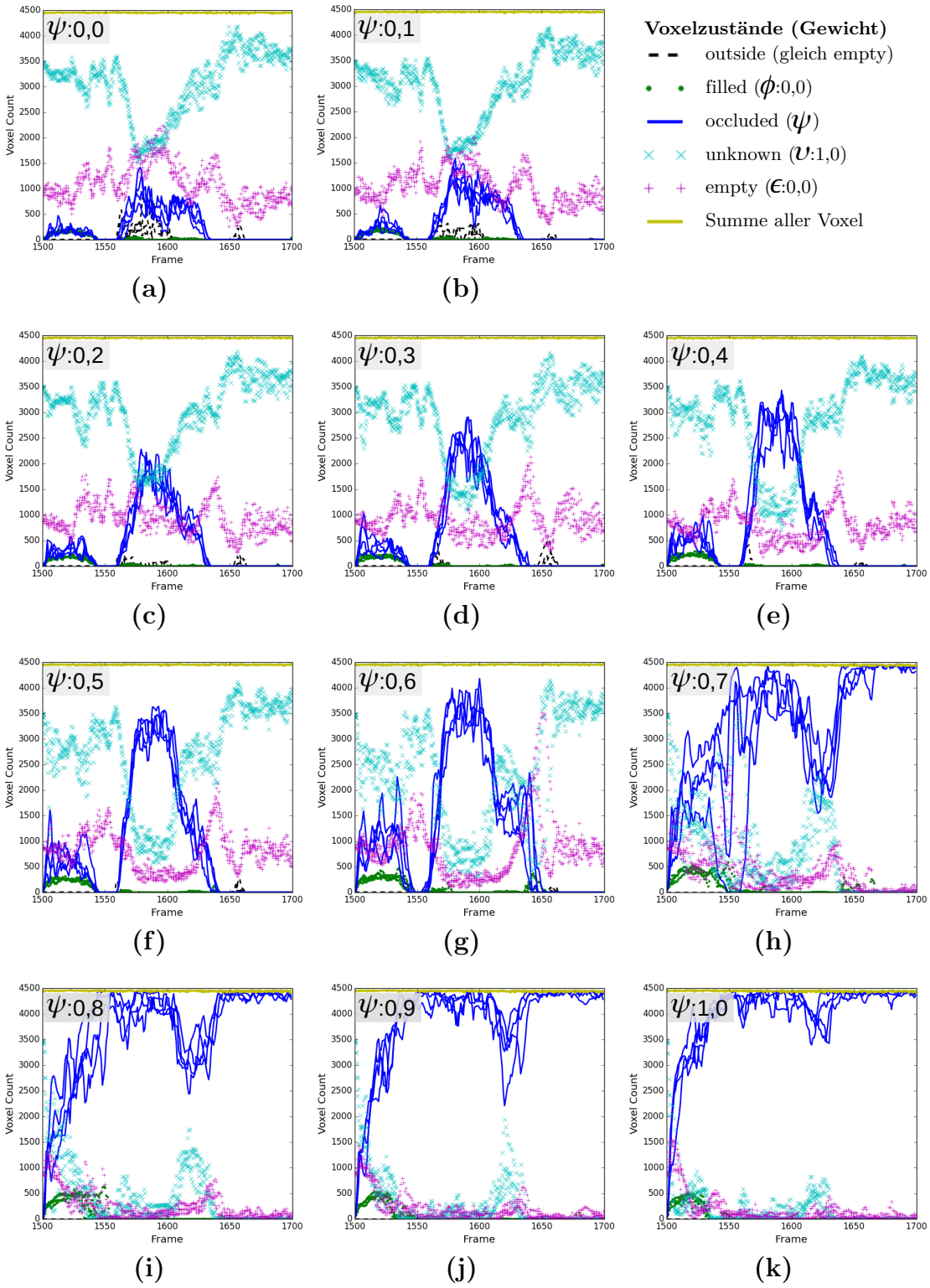


Abb. 7.5: Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel wird schrittweise erhöht von 0,0 (a) bis 1,0 (k). Das Verdeckungsvolumen unter dem Tisch wurde synthetisch vergrößert und führt zu einer partiellen Objektverdeckung der Person.

die Häufigkeiten der *empty*-Voxel im Mittel. Ab einem Gewicht von $\psi = 0{,}3$ sinkt die Anzahl der *unknown*-Voxel deutlich, ebenso wie bei dem realen Verdeckungsvolumen, was negativ bewertet wird. Ab $\psi = 0{,}7$ schafft es der Filter für alle Seeds nicht mehr, der Person aus dem Verdeckungsvolumen heraus zu folgen. Im Vergleich dazu tritt dies bei dem realen Verdeckungsvolumen (vgl. Abb. 7.3) erst ab einer Gewichtung von $\psi = 1,0$ auf. Daran lässt sich erkennen, dass die Größe des Verdeckungsvolumens auch bei der Wahl der Gewichte berücksichtigt werden sollte, um ein erfolgreiches Tracking zu ermöglichen.

7.3.2 Vollständige Objektverdeckung

Zur Erzeugung einer vollständigen Objektverdeckung der Person beim Tracking wird das Verdeckungsvolumen vergrößert, indem alle grünen Voxel aus Abb. 7.1(c) künstlich auf den Zustand *occluded* gesetzt werden. Der Ablauf des Experiments bleibt unverändert: Wieder wird das Gewicht von ψ in Zehntel-Schritten unter Beibehaltung der anderen Gewichte ($\epsilon = 0$, $\phi = 0$, $\upsilon = 1$) erhöht. Die Ergebnisse sind in Abb. 7.6 zu finden.

Für $\psi = 0{,}0$ ergeben sich in Abb. 7.6(a) erwartungsgemäß in den Schwerpunktellipsoiden die geringsten Häufigkeiten der *occluded*-Voxel im Vergleich zu den Diagrammen von (b) bis (k), da diese Voxel hierbei nicht zum Likelihood-Gewicht beitragen. Die Partikelellipsoide positionieren sich hauptsächlich auf den gemessenen *unknown*-Voxeln. Aufgrund der vollständigen Objektverdeckung etwa von Frame 1580 bis Frame 1610 (Person ist nicht sichtbar) dürften für diese Frames jedoch überhaupt keine *unknown*-Voxel gemessen werden und entsprechend auch nicht in den Schwerpunktellipsoiden enthalten sein. Die Erwartung wäre, dass sämtliche Ellipsoide mit 0 gewichtet werden und eine Filterterminierung getriggert wird. Dies ist in dem Experiment allerdings nicht der Fall, wie auch aus Abb. 7.6(a) hervorgeht: Nach Frame 1600 wächst die Häufigkeit der *unknown*-Voxel wieder auf das Niveau wie bei den anderen Diagrammen an, was für eine Weiterverfolgung der Person nach der vollständigen Objektverdeckung spricht. Die Erklärung lässt sich der Visualisierung der Rekonstruktionsdaten entnehmen: Außerhalb des synthetischen Verdeckungsvolumens existieren noch einige *unknown*-Voxel, weil das Verdeckungsvolumen für das Experiment nachträglich künstlich im Voxelraum vergrößert wurde. Eine gleichzeitige Löschung sämtlicher *unknown*-Voxel erfolgte jedoch nicht. Wenige verbliebene *unknown*-Voxel oberhalb der Tischplatte führen deshalb zu einer Diffusion des Filters durch den Tisch. Das Schwerpunktellipsoid ragt dabei zwischen den Frames 1580 und 1610 etwas durch den Tisch in das Verdeckungsvolumen hinein, liegt aber größtenteils auf *empty*-Voxeln. Der hohe Anteil an *empty*-Voxeln zeigt sich auch im Diagramm von Abb. 7.6(a). Wäre die Person real vollständig verdeckt

gewesen, so hätten die Diagramme dies entsprechend gezeigt und es wäre in diesem Beispiel zu einer Terminierung des Filters gekommen.

Gewichtet man die *occluded*-Voxel mit $\psi > 0$, so ändert sich das Filterverhalten. In Abb. 7.6(b) ist der gewünschte Effekt bereits für $\psi = 0{,}1$ zu sehen: Der Filter bewegt sich unter den Tisch in das Verdeckungsvolumen und folgt der Person auch wieder heraus. Zu erkennen ist dies im Diagramm daran, dass bei Frame 1590 fast ausschließlich *occluded*-Voxel innerhalb des Schwerpunktellipsoids liegen, im weiteren Verlauf aber wieder die *unknown*-Voxel den größten Anteil ausmachen. Im Diagramm von Abb. 7.6(c) mit $\psi = 0{,}2$ sieht der Verlauf ebenso recht gut aus, wobei die Häufigkeit der *occluded*-Voxel früher ansteigt und sich erst später wieder absenkt. Eine Erhöhung des Gewichts für die *occluded*-Voxel sorgt demnach für eine frühere Bewegung des Filters in das Verdeckungsvolumen und damit umgekehrt für ein verzögertes Austreten des Filters aus dem Verdeckungsvolumen heraus. Diese Effekte verstärken sich bei weiterer Gewichtserhöhung von ψ, zu sehen in Abb. 7.6(d) bis (f), und führen bei einer Gewichtung mit $\psi = 0{,}6$ in (g) zu einem Verbleib des Filters in dem Verdeckungsvolumen (für drei Seeds). Eine weitere Erhöhung des Werts von ψ resultiert stets in einem Trackingverlust der Person, zu sehen in Abb. 7.6(h) bis (k).

7.3.3 Zusammenfassung

In dem Experiment zur Gewichtung der Verdeckungsvolumina geht es insbesondere um das Zusammenspiel der Gewichte für die *occluded*-Voxel (ψ) und die *unknown*-Voxel (υ), wobei letztere konstant den Wert 1 erhalten. Für das angestrebte Filterverhalten soll in jedem Zeitschritt k die Anzahl verfügbarer *unknown*-Voxel innerhalb der Schwerpunktellipsoide maximiert werden. Die restlichen Voxel sollen möglichst zur Menge der *occluded*-Voxel gehören, wenn sich die Person (mindestens teilweise) in einem Verdeckungsvolumen befindet.

Die Ergebnisse der gewichtsmäßigen Gleichsetzung der *occluded*-Voxel für $\psi = 0.0$ mit Voxeln der Zustände *filled* und *empty* zeigen: Der Filter konzentriert sich ausschließlich auf die *unknown*-Voxel, da nur diese ein Gewicht liefern. Dadurch begibt er sich nur begrenzt in das Verdeckungsvolumen und kann der Person bei einer vollständigen Objektverdeckung nicht mehr ausreichend gut in den verdeckten Bereich folgen. In Abschnitt 7.6 zur Gesamtevaluierung werden weitere Beispiele gezeigt, die diesen Effekt noch sichtbarer machen.

Die Gewichtung der *occluded*-Voxel mit $\psi > 0$ ermöglicht eine Personenverfolgung durch Verdeckungsvolumina hindurch. Ein zu hoher Wert für ψ kann jedoch zu einem verfrühten Diffundieren des Filters weg von der Person in das Verdeckungsvolumen führen. Auch

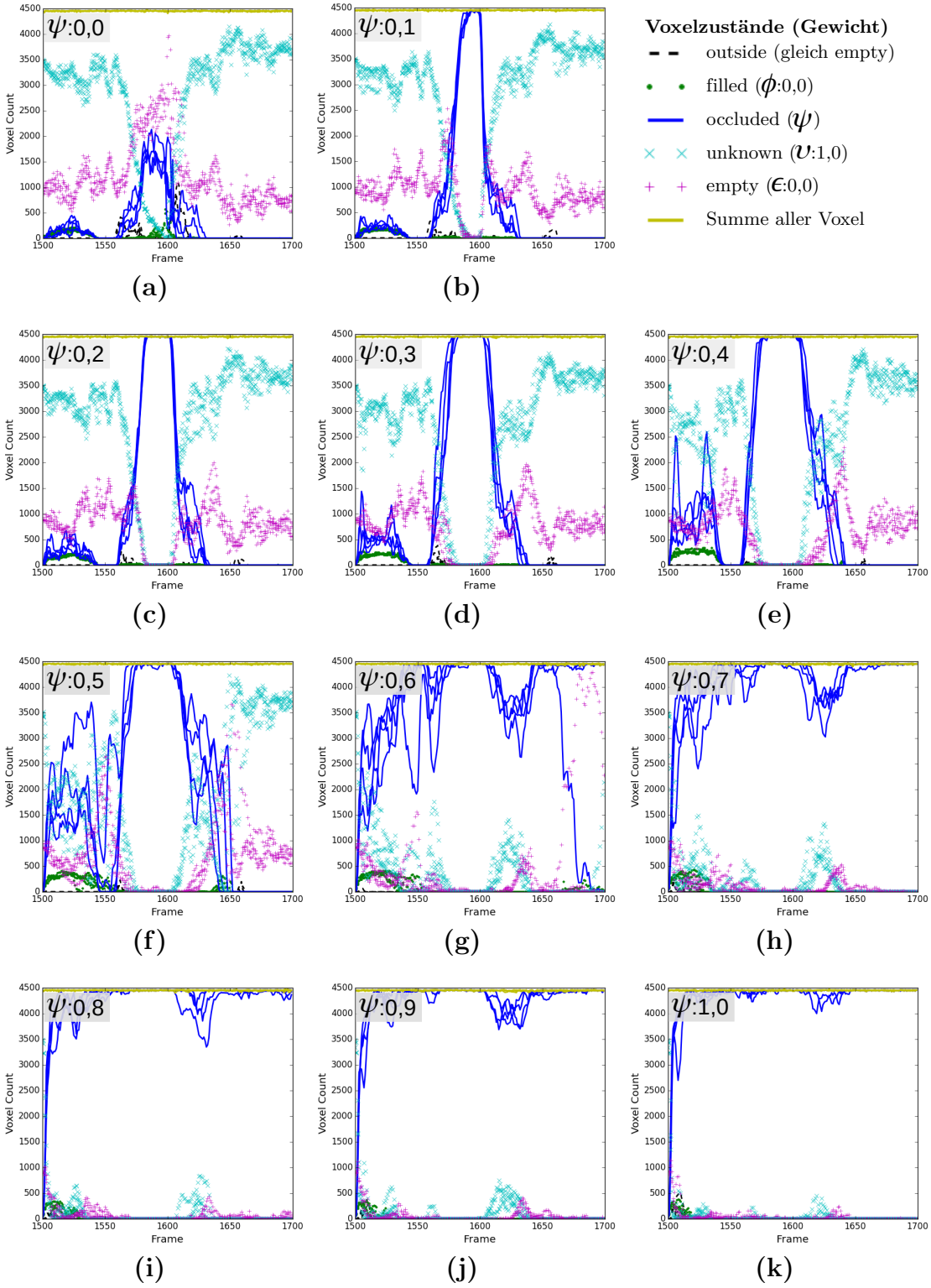


Abb. 7.6: Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel wird schrittweise erhöht von 0,0 (a) bis 1,0 (k). Das Verdeckungsvolumen unter dem Tisch wurde synthetisch vergrößert und führt zu einer vollständigen Objektverdeckung der Person.

kann der Filter der Person dabei meist nicht mehr aus dem Verdeckungsvolumen heraus folgen. Bei der realen Verdeckungssituation aus Abb. 7.1(a) trat dies bei sämtlichen Seeds für $\psi = 1,0$ auf. Bei den künstlich vergrößerten Verdeckungsvolumina der Abbildungen 7.1(b) und (c) konnte dies bereits bei einem geringeren Gewicht von $\psi = 0{,}6$ bzw. $\psi = 0{,}7$ beobachtet werden. Größere Verdeckungsvolumina führen zu diesem Effekt bei gleichzeitig geringeren Werten von ψ.

7.4 Nichtlineare Verstärkung in der Likelihood-Funktion

In dem vorangegangenen Experiment wurden die Voxelzustände linear gewichtet. Nun soll die nichtlineare Verstärkung aus Abschnitt 6.4.1 untersucht werden (Gain). Ellipsoide, die einen hohen Anteil an *unknown*-Voxeln enthalten, bekommen damit eine zusätzliche Verstärkung wie in Formel (7.2) angegeben. Das Ziel besteht darin, den Filter in der Nähe eines Verdeckungsvolumens möglichst gut auf dem Teil der Person zu halten, der aus dem Verdeckungsvolumen herausschaut.

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M|_k^j} \cdot e^{\frac{|\Upsilon_k^j|}{|M_k^j|}} \tag{7.2}$$

7.4.1 Partielle Objektverdeckung

Zunächst wird wieder das reale Verdeckungsvolumen aus Abb. 7.1(a) betrachtet. In Abb. 7.7 sind die Ergebnisdiagramme der Teilsequenz A für die Gewichtung der *occluded*-Voxel mit $\psi = 0{,}0$ bis $0{,}3$ (in Zehntelschritten) und der nichtlinearen Verstärkung (Gain) den entsprechenden Diagrammen der linearen Gewichtung des vorangegangenen Experiments (ohne Gain) gegenübergestellt. Alle weiteren Diagramme für die Gewichtung von ψ in Zehntelschritten mit Werten bis 2,1 können den Abbildungen 9.25 und 9.26 entnommen werden.

Betrachtet man zunächst die *occluded*-Voxel in den Diagrammen von Abb. 7.7, so lässt sich erkennen, dass deren Häufigkeiten bei der nichtlinearen Verstärkung im Schnitt etwas geringer ausfallen als bei der linearen Gewichtung. Auch die Häufigkeiten der *empty*- und *outside*-Voxel sind reduziert, was in jedem Fall erwünscht ist. Die gezählten *outside*-Voxel sind bei der betrachteten Teilsequenz A solche, die sich im Boden befinden. Demnach ragt das Schwerpunktellipsoid nicht weit in den Boden hinein. Dies ist positiv zu bewerten. Auch die *filled*-Voxel zeigen tendenziell eine Verringerung in ihren Häufigkeiten, was aber nur schwer erkennbar ist, da sie absolut nur in geringer

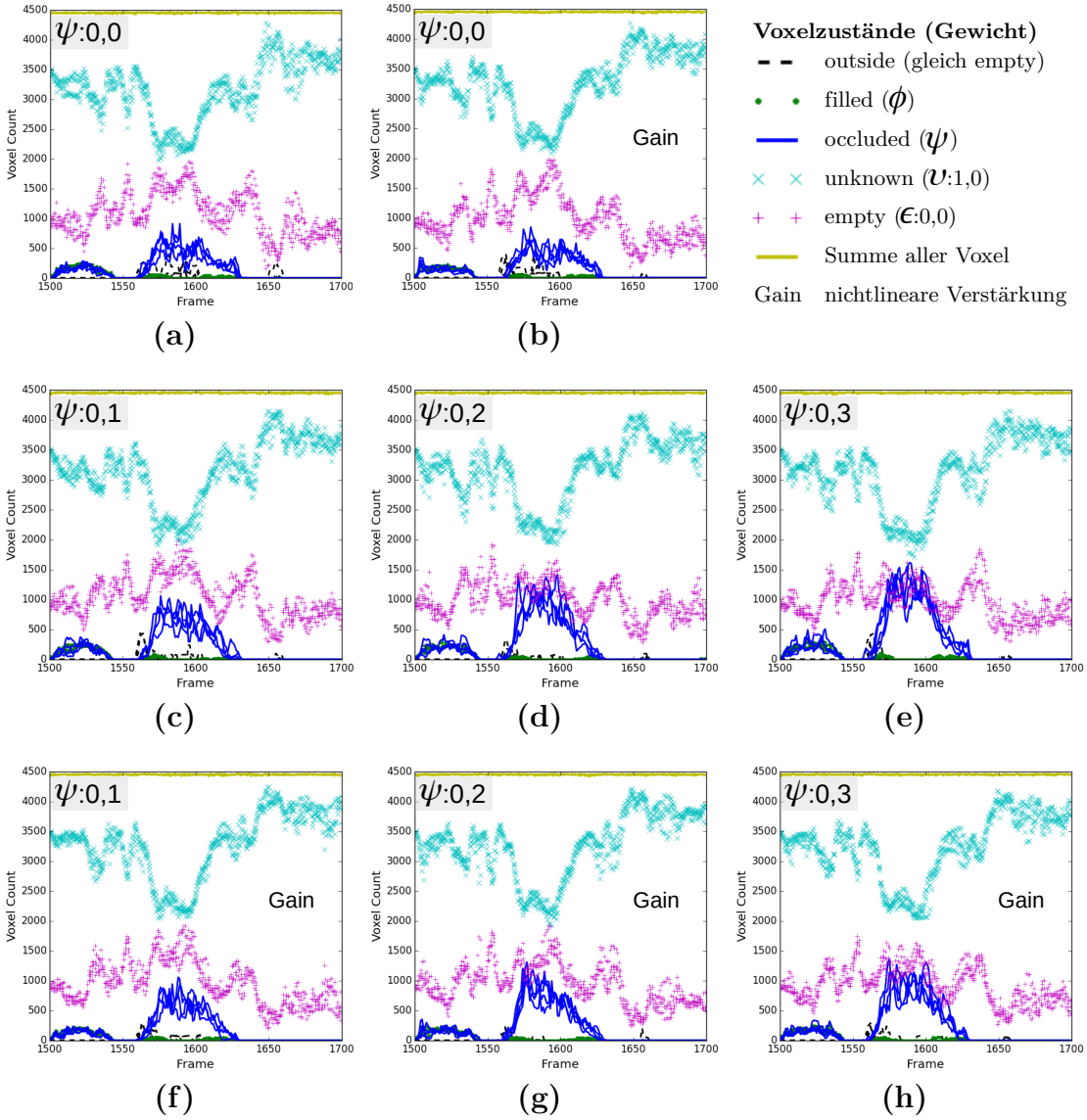


Abb. 7.7: Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel wird schrittweise erhöht von 0,0 bis 0,3. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person. Die lineare Gewichtung in (a), (c), (d), (e) wird jeweils der nichtlinearen Verstärkung der *unknown*-Voxel (Gain) in (b), (f), (g) und (h) gegenübergestellt.

Anzahl vorkommen. Der Filter platziert sich wie erhofft bei der nichtlinearen Verstärkung im Mittel etwas mehr auf den *unknown*-Voxeln, wobei dies erst bei höheren Gewichten von $\psi > 0{,}3$ deutlicher sichtbar wird (vgl. Abb. 9.25 und 9.26).

Die Diagramme der nichtlinearen Verstärkung in Abb. 7.7 sind trotz der beschriebenen Unterschiede sehr ähnlich derer von der linearen Gewichtung des vorangegangenen Experiments. In Abb. 7.8 sind dazu virtuelle Bilder für $\psi = 0{,}1$ der Frames 1526 und 1670 in (a) und (e) respektive (b) und (f) dargestellt. Es lässt sich nur schwer beurteilen,

ob die Schwerpunktellipsoide mit oder ohne Gain besser auf der Person liegen. Dies variiert auch von Frame zu Frame. Für $\psi = 0,2$ liegt in Abb. 7.8(h) das Ellipsoid etwas mehr auf den *unknown*-Voxeln in Bodennähe im Vergleich zu (d), was aber kaum merklich ist. Erst für $\psi > 0{,}2$ hebt sich die nichtlineare Verstärkung auch in den Bildern deutlicher von der linearen Gewichtung ab, was im Folgenden noch erläutert wird.

Von allen Diagrammen mit Gain sehen die Ergebnisse bei $\psi = 0{,}2$ und $\psi = 0{,}3$ am besten aus, da hierbei im Schnitt die geringste Anzahl an *empty*-Voxeln auftritt. Für $\psi > 0{,}3$ (vgl. Abb. 9.25) steigen die Häufigkeiten der *occluded*-Voxel, was aber mit einer Verringerung der *unknown*-Voxel einhergeht und unerwünscht ist. Dieser Effekt wurde bereits im vorhergehenden Experiment beschrieben: Der Filter gelangt mit zunehmender Erhöhung von ψ weiter in das Verdeckungsvolumen unterhalb des Tischs. Bei $\psi = 1,0$ sind deutlich mehr *occluded*-Voxel als *unknown*-Voxel in dem Schwerpunktellipsoid enthalten, da der Filter zu früh in die Verdeckung gezogen wird. Dennoch sorgt der Gain dafür, dass der Filter der Person auch bei $\psi = 1$ im Gegensatz zur linearen Gewichtung noch aus dem Verdeckungsvolumen heraus folgen kann. Vergleicht man die Diagramme von Abb. 9.25 mit Abb. 7.3, so lässt sich erkennen, dass das Diagramm von $\psi = 1$ (mit Gain) dem Diagramm der linearen Gewichtung von $\psi = 0{,}6$ entspricht. Die Erhöhung von ψ hat demnach aufgrund der Verstärkung der *unknown*-Voxel einen schwächeren Effekt als bei der linearen Gewichtung. In Abb. 7.8 wird dazu Frame 1518 für $\psi = 0{,}6$ gezeigt. Während die Person um den Tisch herum läuft, wird das Schwerpunktellipsoid bei der linearen Gewichtung in Abb. 7.8(c) in Richtung des Verdeckungsvolumens „gezogen“. Bei Anwendung des Gains in Abb. 7.8(g) hingegen tritt dieser Effekt nicht deutlich zutage.

In Abb. 9.26 sind Diagramme für $\psi > 1$ zu finden. Erkennbar ist eine tendenzielle Verstärkung der Schwankungen zwischen den Graphen unterschiedlicher Seeds für $\psi > 0{,}9$. Ab der Gewichtung mit $\psi = 1,5$ verbleiben die Filter zunehmend in dem Verdeckungsvolumen und verlieren die Person.

Die detaillierte Betrachtung virtueller Bilder des 3D-Viewers bringt weitere Unterschiede der additiven Gewichtung mit und ohne Gain zum Vorschein, die nun kurz beschrieben werden sollen. Generell kann über alle betrachteten Seeds und Parametrisierungen von ψ hinweg beobachtet werden, dass die Schwerpunktellipsoide bei der nichtlinearen Verstärkung öfter eine länglichere Form aufweisen als bei der linearen Gewichtung, wenn die Person aufgerichtet ist. Dies wird in Abb. 7.8 für Frame 1526 in (a) und (e) sowie Frame 1670 in (b) und (f) für $\psi = 0{,}1$ und Seed 1 gezeigt. Die lineare Gewichtung ist jeweils oben und die nichtlineare Verstärkung unten zu sehen. Die länglichere Form der Schwerpunktellipsoide kann eher positiv bewertet werden, da sie besser in den *unknown*-Voxeln liegt. Allerdings wurde damit öfter eine leichte Latenz beim Eintreten

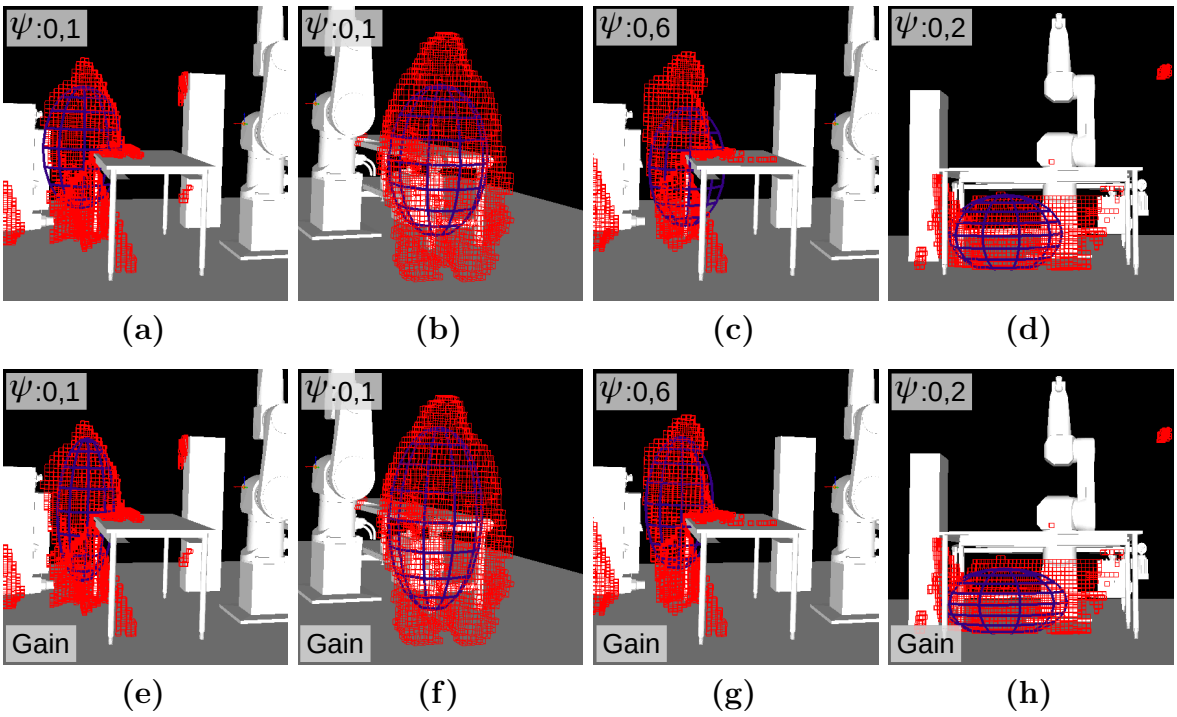


Abb. 7.8: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit variierenden Werten von ψ, mit und ohne nichtlinearer Verstärkung (Gain). Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (violett) sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 1. Dargestellt sind verschiedene Frames.

der Person in das Verdeckungsvolumen unter dem Tisch beobachtet, was jedoch nicht immer der Fall war. Beim Heraustreten der Person aus dem Verdeckungsvolumen konnte für $\psi = 0{,}1$ und $\psi = 0{,}2$ ohne Gain das Schwerpunktellipsoid mehrfach schneller der Person folgen als bei der Gewichtung mit Gain. Mit steigendem Wert von ψ ist jedoch das Tracking der nichtlinearen Verstärkung vorteilhaft. Das Ellipsoid gelangt dann oft zügiger aus dem Verdeckungsvolumen heraus und richtet sich, wie erwähnt, stärker entlang der vertikalen Achse aus, da die „Anziehung“ der *unknown*-Voxel bei höherem ψ im Vergleich zur linearen Gewichtung stärker ausfällt.

Synthetisches Verdeckungsvolumen

Als nächstes wird die nichtlineare Verstärkung für das synthetische Verdeckungsvolumen von Abb. 7.1(b) betrachtet, das unter dem Tisch zu einer etwas größeren partiellen Objektverdeckung führt als das reale Verdeckungsvolumen. Die Ergebnisdiagramme dazu sind in Abb. 9.27 zu finden. Für $\psi = 0{,}0$ bis $\psi = 0{,}2$ ist für verschiedene Frames eine Verringerung der Häufigkeiten der *occluded*-Voxel bei der nichtlinearen Verstärkung (Gain) gegenüber der linearen Gewichtung zu sehen. Ähnlich wie bei dem realen Verdeckungsvolumen sind die Unterschiede jedoch gering. Für $\psi = 0{,}3$ und $\psi = 0{,}4$ werden ohne Gain mehr *occluded*-Voxel gezählt als mit Gain, bezogen auf

die Frames, bei denen sich die Person unter dem Tisch befindet. Durch die Präferenz der *unknown*-Voxel schneidet die nichtlineare Verstärkung hierbei besser ab. Ab der Gewichtung mit $\psi = 0{,}5$ verringert sich jedoch die Anzahl der *unknown*-Voxel innerhalb des Schwerpunktellipsoids deutlich, sowohl bei der linearen Gewichtung als auch mit dem Gain. Das Schwerpunktellipsoid begibt sich dabei zunehmend auf die *occluded*-Voxel, weg von den *unknown*-Voxeln. Die weitere Erhöhung des Werts von ψ verstärkt diesen Effekt und führt dann wieder zu den bereits beschriebenen Ergebnissen, bei denen der Filter zu früh in das Verdeckungsvolumen driftet und aus diesem auch nicht wieder heraus gelangt.

Für die betrachtete Verdeckungssituation sieht die Gewichtung mit $\psi = 0{,}3$ und $\psi = 0{,}4$, den Diagrammen nach zu urteilen, am besten aus. Die Anzahl der *occluded*-Voxel ist hierbei höher als für $\psi = 0{,}1$ oder $\psi = 0{,}2$, bei einer ähnlich hohen Anzahl an *unknown*-Voxeln. Die Häufigkeiten der unerwünschten *empty*- und *filled*-Voxel werden damit minimiert. Die Analyse der aufgezeichneten Bildersequenzen des 3D-Viewers bestätigt die Auswertung der Diagramme und die beschriebene Erwartungshaltung. In Gänze kann für die betrachtete synthetische Verdeckungssituation ein besseres Abschneiden der nichtlinearen Verstärkung festgestellt werden. Für $\psi = 0{,}0$ bis $\psi = 0{,}5$ folgt der Filter in den meisten Durchgängen der Person zügiger unter den Tisch und auch wieder aus dem Verdeckungsvolumen heraus als bei der linearen Gewichtung. Der Gain führt wie erwartet zu einer stärkeren Positionierung des Filters auf den *unknown*-Voxeln, was bei dem konkreten synthetischen Verdeckungsvolumen zu einer relativ flachen Form des Filters unterhalb des Tischs führt.

7.4.2 Vollständige Objektverdeckung

Das Experiment zur nichtlinearen Verstärkung wird fortgesetzt für das synthetische Verdeckungsvolumen aus Abb. 7.1(c), das bei der Teilsequenz A zu einer vollständigen Objektverdeckung der Person führt. Die Diagramme mit den aufgetragenen Voxelhäufigkeiten der Schwerpunktellipsoide sind in Abb. 9.28 zu finden. Sie zeigen, dass die beste Gewichtung mit $\psi = 0{,}3$ sowie $\psi = 0{,}4$ erreicht werden kann. Die Ergebnisse sind dabei tendenziell etwas besser als die der linearen Gewichtung.

Im Folgenden werden visuelle Beobachtungen beschrieben. Ohne eine Gewichtung der *occluded*-Voxel bleibt das Schwerpunktellipsoid für $\psi = 0{,}0$ wieder vor dem Tisch stehen und diffundiert dann aufgrund der verbliebenen *unknown*-Voxel durch die Tischplatte zur anderen Tischseite, wo sich die Person aus dem Verdeckungsvolumen heraus begibt. Der Filter folgt der Person dabei nicht durch das Verdeckungsvolumen und verliert nur

aufgrund der gegebenen Störvoxel den Track der Person nicht. Dieser Effekt wurde bereits in Abschnitt 7.3.2 ausführlich beschrieben.

Mit $\psi = 0{,}1$ folgt der Filter der Person wie gewünscht durch das Verdeckungsvolumen. Für $\psi = 0{,}1$ und $\psi = 0{,}2$ sind die Unterschiede zwischen der linearen Gewichtung und deren nichtlinearer Verstärkung wieder gering. Ansonsten sind in etwa die gleichen Effekte wie in dem vorangegangenen Experiment zu beobachten. Der Filter bleibt mit Gain vor dem Verdeckungsvolumen länger auf den *unknown*-Voxeln positioniert als bei der linearen Gewichtung. Beim Verlassen des Verdeckungsvolumens hingegen ist es (für $\psi > 0{,}2$) oft umgekehrt: Das Ellipsoid gelangt besser aus dem Verdeckungsvolumen heraus und nimmt wieder schneller eine vertikale Form an. Bei der linearen Gewichtung hingegen benötigt das Schwerpunktellipsoid meist einige Frames länger, um in die vertikale Ausrichtung zu gelangen. Bei höheren Werten von ψ sind diese Effekte ausgeprägter.

Ein deutlicher Unterschied, der bei der vollständigen Objektverdeckung im Vergleich zu den partiellen Objektverdeckungen für $\psi > 0$ auftritt, ist folgender: Bei der Bewegung der Person in das Verdeckungsvolumen bleibt der Filter solange vor dem Tisch stehen, bis dort fast keine *unknown*-Voxel mehr rekonstruiert werden. Erst dann bewegt sich der Filter in die Mitte des Verdeckungsvolumens, was mit einem kurzen Einpendelvorgang verbunden ist. Sobald die Person aus dem Verdeckungsvolumen herauskommt, beschleunigt das Ellipsoid und folgt dieser. Bei $\psi = 0{,}1$ ist das Beschleunigen und Überschwingen des Filters beim Austritt der Person aus dem Verdeckungsvolumen besonders ausgeprägt und das Stehenbleiben innerhalb des Verdeckungsvolumens fast nicht sichtbar, weil das Schwerpunktellipsoid in etwa so lange vor dem Tisch bleibt, bis die Person auf der anderen Seite wieder zum Vorschein kommt und dabei *unknown*-Voxel erzeugt. Das Schwerpunktellipsoid wandert dabei ohne Unterbrechung durch das Verdeckungsvolumen. Bei $\psi = 0{,}4$ gelangt der Filter früher in die Mitte des Verdeckungsvolumens und verweilt dort, bis die *unknown*-Voxel, die rekonstruiert werden, wenn die Person aus dem Verdeckungsvolumen hervorkommt, den Filter wieder auf sich „ziehen“.

Einzelne Durchgänge mit höherer Gewichtung, beispielsweise mit $\psi = 0{,}6$ (Seed 3), sind hinsichtlich der Filterbewegung durch das Verdeckungsvolumen besser als das Tracking mit einem kleineren Wert von ψ, aufgrund der gleichmäßigen Bewegung des Filters. Allerdings folgt das Schwerpunktellipsoid dann zu Beginn von Teilsequenz A der Person nicht um den Tisch herum, sondern diffundiert vorzeitig in das Verdeckungsvolumen. Das bessere Folgen des Filters durch das Verdeckungsvolumen, welches der ungefähren Bewegung der Person unter dem Tisch entspricht, wird dabei mit einem schlechteren Filterverhalten außerhalb des Verdeckungsvolumens erkauft. Daher wird ein zu hoher Wert von ψ bei der nichtlinearen Verstärkung ebenso wenig empfohlen wie bei der

linearen Gewichtung, wenn es sich um ein Verdeckungsvolumen mit Ausmaßen handelt, die zu einer größeren partiellen oder vollständigen Objektverdeckung des Objekts von Interesse führen können.

7.4.3 Zusammenfassung

Für Werte von $\psi < 0{,}3$ ist schwer zu entscheiden, ob eine additive Gewichtung mit oder ohne Gain zu den besseren Tracking-Ergebnissen führt, da die Ergebnisse sehr wechselhaft sind (je nach Seed, Frame und Verdeckungsvolumen). Von der Tendenz her kann der nichtlinearen Verstärkung dennoch der Vorrang gegeben werden, sowohl bei den ausgewerteten Diagrammen als auch bei der visuellen Bewertung, aufgrund der mindestens leichten Präferenz der *unknown*-Voxel durch den Filter, welche die rekonstruierten Personen repräsentieren. Bei Gewichtungen ab $\psi = 0{,}3$ kann die nichtlineare Verstärkung durchschnittlich als etwas vorteilhafter bewertet werden gegenüber der rein linearen Gewichtung, weil der Filter dann stärker die Voxel des Zustands *unknown* gegenüber der *occluded*-Voxel bevorzugt. Betrachtet man alleinig die nichtlineare Verstärkung, so zeigten sich die besten Ergebnisse für die betrachteten Verdeckungsvolumina bei einer Gewichtung mit $\psi = 0{,}2$ bis $\psi = 0{,}4$.

Beobachtet wurde weiterhin, dass die Schwerpunktellipsoide bei der nichtlinearen Verstärkung eine vertikalere Form annehmen, wenn die Person eine aufrechte Haltung besitzt. Bei dem Eintreten des Ellipsoids in das Verdeckungsvolumen der Teilsequenz A ist tendenziell eine Verzögerung gegenüber der linearen Gewichtung zu bemerken. Das Austreten des Ellipsoids aus dem Verdeckungsvolumen hingegen gelingt bei der nichtlinearen Verstärkung für $\psi > 0{,}2$ meist schneller als ohne den verstärkenden Term. Eine aufrechte vertikale Form des Schwerpunktellipsoids wird dabei oft zügiger eingenommen.

Wird die Gewichtung der *occluded*-Voxel zu hoch gewählt, so diffundiert der Filter auch bei der nichtlinearen Verstärkung frühzeitig in das Verdeckungsvolumen, wenn sich die Person in Tischnähe befindet. Auch kann der Filter das Verdeckungsvolumen dann nicht mehr verlassen, um der Person zu folgen.

Aufgrund der dargestellten Ergebnisse wird in den weiteren Experimenten dieser Dissertation häufig die nichtlineare Verstärkung eingesetzt (Gain).

7.5 Bestrafende Terme in der Likelihood-Funktion

In diesem Abschnitt wird die Gewichtung der verbleibenden zwei Voxelzustände *filled* und *empty* thematisiert. Diese werden in der Likelihood-Funktion bestraft. Für die Untersuchungen wird ausschließlich das reale Verdeckungsvolumen ohne synthetische Vergrößerung herangezogen, da dies für die Betrachtungen dieses Abschnitts ausreichend erscheint. In der Gesamtevaluierung von Abschnitt 7.6 wird die Wirkung der Bestrafungsterme hingegen auch bei größeren Verdeckungsvolumina untersucht.

7.5.1 Bestrafung der *filled*-Voxel

Die *filled*-Voxel repräsentieren die statischen Objekte der Roboterarbeitszelle und werden mithilfe von 3D-Modellen dieser Objekte bestimmt. Wie in Abschnitt 6.4.2 beschrieben, wird in dieser Dissertation davon ausgegangen, dass keine gleichzeitige Belegung eines *filled*-Voxels durch eine Person und ein statisches Objekt möglich ist. Demzufolge erscheint es auch nicht plausibel, Partikelzustände zuzulassen, bei denen sich *filled*-Voxel innerhalb der zugehörigen Ellipsoide befinden. Das Ersetzen sämtlicher Ellipsoide auf die dies zutrifft, wäre jedoch sehr limitierend, da solche Fälle häufig auftreten, weil ein Ellipsoidmodell keine genaue Approximation eines Menschen erlaubt und die Filterbewegung damit entsprechend stark eingeschränkt wird.

Deshalb wird eine Bestrafung aller *filled*-Voxel Φ_k^j vorgeschlagen, die sich innerhalb des jeweils betrachteten Partikelellipsoids j befinden. Entsprechend Formel 7.3 wird ein bestrafender Term mit dem additiven Gewicht, bestehend aus der positiven Gewichtung der *unknown*- und *occluded*-Voxel, multipliziert.

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{\left(\frac{|\Phi_k^j|}{|M_k^j|} - s\right)^a}{(-s)^a} \quad \text{mit} \quad s = 1, \quad 1 \leq a, \quad a \in \mathbb{N} \tag{7.3}$$

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{e^{\frac{|\Upsilon_k^j|}{|M_k^j|}}}{e} \cdot \frac{\left(\frac{|\Phi_k^j|}{|M_k^j|} - s\right)^a}{(-s)^a} \quad \text{mit} \quad s = 1, \quad 1 \leq a, \quad a \in \mathbb{N} \tag{7.4}$$

Die Bestrafung des Partikels ist umso größer, je mehr *filled*-Voxel in dem Ellipsoid enthalten sind und je höher der Exponent a gewählt wird. Für diesen werden die Ergebnisse von $a = 2, 8, 32$ miteinander verglichen. Der Kontrollparameter $s = 1$ bleibt konstant, um den gewünschten Funktionsverlauf zu erhalten. In Formel 7.4, die auch in den Experimenten diesen Abschnitts eingesetzt wird, ist zusätzlich noch der Term zur nichtlinearen Verstärkung der *unknown*-Voxel (Gain) Teil der Berechnung.

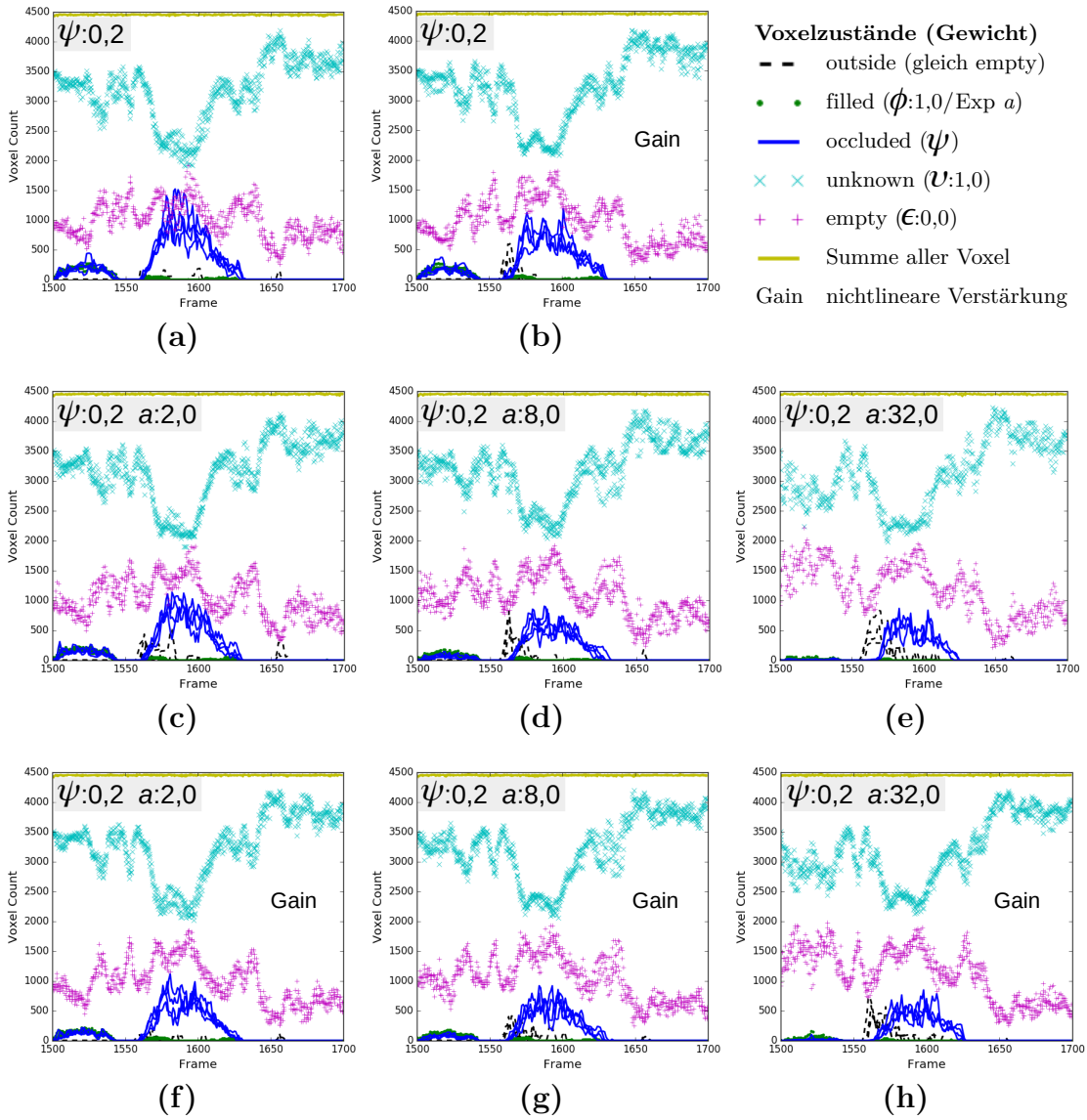


Abb. 7.9: Bestrafung von *filled*-Voxeln mit variierendem Exponenten a des Bestrafungsterms, mit und ohne nichtlineare Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0{,}2$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

Es wird wieder die Teilsequenz A eingesetzt, um die Auswirkung der Bestrafung zu bewerten. Die *occluded*-Voxel werden konstant mit $\psi = 0{,}2$ gewichtet, da dies in den vorherigen Untersuchungen im Durchschnitt zu guten Ergebnissen führte. In Abb. 7.9(a) und (b) sind zum Vergleich die entsprechenden Diagramme der vorangegangenen Experimente dargestellt (lineare Gewichtung und nichtlineare Verstärkung). Die Diagramme von Abb. 7.9(c) bis (h) zeigen die Ergebnisse, bei denen eine zusätzliche Bestrafung der *filled*-Voxel erfolgt. In den Abbildungen 9.29 bis 9.33 sind ergänzend alle Diagramme für die Gewichtungen der *occluded*-Voxel mit $\psi = 0{,}1$ bis $\psi = 0{,}5$ (in Zehntelschritten)

zu finden. Die nun wie folgt beschriebenen qualitativen Zusammenhänge ändern sich dabei jedoch unter Variation von ψ nur unwesentlich.

Zunächst sollen die Unterschiede der *unknown*-Voxel betrachtet werden: Auffällig ist zu Beginn der Teilsequenz A für den Zeitraum vor Frame 1550 eine erhöhte Varianz der Graphen von den vier Seeds bei einer recht hohen Bestrafung der *filled*-Voxel mit $a = 32$. Dies lässt sich mit dem starken „Abstoßungseffekt“ des Filters durch den Tisch begründen. Das resultierende Schwerpunktellipsoid liegt weiter vom Tisch entfernt, was auch aus Abb. 7.10(b) im Vergleich zu (a) hervorgeht. Die Bestrafung mit $a = 8$ erhöht lokal die Varianz der Graphen zwischen den Frames 1610 und 1650, zu sehen in Abb. 7.9(d) und (g). Dies tritt in dem Moment auf, in welchem die Person aus dem Verdeckungsvolumen hervorkommt und sich wieder aufrichtet.

Die Betrachtung von Abb. 7.9 lässt weiterhin eine Reduktion der Häufigkeit an *occluded*-Voxeln mit zunehmender Bestrafung der *filled*-Voxel erkennen. Dies gilt sowohl für die einfache lineare Gewichtung als auch für die zusätzliche nichtlineare Verstärkung. Fehlt die Bestrafung wie in den Diagrammen von Abb. 7.9(a) und (b), so führt dies zu einer höheren Anzahl an *occluded*-Voxeln. Für den Zeitraum vor Frame 1550 Frames wird dies besonders sichtbar, was der Situation entspricht, bei der die Person um den Tisch herum läuft. Dieser Effekt lässt sich auf die Lage der *occluded*- und *filled*-Voxel zurückführen. Unter der Tischplatte befinden sich die *occluded*-Voxel. Der Tisch selbst besteht aus *filled*-Voxeln. Voxel dieser Klassen sind meist räumlich miteinander verbunden. Durch die positive Gewichtung mit $\psi = 0{,}2$ wird der Filter zwar auf die *occluded*-Voxel „gezogen“, gleichzeitig werden die *filled*-Voxel jedoch bestraft. Dadurch wirkt der Tisch als abstoßende Kraft auf den Filter und dies umso stärker, je mehr *filled-Voxel* sich in den Ellipsoiden befinden und je höher der Wert des gewählten Exponenten a ist. Umgekehrt nehmen die Häufigkeiten der *occluded*- und der *filled*-Voxel im Schwerpunktellipsoid mit der Erhöhung von ψ zu (vgl. Abbildungen 9.29 bis 9.33). Dennoch ist dieser Effekt auch bei $\psi = 0{,}5$ gering, wenn die *filled-Voxel* stark bestraft werden, wie zum Beispiel mit $a = 32$. Der Abstoßungseffekt überwiegt dann. Beispielsweise für die Frames < 1500 diffundiert der Filter nicht verfrüht in das Verdeckungsvolumen unter den Tisch, so wie ohne die Bestrafung der *filled*-Voxel im vorherigen Experiment für $\psi = 0{,}5$.

Ein weiterer Effekt verstärkt sich mit zunehmender Bestrafung der *filled*-Voxel in der Teilsequenz A. Die Häufigkeit der *outside*-Voxel steigt bei der linearen Gewichtung sichtbar zwischen den Frames 1550 und 1610 an, zu sehen in Abb. 7.9(c), (d) und (e). Die Erhöhung der Anzahl dieser Voxel entsteht während der Bewegung der Person unter den Tisch. Der Filter weicht dabei in Bereiche aus, die weniger stark bestraft werden als der Tisch, oder die belohnt werden. Dazu zählt der nicht modellierte Boden, dessen *outside*-Voxel mit 0 gewichtet werden. Bei der nichtlinearen Verstärkung bewegt

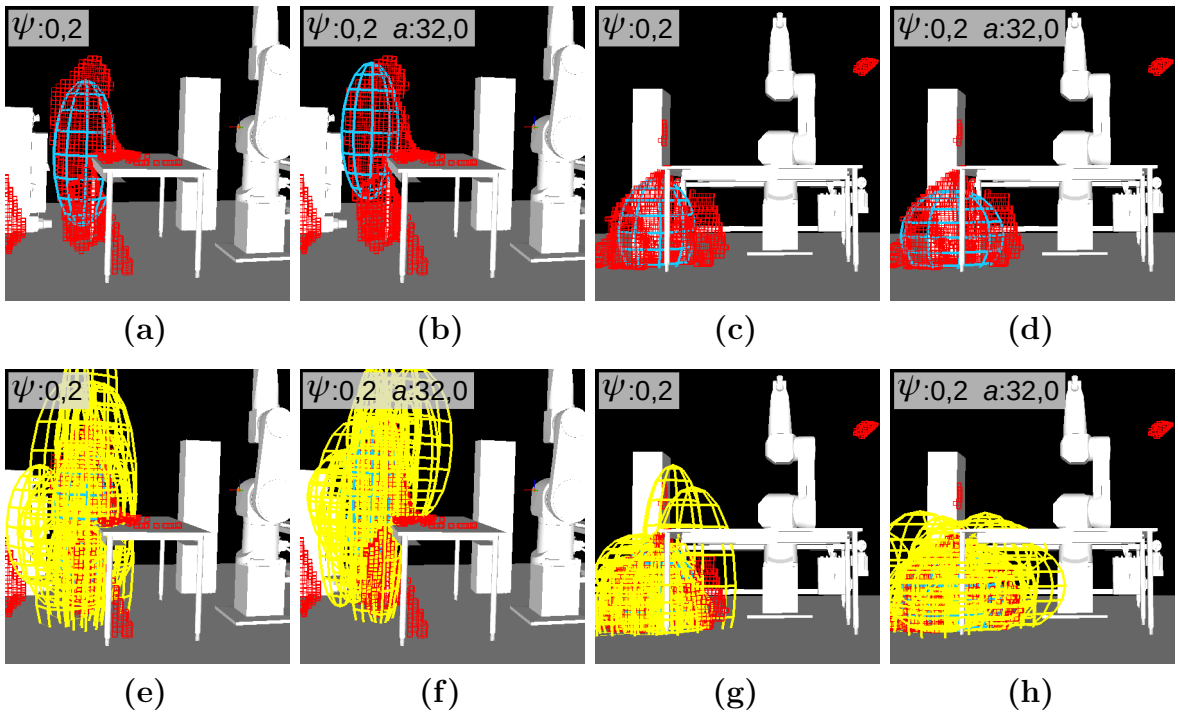


Abb. 7.10: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}2$ und nichtlinearer Verstärkung. Bestrafung der *filled*-Voxel mit $a = 32$ in (b), (d), (f) und (h). Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (hellblau), jedes 25. Partikelellipsoid der 500 Partikel (gelb) sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 2. Zu sehen sind Frame 1518 in (a), (b), (e) und (f) sowie Frame 1567 in (c), (d), (g) und (h).

sich das Schwerpunktellipsoid jedoch auch ohne Bestrafung der *filled*-Voxel in den Boden. Durch die Bevorzugung der *unknown*-Voxel verändern die Partikelellipsoide anscheinend ihre Form nicht schnell genug und weichen dann in Richtung Boden aus, was auch das resultierende Schwerpunktellipsoid widerspiegelt. Eine Ausnahme bildet dabei die gleichzeitige Bestrafung der *filled*-Voxel mit $a = 2$. Hierbei folgt der Filter für unterschiedliche Seeds dem Objekt gut unter den Tisch.

Für den letzten der fünf Voxelzustände wird nun noch die Auswirkung des bestrafenden Terms auf die Häufigkeiten der *filled*-Voxel selbst betrachtet. Die Anzahl der *filled*-Voxel in Teilsequenz A ist recht gering, weil der Tisch ein geringes Volumen einnimmt. Dennoch kann insbesondere zu Beginn der Sequenz (vor Frame 1550) bei Betrachtung der verschiedenen Diagramme in Abb. 7.9 eine Verringerung der Voxelanzahl mit steigender Bestrafung festgestellt werden. Dies betrifft sowohl die lineare Gewichtung in Abb. 7.9(c), (d) und (e) als auch die entsprechende nichtlineare Verstärkung in Abb. 7.9(f), (g) und (h). Je größer die Bestrafung ist, desto weniger *filled*-Voxel befinden sich in den Schwerpunktellipsoiden. Dies bestätigt das gewünschte Verhalten, welches mit der Bestrafung der *filled*-Voxel erzielt werden soll.

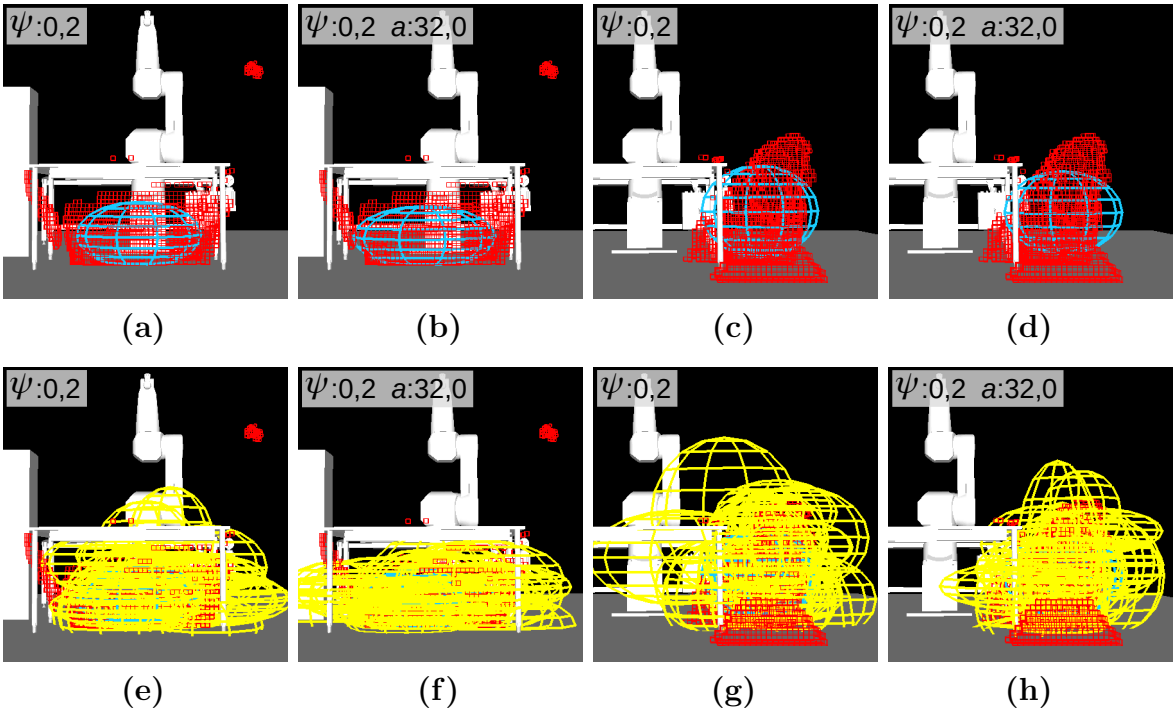


Abb. 7.11: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}2$ und nichtlinearer Verstärkung. Bestrafung der *filled*-Voxel mit $a = 32$ in (b), (d), (f) und (h). Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (hellblau), jedes 25. Partikelellipsoid der 500 Partikel (gelb) sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 2. Zu sehen sind Frame 1596 in (a), (b), (e) und (f) sowie Frame 1629 in (c), (d), (g) und (h).

Nach Auswertung der Diagramme in Abb. 7.9 sollen nun zugehörige Visualisierungen diskutiert werden. In den Abbildungen 7.10 und 7.11 sind Bilder verschiedener Frames von der nichtlinearen Verstärkung (Gain) für $\psi = 0{,}2$ dargestellt. Die Schwerpunktellipsoide sind hellblau eingezeichnet. Die gestreuten Ellipsoide sind in gelb gehalten, wobei von den 500 verwendeten Partikeln nur jedes 25. Ellipsoid eingezeichnet ist. Bilder von der nichtlinearen Verstärkung in (a), (c), (e) und (g) werden einer Gewichtung mit zusätzlicher Bestrafung der *filled*-Voxel in (b),(d),(f) und (h) gegenübergestellt. Der gewählte Exponent a liegt bei einem Wert von 32. Zu sehen sind die Ergebnisse von Seed 2.

In Frame 1518 der Abb. 7.10(b) wirkt die Tischplatte als Barriere und der Filter diffundiert weniger weit in die Tischplatte hinein als in (a). Das Schwerpunktellipsoid in (b) liegt auch etwas höher in den *unknown*-Voxeln. Die Betrachtung der zugehörigen Partikelellipsoide in Abb. 7.10(f) zeigt das Verhalten des Filters noch genauer, da die Partikel den Tisch im Vergleich zu (e) nur wenig schneiden. In Frame 1567 der Abb. 7.10(d), bei welchem sich die Person in das Verdeckungsvolumen des Tischs begibt, ändern die Partikel ihre Form größtenteils, bevor sie in das Verdeckungsvolumen diffundieren. Sie gehen dabei auch in den Boden. Die Anpassung der Partikel unter dem

Tisch erscheint dabei etwas besser in (h) als bei fehlender Bestrafung der *filled*-Voxel (g).

In Abb. 7.11 werden die Frames 1596 und 1629 gezeigt. Unter dem Tisch ragt der Filter mit Bestrafung der *filled*-Voxel in Abb. 7.11(b) insgesamt etwas weniger weit in die *occluded*-Voxel hinein als ohne (a). Der Filter befindet sich also weiter von der Tischplatte entfernt. Auch die einzelnen Ellipsoide in (f) lassen erkennen, dass sie mehr Abstand zur Tischplatte halten und deren Form besser an den Bereich unter den Tisch angepasst ist als bei fehlender Bestrafung in (e). Beim Verlassen des Verdeckungsvolumens ist ebenfalls erkennbar, dass der Filter der Tischplatte ausweicht und erst hinter dem Tisch wieder stärker in die vertikale Ausrichtung gelangt. Ein Vergleich der Schwerpunktellipsoide für Frame 1629 in Abb. 7.11(c) und (d) lässt dies nicht erkennen, jedoch die zugehörigen Ellipsoide in Abb. 7.11(h) im Vergleich zu (g).

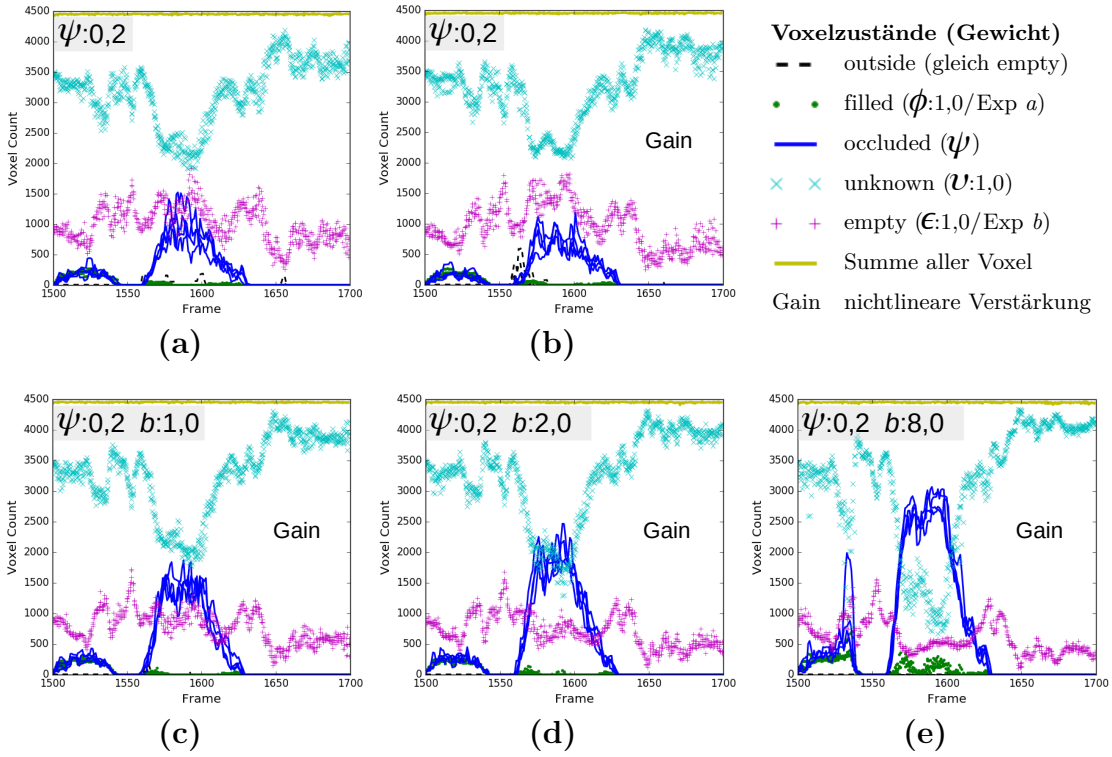


Abb. 7.12: Bestrafung von *empty*-Voxeln mit variierendem Exponenten b des Bestrafungsterms und nichtlinearer Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0{,}2$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

7.5.2 Bestrafung der *empty*-Voxel

In diesem Abschnitt wird die Bestrafung der *empty*-Voxel E_k^j innerhalb des Partikelellipsoids j nach Formel 7.5 untersucht, zunächst ohne gleichzeitige Bestrafung der *filled*-Voxel. Die *outside*-Voxel erhalten die gleiche Bestrafung wie die *empty*-Voxel. Sie werden in den Diagrammen separat gezählt, sind in den Formeln jedoch nicht explizit ausgewiesen, sondern in der Voxelmenge E_k^j jedes Partikels enthalten.

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{e^{\frac{|\Upsilon_k^j|}{|M_k^j|}}}{e} \cdot \frac{\left(\frac{|E_k^j|}{|M_k^j|} - s\right)^b}{(-s)^b} \quad \text{mit} \quad s = 1, \quad 1 \leq b, \quad b \in \mathbb{N} \tag{7.5}$$

In Abb. 7.12 sind Diagramme von Gewichtungen mit $\psi = 0{,}2$ zu sehen. „Gain“ steht wieder für die nichtlineare Verstärkung der *unknown*-Voxel. In Abb. 7.12(a) und (b) sind als Referenz wiederholt die Diagramme der Gewichtungen zu sehen, bei denen keine Voxelbestrafung erfolgt.

Eine Bestrafung der *empty*-Voxel führt im Mittel über alle Frames zu einer verrin-

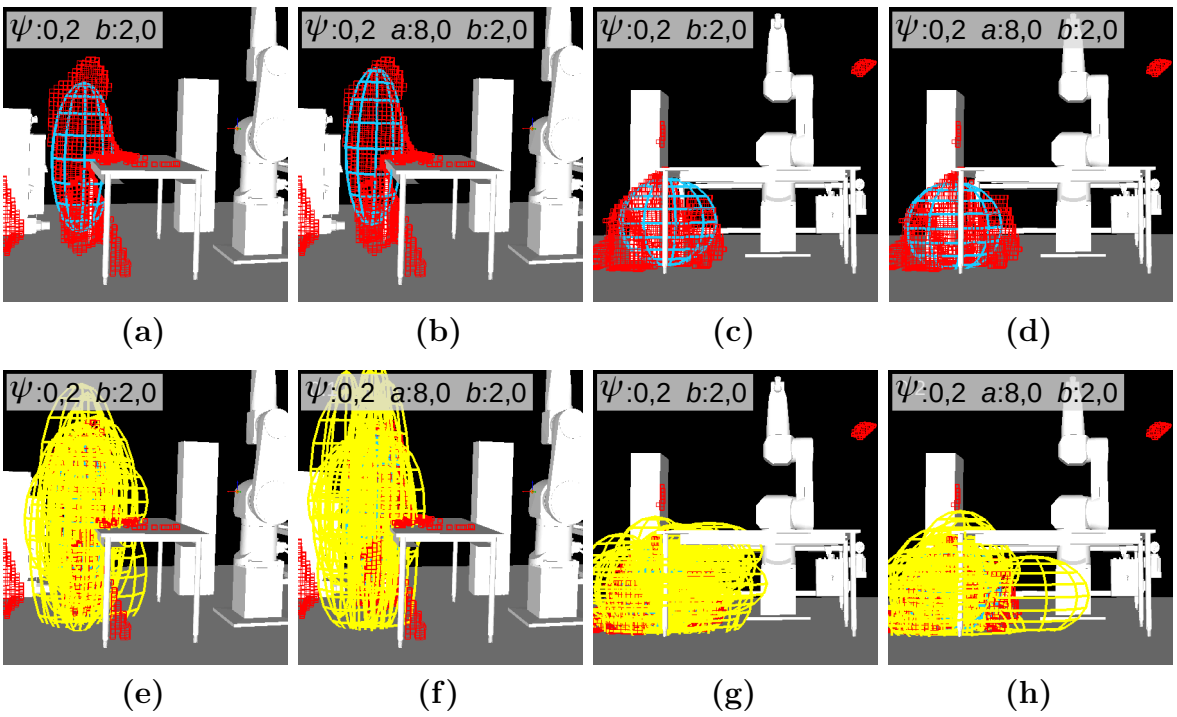


Abb. 7.13: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}2$ und nichtlinearer Verstärkung. Bestrafung der *empty*-Voxel mit $b = 2$. In (b), (d), (f) und (h) werden zusätzlich die *filled*-Voxel mit $a = 8$ bestraft. Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (hellblau), jedes 25. Partikelellipsoid der 500 Partikel (gelb) sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 2. Zu sehen sind Frame 1518 in (a), (b), (e) und (f) sowie Frame 1567 in (c), (d), (g) und (h).

gerten Häufigkeit der *empty*- und *outside*-Voxel innerhalb des Schwerpunktellipsoids. Dadurch wird implizit eine Verstärkung der *unknown*- und *occluded*-Voxel, aber auch der *filled*-Voxel herbeigeführt, wodurch sich die Partikel verstärkt auf diesen Voxeln platzieren. Konkret in der Teilsequenz A diffundiert der Filter unterhalb der Tischplatte etwas weniger weit in den Boden und dafür stärker auf die *unknown*-Voxel und in das Verdeckungsvolumen. Eine abstoßende Wirkung übt der Tisch hierbei nicht aus. In Abb. 7.12(c), (d) und (e) steigt erkennbar die Häufigkeit der *occluded*-Voxel gegenüber der fehlenden Bestrafung von *empty*-Voxeln in Abb. 7.12(a) und (b). In (e) sinkt aber gleichzeitig die Häufigkeit der *unknown*-Voxel, was unerwünscht ist. Eine Bestrafung mit $b = 2$ ist demzufolge grenzwertig. Ein noch höherer Exponent von $b >= 8$ führte in den Experimenten häufig dazu, dass der Filter das Verdeckungsvolumen nicht mehr verlässt. Bei einer zusätzlichen Bestrafung der *filled*-Voxel (vgl. Abb. 7.15) ist der Anteil der *occluded*-Voxel sichtbar geringer, weil der Tisch als „abstoßende Kraft“ wirkt. Es ist auch eine positiv zu bewertende durchschnittliche Erhöhung der Häufigkeit der *unknown*-Voxel erkennbar. Betrachtet man die Visualisierungen der Abbildungen 7.13 und 7.14, so lassen sich kleine Unterschiede mit und ohne zusätzlicher Bestrafung der *filled*-Voxel

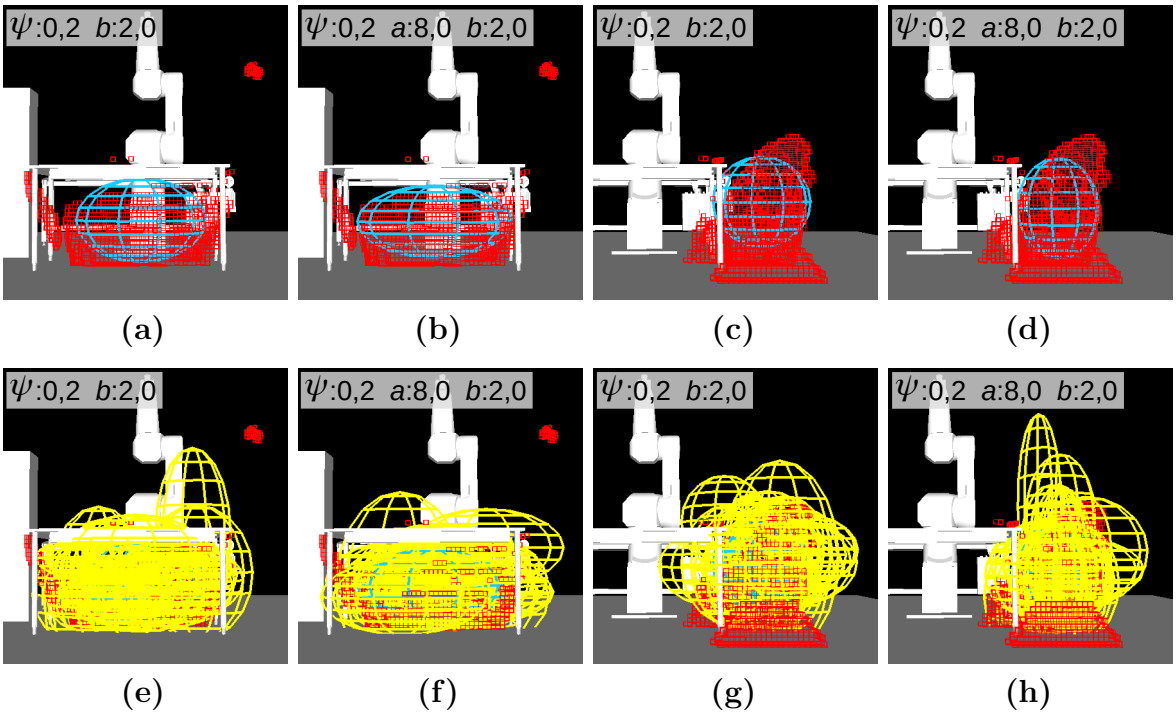


Abb. 7.14: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}2$ und nichtlinearer Verstärkung. Bestrafung der *empty*-Voxel mit $b = 2$. In (b), (d), (f) und (h) werden zusätzlich die *filled*-Voxel mit $a = 8$ bestraft. Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (hellblau), jedes 25. Partikelellipsoid der 500 Partikel (gelb) sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 2. Zu sehen sind Frame 1596 in (a), (b), (e) und (f) sowie Frame 1629 in (c), (d), (g) und (h).

erkennen. Detaillierter wird diese Gewichtungskombination im folgenden Abschnitt behandelt.

7.5.3 Bestrafung der *empty*- und *filled*-Voxel

In diesem Experiment werden sowohl die *filled*-Voxel als auch die *empty*-Voxel (inklusive der *outside*-Voxel) bestraft. Die Häufigkeiten beider Voxelzustände in den Schwerpunktellipsoiden sollen möglichst gering sein. Die Bestrafung wird durch die beiden Terme auf der rechten Seite von Formel (7.6) vorgenommen.

$$w_k^j = \frac{(\upsilon \cdot |\Upsilon_k^j| + \psi \cdot |\Psi_k^j|)}{|M_k^j|} \cdot \frac{e^{\frac{|\Upsilon_k^j|}{|M_k^j|}}}{e} \cdot \frac{\left(\frac{|\Phi_k^j|}{|M_k^j|} - s\right)^a}{(-s)^a} \cdot \frac{\left(\frac{|\mathrm{E}_k^j|}{|M_k^j|} - s\right)^b}{(-s)^b} \tag{7.6}$$

$$\text{mit} \quad s = 1, \quad 1 \leq a, \quad a \in \mathbb{N}, \quad 1 \leq b, \quad b \in \mathbb{N}$$

Erreicht werden soll durch diese Vorgehensweise eine schärfere Fokussierung des Filters auf der Messung (den *unknown*-Voxeln) bei gleichzeitigem Ausweichen der statischen Objekte (den *filled*-Voxeln). Experimentell zeigte sich, dass die Bedingung $b <= a$ gelten sollte, da es einem Filter andernfalls erschwert wird, (größere) Verdeckungsvolumina nach deren Betreten wieder zu verlassen.

In Abb. 7.15 sind Diagramme für $\psi = 0{,}2$ dargestellt. Als Referenzen sind die lineare Gewichtung mit und ohne nichtlineare Verstärkung in (a) und (b) zu sehen sowie die Ergebnisse der ausschließlichen Bestrafung der *filled*-Voxel in (c) bis (e). Diesen wird die erweiterte Gewichtungsfunktion aus Formel 7.6 gegenübergestellt, bei der sowohl die *filled*- als auch die *empty*-Voxel in (f) bis (k) bestraft werden. Für die *filled*-Voxel werden die Exponenten $a = 2$, 8, 32 und für die *empty*-Voxel die Exponenten $b = 1$ und $b = 2$ betrachtet. Für diese Diagramme lässt sich Folgendes feststellen: Deutlich sichtbar wachsen die Häufigkeiten der *occluded*-Voxel zwischen den Frames 1550 bis 1625 mit Bestrafung der *empty*-Voxel und zwar umso stärker, je höher der Wert des Exponenten b ist. Damit einher geht die gewünschte Reduktion der *empty*-Voxel. Zu Beginn der Teilsequenz A (bis Frame 1550) sind die Ergebnisse aller Diagramme recht ähnlich, außer bei den stärkeren Bestrafungen der *filled*-Voxel mit einem Exponenten von 32 in Abb. 7.15(e), (h) und (k). Die Anzahl der *empty*-Voxel ist hier im Vergleich zu den anderen Diagrammen etwas höher. Ab Frame 1630 sind kaum Unterschiede zwischen den verschiedenen Gewichtungen feststellbar.

Die Betrachtung der *filled*-Voxel zeigt keinen markanten Unterschied bei der kombinierten oder der separaten Anwendung der Bestrafungsterme. Das Gleichbleiben der

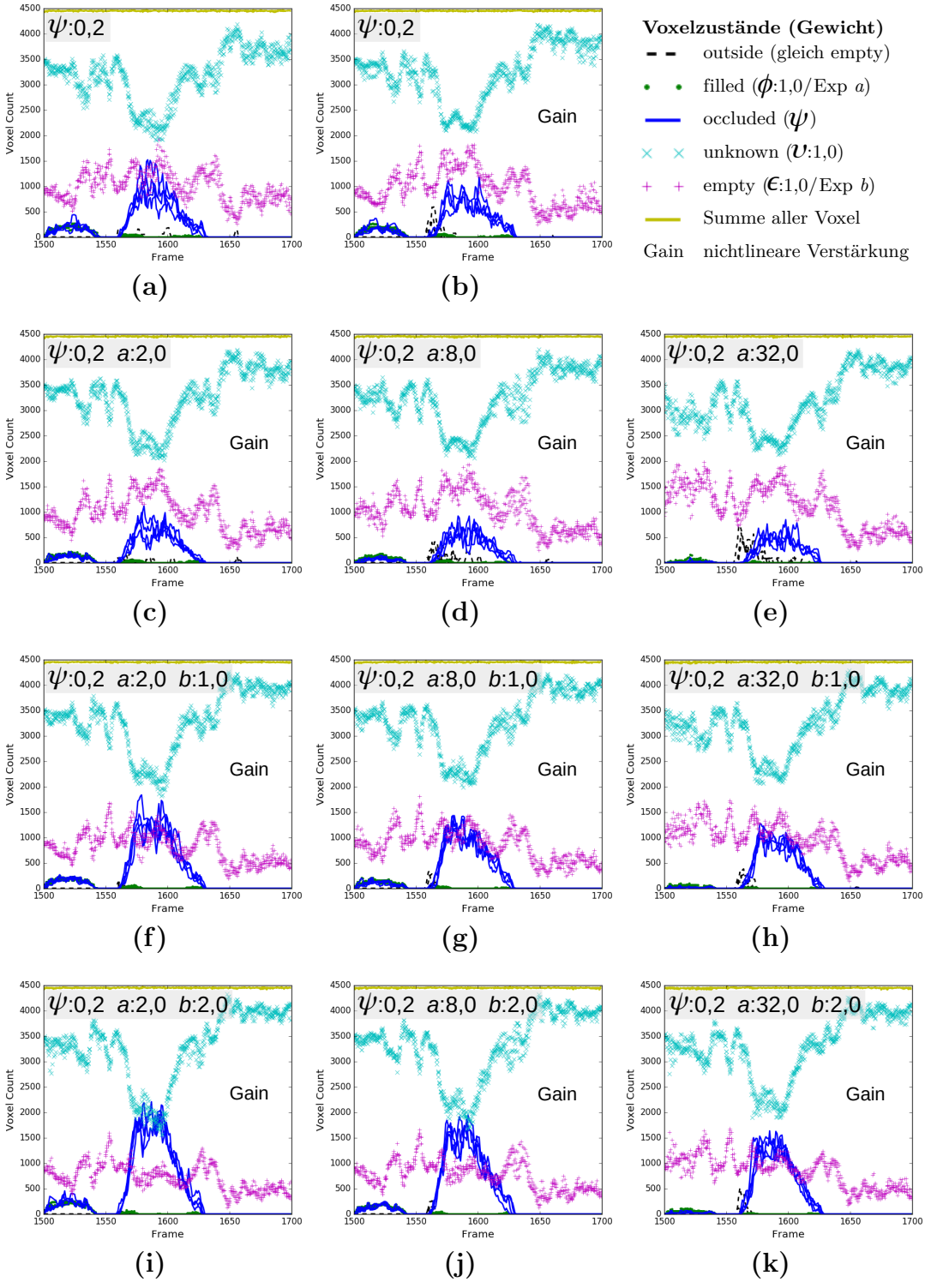


Abb. 7.15: Bestrafung von *filled*- und *empty*-Voxeln mit variierenden Exponenten a und b der Bestrafungsterme, mit nichtlinearer Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung der *filled*-Voxel gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0{,}2$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

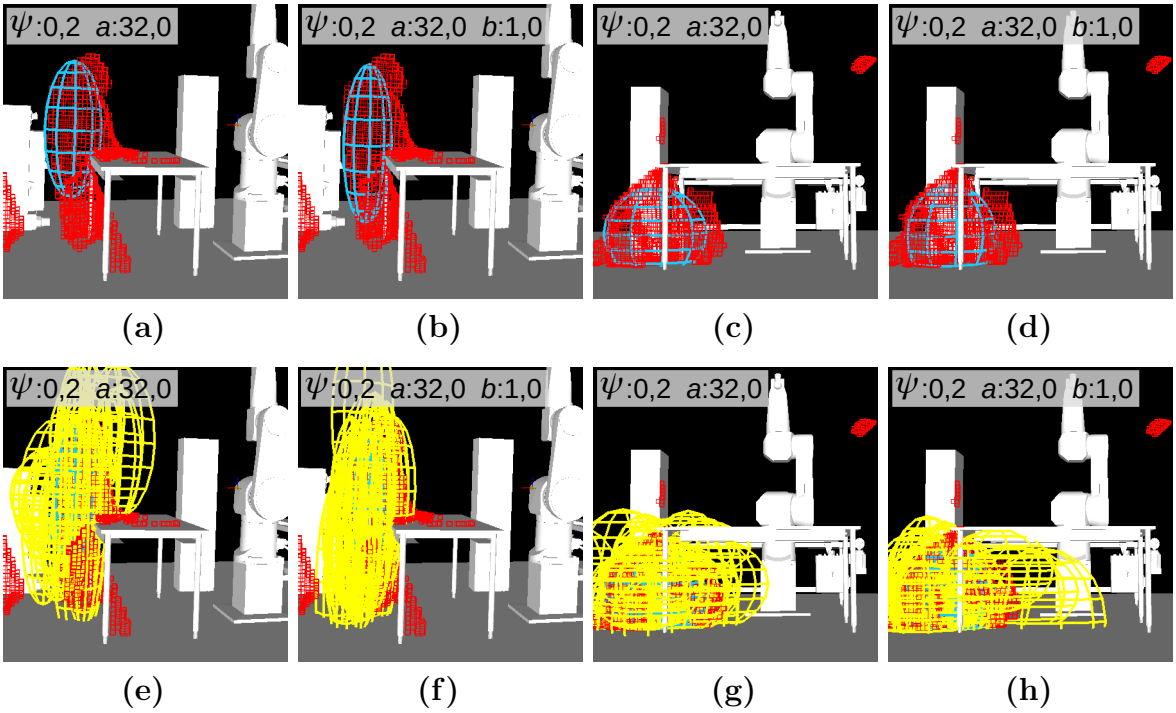


Abb. 7.16: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}2$ und nichtlinearer Verstärkung. Bestrafung der *filled*-Voxel mit $a = 32$. In (b), (d), (f) und (h) werden zusätzlich die *empty*-Voxel mit $b = 1$ bestraft. Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (hellblau), jedes 25. Partikelellipsoid der 500 Partikel (gelb) sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 2. Zu sehen sind Frame 1518 in (a), (b), (e) und (f) sowie Frame 1567 in (c), (d), (g) und (h).

unknown-Voxel ist erstrebenswert. Die Bestrafung der *empty*-Voxel mit $b = 1$ in Zeile 3 der Abb. 7.15 liefert hierfür akzeptable Ergebnisse. Für $b = 2$ in Zeile 4 sinkt jedoch die Häufigkeit der *unknown*-Voxel während der Objektverdeckung unterhalb des Tischs stark ab, zugunsten der *occluded*-Voxel. Die Häufigkeit der *outside*-Voxel, die zusammen mit den *empty*-Voxeln bestraft werden, reduziert sich zu Beginn der Objektverdeckung (ab ca. Frame 1550). Dies ist positiv zu bewerteten.

Abschließend werden noch Bilder ausgewertet, die den Unterschied zwischen einer Gewichtung mit und ohne Bestrafung der *empty*-Voxel mit $b = 1$ zeigen. Betrachtet wird Frame 1518 in Abb. 7.16. Das Schwerpunktellipsoid ist in (b) etwas gestreckter entlang der vertikalen Achse als in (a). Auch die dargestellten Partikel in (f) im Vergleich zu (e) lassen darauf schließen, dass sich die Streuung stärker an der vertikalen Achse orientiert und die Ellipsoide der Partikel mehr *unknown*-Voxel beinhalten. Die Partikel bei fehlender Bestrafung der *empty*-Voxel streuen anscheinend über dem Tisch etwas stärker in den leeren Bereich (e). Zu Beginn der Objektverdeckung ab Frame 1567 geht das Schwerpunktellipsoid in (d) verzögert in die horizontale Ausrichtung im Vergleich zu (c). Die zugehörigen Partikelellipsoide bleiben ebenso etwas stärker unter dem Tisch

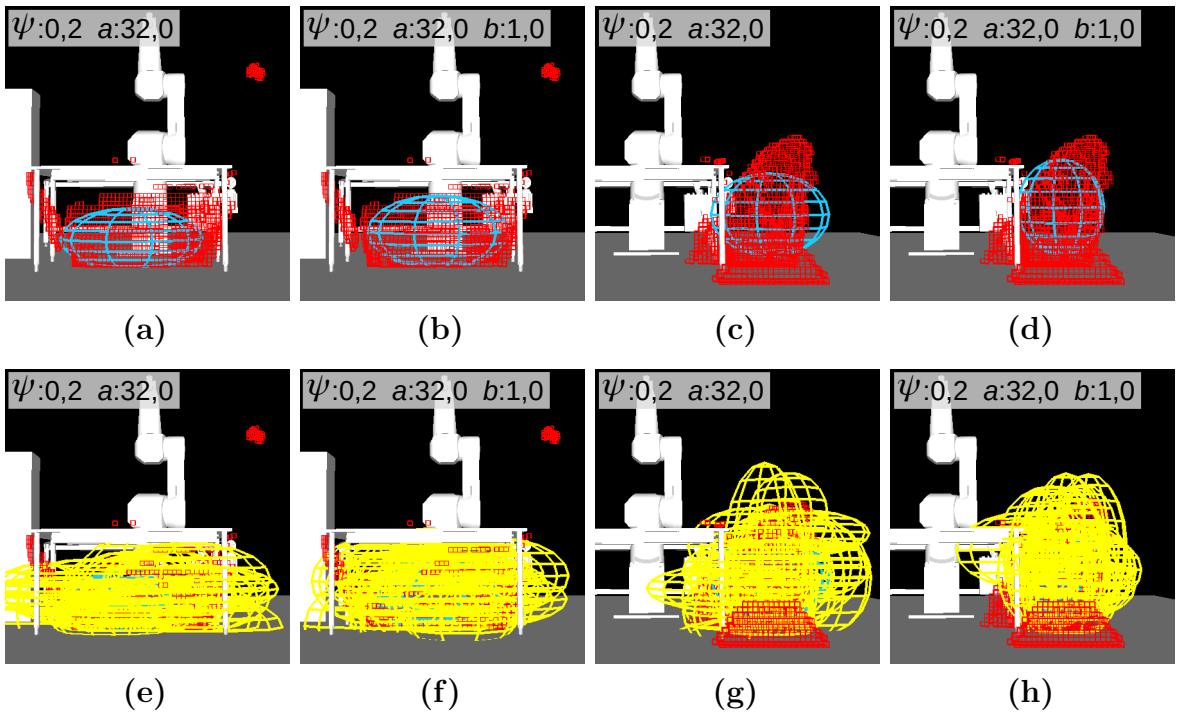


Abb. 7.17: Virtuelle Ansichten des Trackings für eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}2$ und nichtlinearer Verstärkung. Bestrafung der *filled*-Voxel mit $a = 32$. In (b), (d), (f) und (h) werden zusätzlich die *empty*-Voxel mit $b = 1$ bestraft. Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide des Partikelfilters (hellblau), jedes 25. Partikelellipsoid der 500 Partikel (gelb) sowie die statischen Objekte (weiß). Die Filterinitialisierung erfolgte mit Seed 2. Zu sehen sind Frame 1596 in (a), (b), (e) und (f) sowie Frame 1629 in (c), (d), (g) und (h).

in (h) und gehen etwas weniger auf die leeren Voxel als in (g). Dies sind jedoch nur Nuancen in diesem spezifischen Durchlauf.

In Abb. 7.17 befindet sich die Person in Frame 1596 unter dem Tisch. Das Schwerpunktellipsoid bei Bestrafung der *empty*-Voxel in Abb. 7.17(b) geht weniger weit in den Boden und befindet sich näher an der Tischplatte als bei ausbleibender Bestrafung dieser Voxelzustände in (a). Die dargestellten Partikel lassen ebenfalls erahnen, dass in (f) eine leichte Ausrichtung des Filters zur Tischplatte hin besteht im Vergleich zu (e). In Frame 1629 ist der Unterschied zwischen den beiden Gewichtungsvarianten am deutlichsten zu erkennen. Beim Heraustreten aus dem Verdeckungsvolumen führt die Bestrafung der *empty*-Voxel zu einer schnelleren vertikalen Ausrichtung des Filters in Abb. 7.17(d) im Vergleich zu (e). Die einzelnen Partikel zeigen dabei in (h) eine stärkere Positionierung auf den Messdaten. Sie liegen weniger stark auf leeren Voxeln als in (g).

7.5.4 Zusammenfassung

In Teilsequenz A führt eine Bestrafung der *filled*-Voxel zur Wirkung des Tischs als „abstoßende Kraft“, wodurch sich der Abstand zwischen Schwerpunktellipsoid und Tisch vergrößert. Der Tracking-Verlauf und die konkrete Objektlokalisierung wird dabei in der Teilsequenz A nur gering gestört. Die Bestrafung der *empty*-Voxel (und *outside*-Voxel) führt zur „Meidung“ des Filters von Bereichen, die leer sind oder außerhalb des Überwachungsraums liegen.

Die Kombination beider Bestrafungsterme nach Formel 7.6 soll eine schärfere Fokussierung des Filters auf die Messung (den *unknown*-Voxeln), bei gleichzeitigem Ausweichen der statischen Objekte (den *filled*-Voxeln) erbringen. Bei geeigneter Parametrisierung (z. B. $\psi = 0{,}2$; $a = 8$; $b = 1$) führt dies in dem betrachteten Beispiel zu einem positiven Tracking-Verlauf mit folgenden Tendenzen: Außerhalb des Verdeckungsvolumens nimmt das Schwerpunktellipsoid eine gestreckte Form an und diffundiert durch die Bestrafung der *filled*-Voxel nicht vorzeitig unter den Tisch. Zu Beginn der Objektverdeckung unterhalb des Tischs diffundiert der Filter weniger in den Boden hinein aufgrund der Bestrafung der *outside*-Voxel (in gleichem Maße wie die *empty*-Voxel). Je stärker die *empty*-Voxel bestraft werden, desto mehr platziert sich der Filter während der Objektverdeckung auf den *occluded*-Voxeln unter dem Tisch. Im Anschluss an die Objektverdeckung liegt der Filter wieder in gestreckter Form auf den *unknown*-Voxeln (der Person).

Erkennbar ergeben sich bei der Gewichtung der Voxelzustände Ziele, die einander entgegen wirken können. Dies entsteht aufgrund der räumlichen Nähe und Nachbarschaften, die Voxel unterschiedlicher Voxelklassen zueinander aufweisen, und spiegelt sich in der resultierenden Filterbewegung wider. Die Werte der Gewichtungsparameter müssen deshalb entsprechend ausbalanciert sein. So betrifft dies die Bestärkung der *occluded*-Voxel bei gleichzeitiger Bestrafung der *filled*-Voxel, die räumlich zusammenhängen. Auch die Bestrafung der *empty*-Voxel (und *outside*-Voxel) kann konträr zur Bestrafung der *filled*-Voxel wirken. Die Bestrafung erst genannter Voxelklassen führt zu einer Bevorzugung der anderen drei Voxelzustände mit entsprechender Filterbewegung zur Tischplatte und zum Verdeckungsvolumen hin, was sich deutlich für $b > 1$ zeigt. Hingegen resultiert die gleichzeitige Bestrafung der *filled*-Voxel und damit des Tischs wieder in einer entgegengesetzt gerichteten Filterbewegung.

Ein weiterer Aspekt ist die bestehende Abhängigkeit der Wirkung des gewählten Exponenten a im Bestrafungsterm der *filled*-Voxel von der konkreten Anzahl im Raum existierender Voxel dieses Zustands, die sich aus dem Volumen der statischen Objekte ergibt. Dies wurde bereits in Abschnitt 6.4 erläutert. Der betrachtete Tisch ist flach und

hat schmale Beine, wodurch die Anzahl der *filled*-Voxel, welche in den Ellipsoiden, die den Tisch schneiden, vorkommen, begrenzt ist. So zeigte sich für die Teilsequenz A, dass auch eine recht hohe Bestrafung mit $a = 32$ zu noch guten Tracking-Ergebnissen führt. Hingegen wird für die gleiche Funktion bei den *empty*-Voxeln ein Wert von $a \leq 2$ verwendet. Andernfalls würde der Filter zu stark in das Verdeckungsvolumen diffundieren, weg von den gegebenen *unknown*-Voxeln. Dies kann ab $b \geq 8$ (was auch von den anderen Parametern noch abhängt) einen Verbleib des Filters in dem Verdeckungsvolumen bewirken. Dies hängt ebenfalls mit den Häufigkeiten der *empty*-Voxel in den Ellipsoiden zusammen, jedoch auch mit den Abhängigkeiten der Voxelzustände voneinander. So führt eine Bestrafung der *emtpy*-Voxel implizit zu einer Verstärkung der anderen nicht bestraften Voxelzustände. Bei größeren Verdeckungsvolumina kann dies auch einen entsprechend großen Effekt bei der Filterbewegung auf die *occluded*-Voxel hervorrufen, was im Gegensatz zu dem Ziel der nichtlinearen Verstärkung des additiven Terms steht.

In Abschnitt 7.6 wird das Filterverhalten in weiteren Situationen betrachtet. Als Alternative zur Bestrafung der *filled*-Voxel wird ein Kollisionstest untersucht, mit welchem verhindert werden kann, dass ein Filter durch die statischen Objekte vollkommen hindurch diffundiert.

7.6 Gesamtevaluierung mit zwei Personen

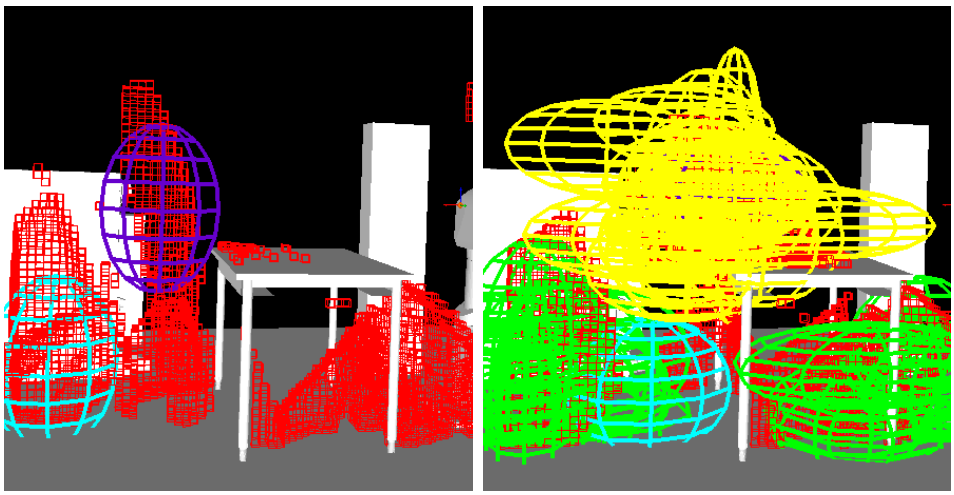
In den vorangegangenen Untersuchungen dieses Kapitels wurden die einzelnen Parameter und Bestandteile der Likelihood-Funktion anhand der Teilsequenz A analysiert. Getrackt wurde eine einzelne Person. Die Ergebnisse sollen nun in weiteren Situationen validiert werden. Dazu werden zwei Personen verfolgt, die sich in der Roboterarbeitszelle bewegen. Zunächst wird eine spezielle herausfordernde Situation (Teilsequenz B) im Detail betrachtet, bevor die Tracking-Ergebnisse auch von anderen Teilsequenzen ausgewertet werden.

7.6.1 Tracking bei Rekonstruktionsartefakten

In der Teilsequenz B, zu sehen in Abb. 7.18, klettert die rote Person über den Tisch, während sich die grüne Person seitlich unter den Tisch in ein Verdeckungsvolumen begibt und auf der anderen Seite wieder hervorkommt. Herausfordernd an dieser Sequenz ist das Verfolgen der grünen Person. Während sich diese in dem Verdeckungsvolumen befindet, erzeugt die rote Person bei ihrer Bewegung über den Tisch ein Rekonstruktionsartefakt am Boden (bestehend aus *unknown*-Voxeln) in der Nähe des Tischs (vgl. Abb. 7.19(a), links). Dieses kann den Filter der grünen Person auf sich „ziehen“ und einen Tracking-

Abb. 7.18: Bilder der Teilsequenz B, zu betrachten von links oben nach rechts unten. Die rote Person klettert über den Tisch. Die grüne Person begibt sich währenddessen unter den Tisch und wird dabei partiell für das Kamerasystem verdeckt.

Verlust auslösen. In Abb. 7.19(a) ist dies visualisiert. Die Schwerpunktellipsoide beider Filter sind zu sehen (violett und hellblau). Das hellblaue Ellipsoid befindet sich auf dem rekonstruierten Artefakt, anstatt auf dem Voxelsegment, das tatsächlich zur Person gehört (rote Voxel rechts im Bild). In Abb. 7.19(b) teilen sich die Partikelellipsoide des zugehörigen Filters (grün) zwischen den Voxelsegmenten des Artefakts und der realen Messung auf. Es werden zwei Hypothesen mit dem Partikelfilter verfolgt. Die



(a) (b)

Abb. 7.19: Störendes Artefakt beim Tracking in Teilsequenz B, auf dem fälschlicherweise das Schwerpunktellipsoid (hellblau) des Filters der grünen Person liegt. Zu sehen sind die *unknown*-Voxel (rot), Schwerpunktellipsoide zweier Partikelfilter (hellblau und violett), jedes 25. Partikelellipsoid der 500 Partikel beider Filter (gelb und grün), sowie die statischen Objekte (weiß). Die Ellipsoide eines Filters (grün) sind aufgeteilt auf zwei Voxelsegmente und verfolgen damit zwei verschiedene Hypothesen.

gelben Partikelellipsoide in Abb. 7.19(b) gehören zum Filter, der die rote Person trackt. Die rote Person wird in den meisten Durchgängen gut verfolgt. Falls nicht, geht damit auch gleichzeitig ein Tracking-Verlust der grünen Person einher, weshalb der Fokus der Betrachtungen auf die grüne Person gelegt wird.

Für ein erfolgreiches Mehrpersonen-Tracking muss zu jedem Zeitpunkt k das Datenassoziationsproblem gelöst werden: Es wird eine Zuordnung der Messungen (der einzelnen *unknown*-Voxel) zu den bestehenden Filtern benötigt, die jedoch bei dem gegebenen Verfahren nicht eindeutig vorgenommen werden kann. Ohne eine Funktionalität zur Lösung des Datenassoziationsproblems können sich Filter leicht auf denselben Teil einer Messung positionieren, wenn sich dynamische Objekte zu nahe kommen (Merge-Ereignis). Daraus ergibt sich häufig die Verfolgung ein und desselben Objekts durch mehrere Filter, während der Track eines anderen Objekts verloren wird. Wie in Abschnitt 6.3.2 beschrieben, wird zur Lösung des Problems eine Blocking-Methode eingesetzt. Dabei werden jeweils die Abstände zwischen den Partikeln des betrachteten Filters und den besten Schätzungen des letzten Zeitschritts aller anderen Filter berechnet. Geringe Abstände werden bestraft, wodurch sich entsprechend kleine Likelihood-Gewichte für die Partikel ergeben, die anderen Filtern zu nahe kommen.

Für die Teilsequenz B aus Abb. 7.18 werden zwei Filter initialisiert (vgl. Abschnitt 6.3.3). Untersucht wird der Tracking-Erfolg für die grüne Person anhand der Position des Schwerpunktellipsoids, die visuell in den Bildern des 3D-Viewers bewertet wird. Die

nichtlineare Verstärkung der *unknown*-Voxel (Gain) kommt in jedem Durchgang zum Einsatz ebenso wie die beschriebene Blocking-Methode. Die Ergebnisse der Beobachtungen zu verschiedenen Parametrisierungen sind in den Tabellen 7.20 und 7.21 eingetragen und sollen nun im Detail beschrieben werden. Dabei werden die Durchgänge jedes Parametersets wieder mit einem pseudorandomisierten Zufallsgenerator mit vier festen Startwerten (Seeds) initialisiert, um eine bessere Vergleichbarkeit der Ergebnisse verschiedener Parametrisierungen zu ermöglichen. Ist in der Tabelle ein Häkchen vermerkt (✓), so trackt der Filter die Person recht gut; die visuelle Beurteilung ergibt eine ungefähre Übereinstimmung des Schwerpunktellipsoids mit der visuell geschätzten tatsächlichen Objektposition. Ein Objektverlust des Filters ist mit (✗) gekennzeichnet. Ist ein Kreis vermerkt (○), so zeigt dies eine temporär auftretende stärkere Abweichung des Schwerpunktellipsoids von der tatsächlichen Objektposition an, wobei das Objekt final nicht verloren wird.

Mit dem Partikelfilter wird zwar explizit ein Tracking-Verfahren eingesetzt, welches die Verfolgung mehrerer Hypothesen ermöglicht, um in mehrdeutigen Situationen das Objekt nicht zu verlieren. Dennoch kann die Abweichung des Schwerpunktellipsoids von der visuell geschätzten tatsächlichen Objektposition für die betrachteten Situationen als Maß eingesetzt werden, um die Tracking-Güte zu bewerten. Wenn die Partikel nicht gleichmäßig um das Objekt streuen, so wird dies negativ bewertet, da die Parametrisierung der Likelihood-Funktion in diesem Fall nicht zu einer approximierten unimodalen Dichte führt. In der Teilsequenz B tritt dies beispielsweise dann auf, wenn sich eine größere Partikelanzahl auf ein Artefakt außerhalb des Tischbereichs legt, obwohl sich die Person unter dem Tisch befindet. Im Effekt tritt dann eine deutliche Positionsabweichung zwischen Schwerpunktellipsoid und realer Objektposition zutage.

Partielle Objektverdeckung

Zunächst wird wieder das reale Verdeckungsvolumen aus Abb. 7.1(a) untersucht, das zu einer partiellen Objektverdeckung der Person führt. Anschließend wird im nächsten Abschnitt das, um die Voxel aus Abb. 7.1(c) erweiterte, Verdeckungsvolumen betrachtet, welches in Teilsequenz B eine vollständige Objektverdeckung herbeiführt.

In den ersten vier Zeilen von Tabelle 7.20 werden keine Bestrafungsterme angewandt. Die Zeile 1 entspricht dabei der Vorgehensweise von [Canton-Ferrer et al., 2011]: Bezüglich der geometrischen Einpassung erhalten nur die *unknown*-Voxel ein positives Gewicht. Die Voxel der anderen Voxelzustände werden nicht voneinander differenziert und mit 0 gewichtet. Diese Parametrisierung führt bereits zu einem Tracking-Erfolg, was damit begründet werden kann, dass das reale Verdeckungsvolumen den Filter nicht übermäßig stört. In Abbildung 7.22(a) ist zu sehen wie sich das Schwerpunktellipsoid unter dem

Zeile	Gewichte		Gain	Bestrafung		Block.-	Kollis.-	Seed Nummer			
	υ	ψ		a	b	methode	test	1	2	3	4
1	1	0,0	✓	-	-	✓	-	✓	✓	✓	✓
2	1	0,2	✓	-	-	✓	-	✓	✓	✓	✓
3	1	0,0	✓	-	-	✓	✓	○	✓	✓	✓
4	1	0,2	✓	-	-	✓	✓	○	○	✓	○
5	1	0,0	✓	2	-	✓	-	✓	✓	✓	✓
6	1	0,2	✓	2	-	✓	-	○	✓	✓	○
7	1	0,0	✓	2	-	✓	✓	✓	✓	✓	○
8	1	0,2	✓	2	-	✓	✓	✓	✓	X	✓
9	1	0,0	✓	8	-	✓	-	✓	○	○	✓
10	1	0,2	✓	8	-	✓	-	✓	✓	X	✓
11	1	0,0	✓	8	-	✓	✓	✓	○	✓	✓
12	1	0,2	✓	8	-	✓	✓	✓	○	✓	○
13	1	0,0	✓	32	-	✓	-	✓	✓	✓	○
14	1	0,2	✓	32	-	✓	-	✓	✓	○	X
15	1	0,0	✓	32	-	✓	✓	○	✓	○	○
16	1	0,2	✓	32	-	✓	✓	✓	X	X	✓
17	1	0,0	✓	-	1	✓	-	✓	✓	✓	✓
18	1	0,2	✓	-	1	✓	-	✓	✓	✓	✓
19	1	0,0	✓	-	1	✓	✓	✓	✓	✓	✓
20	1	0,2	✓	-	1	✓	✓	✓	✓	✓	✓
21	1	0,0	✓	-	2	✓	-	✓	✓	✓	✓
22	1	0,2	✓	-	2	✓	-	✓	✓	✓	✓
23	1	0,0	✓	-	2	✓	✓	✓	✓	✓	✓
24	1	0,2	✓	-	2	✓	✓	✓	✓	✓	✓
25	1	0,0	✓	-	8	✓	-	○	○	○	○
26	1	0,2	✓	-	8	✓	-	X	X	X	X
27	1	0,0	✓	-	8	✓	✓	○	X	X	○
28	1	0,2	✓	-	8	✓	✓	X	X	X	X

Abb. 7.20: Untersuchte Parametrisierungen für die Teilsequenz B. Die nichtlineare Verstärkung (Gain) der *unknown*-Voxel sowie die Blocking-Methode werden überall eingesetzt. Die Ergebnisse beziehen sich auf die Tracking-Güte der grünen Person, die sich unter den Tisch in das **reale Verdeckungsvolumen** begibt und **partiell verdeckt** wird. Die Position des Schwerpunktellipsoids wird für die Person visuell bewertet. Jeder Parametersatz wird mit einem pseudorandomisierten Zufallsgenerator mit vier festen Startwerten (Seeds) initialisiert. Ein Häkchen (✓) bedeutet, dass der Filter die grüne Person gut trackt. Die visuelle Beurteilung ergibt dabei eine ungefähre Übereinstimmung des Schwerpunktellipsoids mit der geschätzten tatsächlichen Objektposition über die betrachtete Sequenz hinweg. Verliert der Filter die Person mindestens einmal, so ist dies mit (X) gekennzeichnet. Ein Kreis ist eingetragen (○), wenn das Schwerpunktelliposid mindestens einmal temporär stärker von der tatsächlichen Position der Person abweicht, die Person aber nicht vollständig verloren wird.

Zeile	Gewichte		Gain	Bestrafung		Block.-	Kollis.-	Seed Nummer			
	υ	ψ		a	b	methode	test	1	2	3	4
29	1	0,0	✓	2	1	✓	-	✓	✓	✓	○
30	1	0,2	✓	2	1	✓	-	✓	✗	✓	✓
31	1	0,0	✓	2	1	✓	✓	✓	✓	✓	✓
32	1	0,2	✓	2	1	✓	✓	✓	✓	✓	✓
33	1	0,0	✓	8	1	✓	-	✓	✓	✓	✓
34	1	0,2	✓	8	1	✓	-	✓	✓	✓	✓
35	1	0,0	✓	8	1	✓	✓	✓	✓	✓	✓
36	1	0,2	✓	8	1	✓	✓	✓	✓	✓	✓
37	1	0,0	✓	32	1	✓	-	✓	✓	✓	✓
38	1	0,2	✓	32	1	✓	-	✓	✓	✓	○
39	1	0,0	✓	32	1	✓	✓	✓	✓	✓	✓
40	1	0,2	✓	32	1	✓	✓	✓	✓	✓	✓
41	1	0,0	✓	2	2	✓	-	✓	✓	✓	✓
42	1	0,2	✓	2	2	✓	-	✓	✓	✓	✓
43	1	0,0	✓	2	2	✓	✓	✓	✓	✓	✓
44	1	0,2	✓	2	2	✓	✓	✓	✓	✓	✓

Abb. 7.21: Untersuchte Parametrisierungen für die Teilsequenz B, gegeben das reale Verdeckungsvolumen, das zu einer partiellen Objektverdeckung führt, Fortsetzung der Tabelle aus Abb. 7.20.

Tisch auf die vorhandenen *unknown*-Voxel (rot) legt und daher für den gegebenen Frame 1890 (Seed 2) eine recht längliche Form besitzt. Wichtig ist, wie bereits beschrieben, dass die Blocking-Methode eingesetzt wird. Ohne diese würden sich beide Filter auf der Person positionieren, die sich auf dem Tisch befindet, da hierzu ein größerer Bereich von *unknown*-Voxeln rekonstruiert wurde, der zu höheren Ellipsoidgewichten führt. In Abb. 7.22(b) ist zum Vergleich die Parametrisierung von Zeile 44 (Frame 1890, Seed 2) dargestellt, deren Ergebnis visuell besser bewertet und noch diskutiert wird.

In Zeile 2 von Abb. 7.20 werden die *occluded*-Voxel positiv mit 0,2 gewichtet. Dadurch streckt sich das Schwerpunktellipsoid noch mehr, wodurch es das Verdeckungsvolumen stärker schneidet als in Zeile 1. Der Tracking-Verlauf ändert sich etwas, bleibt aber erfolgreich. Im nächsten Schritt wird der Kollisionstest hinzugefügt, zunächst ohne die positive Gewichtung der *occluded*-Voxel (Zeile 3), anschließend mit dieser Gewichtung (Zeile 4). Durch die Kollisionsvermeidung weicht das Schwerpunktellipsoid dem Tisch aus. Im Effekt geht das Ellipsoid, während sich die Person unter den Tisch begibt, etwas weiter in den Boden. Nach Initialisierung mit Seed 1 in Zeile 3 schwingt zudem das Schwerpunktellipsoid kurzzeitig aus dem Bereich unterhalb des Tischs zu dem Artefakt hin. Das Ergebnis sieht ähnlich aus wie das von Zeile 8, welches in Abb. 7.23(n) dargestellt ist (Frame 1920, Seed 3). In Zeile 4, bei der zusätzlich die *occluded*-Voxel ein Gewicht erhalten, diffundiert das Schwerpunktellipsoid bei Seed 2 und 4 zunächst

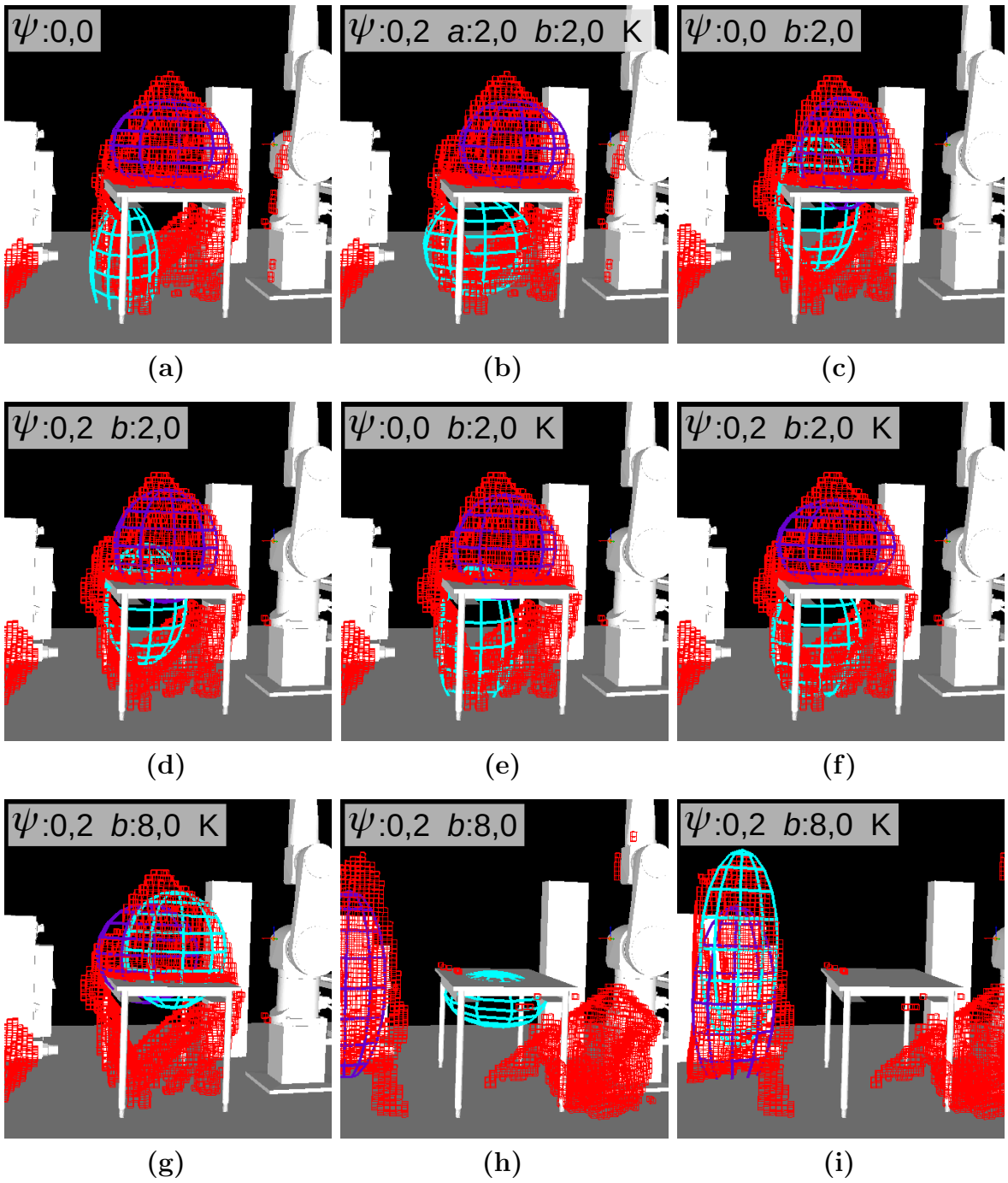


Abb. 7.22: Virtuelle Ansichten des Trackings, Gewichtung mit nichtlinearer Verstärkung (Gain) und verschiedenen Parametrisierungen (K: Kollisionstest). Betrachtet werden mehrere Frames der Teilsequenz B. Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide zweier Partikelfilter (hellblau und violett) sowie die statischen Objekte (weiß).

auf das Artefakt, bevor es unter den Tisch gelangt. Dies ist in Abb. 7.23(m) zu sehen (Frame 1873, Seed 2). Für Seed 1 schwingt der Filter unter dem Tisch etwas hin und her, entfernt sich also etwas von dem zu trackenden Objekt. Hierbei ergibt die Kombination aus Kollisionstest und Gewichtung der *occluded*-Voxel ein schlechteres Filterverhalten.

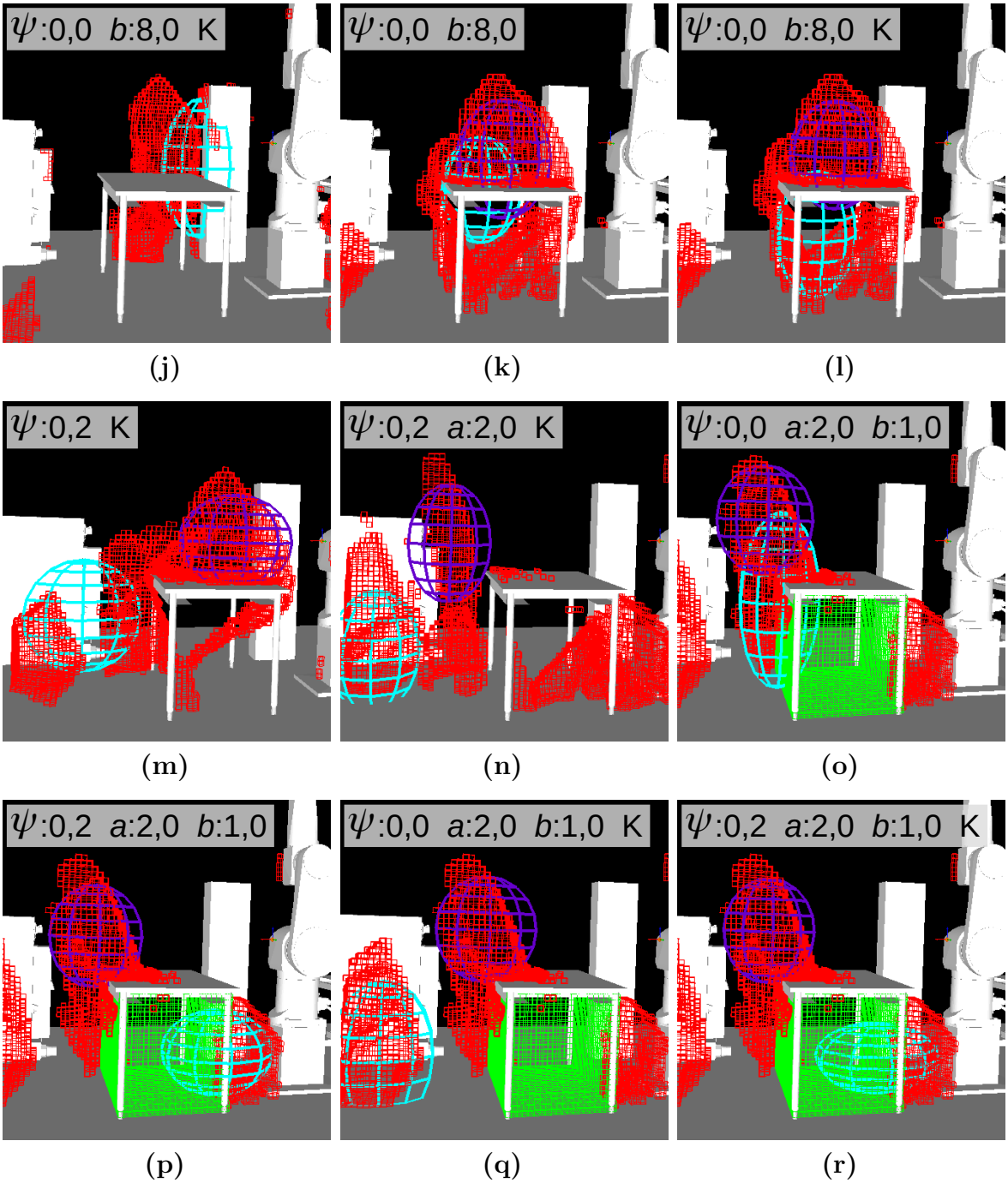


Abb. 7.23: Virtuelle Ansichten des Trackings, Gewichtung mit nichtlinearer Verstärkung (Gain) und verschiedenen Parametrisierungen (K: Kollisionstest). Betrachtet werden mehrere Frames der Teilsequenz B. Gezeigt werden die *unknown*-Voxel (rot), Schwerpunktellipsoide zweier Partikelfilter (hellblau und violett), ein synthetisches Verdeckungsvolumen (grün) sowie die statischen Objekte (weiß).

In den Zeilen 5 bis 16 werden zusätzlich die *filled*-Voxel bestraft. Die Bestrafung mit $a = 2$ (Zeilen 5–8) führt dann, wenn die *occluded*-Voxel mit 0,2 gewichtet werden und/oder ein Kollisionstest durchgeführt wird, nicht in allen Fällen zu einem erfolgreichen Tracking: Dreimal entfernt sich das Schwerpunktellipsoid von dem Objekt. In Zeile 8 verliert

der Filter das Objekt, dabei schwingt dieser unter dem Tisch hervor auf das Artefakt wie in Abb. 7.23(n) zu sehen (Frame 1920, Seed 3). Danach bewegt sich der Filter auf die andere Person. Auch die Ergebnisse der Bestrafung mit $a = 8$ der Zeilen 9–12 sind ebenfalls durchwachsen und es tritt ein Tracking-Verlust auf, weil der Filter aus dem Verdeckungsvolumen heraus auf das Artefakt diffundiert. Eine stärkere Bestrafung der *filled*-Voxel mit $a = 32$ (Zeilen 13–16) erbringt hierbei keinen positiven Effekt. In drei Fällen wird die Person verloren. Die Unterschiede zwischen der Verwendung des Bestrafungsterms für die *filled*-Voxel mit oder ohne Kollisionstest gehen aus den Durchgängen nicht eindeutig hervor. In allen drei Fällen ist zu beobachten, dass der Filter tendenziell einen größeren Abstand zur Tischplatte einhält und damit zunächst stärker in den Boden diffundiert, bevor er unter den Tisch gelangt.

Bei den nun folgenden Parameterkombinationen werden die *empty*-Voxel anstatt der *filled*-Voxel bestraft. Der Exponent b wird dabei mit 1, 2 oder 8 gewichtet. In den beiden ersten Fällen verläuft das Tracking durchweg positiv (Zeilen 17–24). Bei einer erhöhten Bestrafung der *empty*-Voxel schneidet das Schwerpunktellipsoid stärker die Tischplatte und enthält mehr *filled*- und *occluded*-Voxel. Die unbestraften *filled*-Voxel führen zu höheren Partikelgewichten als die bestraften *empty*-Voxel. Bei Aktivierung des Kollisionstests hält das Schwerpunktellipsoid erfolgreich einen Abstand zur Tischplatte. Für einen gewählten Exponenten von $b = 1$ geht das Ellipsoid in den meisten Fällen etwas in den Boden. Hingegen ergibt sich für $b = 2$ eine Meidung des Bodens, weil die *outside*-Voxel genauso wie die *empty*-Voxel höher bestraft werden. In den Abbildungen 7.22(c) bis (f) sind die Ergebnisse für Frame 1892 der Zeilen 21–24 (von Seed 3) zu sehen. Die Abb. 7.22(f), bei welcher sowohl der Kollisionstest als auch die Gewichtung der *occluded*-Voxel aktiv ist, sieht dabei am besten aus. Warum eine Bestrafung der *empty*-Voxel zu einer Verbesserung der Tracking-Ergebnisse führt, lässt sich wie folgt begründen: Die spezifische Herausforderung der Teilsequenz B besteht in der Störung durch Artefakte, die am Zellrand von der roten Person verursacht werden, insbesondere in dem Moment, in dem sie über den Tisch klettert. Diese Pseudomessung kann den Filter „auf sich ziehen“, obwohl sich an der Stelle keine Person befindet, dargestellt in den Abbildungen 7.23(m) und (n). Die Bestrafung der *empty*-Voxel wirkt für den Filter wie eine Barriere im Raum und kann ein Abdriften des Filters auf ein Artefakt verhindern.

Ist der Wert des Exponenten zu hoch wie bei $b = 8$, so entstehen negative Effekte. Zu Beginn der Teilsequenz B bleibt der Filter kurzzeitig im Bereich des Regals mit *filled*- und *occluded*-Voxeln stehen, wie in Abb. 7.23(j) für Zeile 27 gezeigt (Frame 1828, Seed 3). Für Partikel, die nicht exakt genug auf dem zu verfolgenden Objekt liegen, ergeben sich aufgrund der nicht bestraften Voxelzustände des Regals relativ gesehen höhere Partikelgewichte als für Partikel, die eine größere Anzahl an *empty*-Voxeln

beinhalten. Später verbleibt der Filter lange oder final in dem Verdeckungsvolumen des Tischs (Zeilen 25–26), zu sehen in Abb. 7.22(h) für Zeile 26 (Frame 1940, Seed 1). Ohne Kollisionstest diffundieren beide Filter in Zeile 25 stark in die Tischplatte, siehe Abb. 7.23(k) (Frame 1892, Seed 1). Ein aktivierter Kollisionstest in Zeile 27 verhindert dies, wie in Abb. 7.23(l) gezeigt (Frame 1892, Seed 1). Dabei gehen die Filter für die Seeds 2 und 3 über den Tisch auf die rote Person und verfolgen diese anschließend beide weiter. In Zeile 28 geschieht das gleiche für alle Seeds. Dargestellt ist ein Beispiel in Abb. 7.22(g) für Zeile 28 (Frame 1892, Seed 1). Ein resultierendes Verfolgen der roten Person mit beiden Filtern ist für Zeile 28 in Abb. 7.22(i) zu finden (Frame 1940, Seed 1).

In den nachfolgenden Zeilen 29–44 (vgl. Abb. 7.21) werden beide Bestrafungsterme angewandt. Der Exponent b der *empty*-Voxel wird zwischen 1 und 2 variiert zur Verhinderung einer zu starken Bestrafung, die zu einem temporären oder finalen Verbleiben des Filters in dem Verdeckungsvolumen führen kann. Die gleichzeitige Bestrafung der *filled*-Voxel (die räumlich mit den *occluded*-Voxeln verbunden sind) mildert dies zwar ab, dennoch ist von einer stärkeren Bestrafung der *empty*-Voxel abzusehen, um eine Verarmung der Partikel in ihrer Streucharakteristik zu vermeiden. Die Bestrafung der *filled*-Voxel wird bei $b = 1$ variiert zwischen $a = 2$, 8 und 32. Aus dem Vergleich der Bildersequenzen für $a = 2$ und $a = 8$ (Zeilen 29, 30 im Vergleich zu Zeilen 33, 34) geht hervor, dass sich das Schwerpunktellipsoid bei der ersten Variante erkennbar weiter in der Tischplatte befindet (während die Person unter dem Tisch positioniert ist). Die Bestrafung der *filled*-Voxel hat demnach den erwarteten Effekt der Abstoßung des Schwerpunktellipsoids. In den Varianten mit Kollisionstest (Zeilen 31, 32 im Vergleich zu Zeilen 35, 36) ist kein deutlicher Unterschied erkennbar. Was jedoch aus der Betrachtung der Bildersequenzen für $a = 32$ hervorgeht, ist die signifikante „Abstoßung“ der Person während des Vorbeigehens am Tisch, wodurch das resultierende Schwerpunktellipsoid vertikal stärker schwankt.

Zuletzt wird der Fall besprochen, für welchen gilt: $a = b = 2$. Bei der Parametrisierung aus Zeile 41 liegt das Schwerpunktellipsoid deutlich in der Tischplatte. Die Gewichtung der *occluded*-Voxel in Zeile 42 mildert dies ab. Der aktivierte Kollisionstest der Zeilen 43 und 44 verhindert das Diffundieren des Schwerpunktellipsoids in die Tischplatte fast gänzlich. Die Durchgänge sind insgesamt positiv zu bewerten. Das Ergebnis aus Zeile 44 ist in Abb. 7.22(b) dargestellt (für Frame 1890, Seed 2).

Nach detaillierter Ergebnisbetrachtung konnten Tracking-Erfolge für verschiedene Parametrisierungen verzeichnet werden. Demnach ist die einfachste Gewichtung aus Zeile 1 bei dem betrachteten Beispiel eines Verdeckungsvolumens ebenso erfolgreich wie Kombinationen, bei denen alleinig die *empty*-Voxel bestraft werden. Die ausschließliche

Bestrafung der *filled*-Voxel führte für die Teilsequenz B nicht zum Erfolg. Hingegen zeigte die Kombination sämtlicher Gewichtungs- und Bestrafungsmechanismen positive Ergebnisse. Der Kollisionstest verhindert ein vollständiges Durchqueren des Tischs durch den Filter, wobei abhängig vom Parametersatz ein teilweises Durchdringen dennoch auftreten kann und auch für eine gute Personenverfolgung zugelassen werden sollte. Mit der Anwendung eines Kollisionstests sowie der Differenzierung des Voxelzustands *occluded* werden Mechanismen zur Verfügung gestellt, die insbesondere auch bei größeren Objektverdeckungen relevant sein können. Dies soll nun im folgenden Abschnitt näher untersucht werden.

Vollständige Objektverdeckung

Im Folgenden werden die Ergebnisse aus Tabelle 7.24 diskutiert für das hinzugefügte Verdeckungsvolumen aus Abb. 7.1(c), durch das es in Teilsequenz B zu einer vollständigen Objektverdeckung der grünen Person kommt. Die Experimente beginnen im ersten Durchgang wieder mit der ausschließlichen Gewichtung der *unknown*-Voxel in Zeile 1. Der Filter kann der grünen Person bei allen Seeds nicht vollständig unter den Tisch in das Verdeckungsvolumen folgen, er diffundiert auf Artefakte und anschließend auf *unknown*-Voxel, welche zur Messung der roten Person gehören. Die zusätzliche Gewichtung der *occluded*-Voxel in Zeile 2 führt nur bei Seed 1 zum Tracking-Erfolg. Die Anwendung des Kollisionstests in den Zeilen 3 und 4 verbessert die Ergebnisse nicht. In den Bildersequenzen ist zwar erkennbar (nicht dargestellt), dass der Filter mithilfe der Gewichtung der *occluded*-Voxel unter den Tisch gelangen kann (Zeilen 2 und 4), jedoch lässt sich damit nicht das anschließende Driften des Filters aus dem Verdeckungsvolumen auf das Artefakt sowie das Verfolgen der roten Person verhindern.

In den Zeilen 5–16 wird die Gewichtung um den Bestrafungsterm für die *filled*-Voxel erweitert. In den Varianten der Zeilen 5–7 mit $a = 2$ zeigt das Tracking keinen Erfolg. Der Filter gelangt unter den Tisch, schwingt jedoch wieder heraus. Die stärkere Bestrafung mit $a = 8$ führt bei gleichzeitiger Belohnung der *occluded*-Voxel mit $\psi = 0{,}2$ und dem Einsatz des Kollisionstests in 50% der Fälle zu einem erfolgreichen Tracking (Zeile 12). Eine noch stärkere Bestrafung mit $a = 32$ ergibt in allen Fällen einen Tracking-Verlust (Zeile 16). Zu sehen ist auch, dass sich eine mittlere bis starke Bestrafung der *filled*-Voxel mit $a = 8$ oder $a = 32$ bei fehlendem Kollisionstest positiv auf das Tracking auswirken kann (Zeilen 10 und 14), was alleinig mit dem Kollisionstest für die gegebene Konstellation nicht gelingt (Zeile 4). Der Kollisionstest könnte dahingehend als stärkere Bewegungseinschränkung im Zustandsraum bewertet werden als die, die durch den Ansatz der Bestrafung der *filled*-Voxel verursacht wird. Erkennbar ist weiterhin in den Zeilen 1–16, dass ein erfolgreiches Tracking nur in den Fällen gelingt, in denen die

Zeile	Gewichte		Gain	Bestrafung		Block.-methode	Kollis.-test	Seed Nummer			
	υ	ψ		a	b			1	2	3	4
1	1	0,0	✓	-	-	✓	-	X	X	X	X
2	1	0,2	✓	-	-	✓	-	✓	X	○	X
3	1	0,0	✓	-	-	✓	✓	X	X	X	X
4	1	0,2	✓	-	-	✓	✓	X	X	X	X
5	1	0,0	✓	2	-	✓	-	X	X	X	X
6	1	0,2	✓	2	-	✓	-	X	X	X	X
7	1	0,0	✓	2	-	✓	✓	X	X	X	X
8	1	0,2	✓	2	-	✓	✓	X	○	X	✓
9	1	0,0	✓	8	-	✓	-	X	X	X	X
10	1	0,2	✓	8	-	✓	-	X	X	○	X
11	1	0,0	✓	8	-	✓	✓	X	X	X	X
12	1	0,2	✓	8	-	✓	✓	X	X	✓	○
13	1	0,0	✓	32	-	✓	-	X	X	X	X
14	1	0,2	✓	32	-	✓	-	X	✓	○	X
15	1	0,0	✓	32	-	✓	✓	X	X	X	X
16	1	0,2	✓	32	-	✓	✓	X	X	X	X
17	1	0,0	✓	-	1	✓	-	X	✓	✓	✓
18	1	0,2	✓	-	1	✓	-	✓	✓	✓	✓
19	1	0,0	✓	-	1	✓	✓	✓	✓	X	✓
20	1	0,2	✓	-	1	✓	✓	✓	✓	✓	✓
21	1	0,0	✓	-	2	✓	-	✓	✓	✓	✓
22	1	0,2	✓	-	2	✓	-	✓	✓	✓	✓
23	1	0,0	✓	-	2	✓	✓	✓	✓	✓	✓
24	1	0,2	✓	-	2	✓	✓	✓	✓	✓	✓
25	1	0,0	✓	-	8	✓	-	○	○	○	○
26	1	0,2	✓	-	8	✓	-	X	X	X	X
27	1	0,0	✓	-	8	✓	✓	○	X	X	X
28	1	0,2	✓	-	8	✓	✓	X	X	X	X

Abb. 7.24: Untersuchte Parametrisierungen bei **synthetischem Verdeckungsvolumen** für die Teilsequenz B, das zu einer **vollständigen Objektverdeckung** führt. Die nichtlineare Verstärkung (Gain) der unknown-Voxel sowie die Blocking-Methode werden überall eingesetzt. Die Ergebnisse beziehen sich auf die Tracking-Güte der grünen Person, die sich unter den Tisch begibt. Die Position des Schwerpunktellipsoids wird für die Person visuell bewertet. Jeder Parametersatz wird mit einem pseudorandomisierten Zufallsgenerator mit vier festen Startwerten (Seeds) initialisiert. Ein Häkchen (✓) bedeutet, dass der Filter die grüne Person gut trackt. Die visuelle Beurteilung ergibt dabei eine ungefähre Übereinstimmung des Schwerpunktellipsoids mit der tatsächlichen Objektposition über die betrachtete Sequenz hinweg. Verliert der Filter die Person mindestens einmal, so ist dies mit (X) gekennzeichnet. Ein Kreis ist eingetragen (○), wenn das Schwerpunktelliposid mindestens einmal temporär stärker von der tatsächlichen Position der Person abweicht, die Person aber nicht vollständig verloren wird.

Zeile	Gewichte		Gain	Bestrafung		Block.-methode	Kollis.-test	Seed Nummer			
	υ	ψ		a	b			1	2	3	4
29	1	0,0	✓	2	1	✓	-	X	✓	✓	✓
30	1	0,2	✓	2	1	✓	-	✓	✓	✓	✓
31	1	0,0	✓	2	1	✓	✓	X	○	○	✓
32	1	0,2	✓	2	1	✓	✓	✓	✓	✓	✓
33	1	0,0	✓	8	1	✓	-	○	✓	✓	X
34	1	0,2	✓	8	1	✓	-	✓	✓	✓	✓
35	1	0,0	✓	8	1	✓	✓	○	X	✓	X
36	1	0,2	✓	8	1	✓	✓	✓	✓	✓	✓
37	1	0,0	✓	32	1	✓	-	X	X	X	X
38	1	0,2	✓	32	1	✓	-	✓	✓	✓	✓
39	1	0,0	✓	32	1	✓	✓	X	X	X	X
40	1	0,2	✓	32	1	✓	✓	X	✓	✓	✓
41	1	0,0	✓	2	2	✓	-	✓	✓	✓	✓
42	1	0,2	✓	2	2	✓	-	✓	✓	✓	✓
43	1	0,0	✓	2	2	✓	✓	✓	✓	✓	✓
44	1	0,2	✓	2	2	✓	✓	✓	✓	✓	✓

Abb. 7.25: Untersuchte Parametrisierungen bei synthetischem Verdeckungsvolumen für die Teilsequenz B, das zu einer vollständigen Objektverdeckung führt, Fortsetzung der Tabelle aus Abb. 7.24.

occluded-Voxel ein positives Gewicht erhalten. Dabei gelangt das Schwerpunktellipsoid gut unter den Tisch und hält auch einen Abstand zur Tischplatte ein, ohne diese zu durchdringen.

In den Zeilen 17–28 werden alleinig die *empty*-Voxel bestraft. Dies bringt wie in dem vorangegangenen Experiment einen positiven Effekt mit sich, wenn die Bestrafung nicht zu hoch ausfällt, wie für $b = 1$ und $b = 2$. In den Zeilen 17–28 wird jeweils ein Tracking-Erfolg erzielt, unabhängig davon, ob der Kollisionstest ausgeführt wird oder nicht. Ausnahmen sind hierbei die Durchgänge von Seed 1 in Zeile 17 und Seed 3 in Zeile 19, bei denen die *occluded*-Voxel nicht gewichtet werden.

Die Bestrafung der *empty*-Voxel mit $b = 8$ führt in Kombinationen aus positiver Gewichtung der *occluded*-Voxel und/oder der Anwendung des Kollisionstests meist zu einem Verlust der Person (Zeilen 26–28). Die Partikel erhalten in dem Verdeckungsvolumen die höchsten Gewichte und der Kollisionstest wirkt zusätzlich als Barriere, die den Filter unter dem Tisch hält. Meist wird auch der andere Filter dabei noch in das Verdeckungsvolumen unter den Tisch „gezogen“. Die höheren Partikelgewichte in dem Verdeckungsvolumen lassen sich wieder mit der impliziten Aufwertung der *occluded*- und *unknown*-Voxel begründen, wenn die *empty*-Voxel bestraft werden. Zudem liegen die Ellipsoide wieder konzentriert auf den *unknown*-Voxeln, wenn die umgebenden

empty-Voxel deutlich bestraft werden, was das Folgen der Person erschwert, da die Partikelstreuung dadurch zu stark eingeschränkt wird. Wird keine positive Gewichtung der *occluded*-Voxel vorgenommen (Zeile 25) und auch kein Kollisionstest durchgeführt, so ist es möglich, dass der Filter im letzten Moment trotz hoher Bestrafung noch der Person folgt, in den anderen Fällen jedoch meist nicht (Zeilen 26–28). Das Filterverhalten ist in allen Fällen nicht zufriedenstellend.

Zuletzt werden wieder die kombinierten Bestrafungsterme aus Abb. 7.25 betrachtet (Zeilen 29–44). Hierbei lässt sich erkennen, dass die Bestrafung der *empty*-Voxel mit $b = 1$ bei gleichzeitiger Bestrafung der *filled*-Voxel mit $a = 2, 8$ und 32 immer dann einen positiven Tracking-Verlauf ergibt, wenn gleichzeitig die *occluded*-Voxel mit 0,2 gewichtet werden (außer in Zeile 40 für Seed 1, was in Tabelle 7.25 rot markiert ist). Der Unterschied mit und ohne Gewichtung der *occluded*-Voxel geht aus den Abbildungen 7.23(o) bis (r) (Frame 1912, Seed 1) für die Zeilen 29–32 hervor. Die ausschließliche Bestrafung der *empty*-Voxel mit $b = 1$ genügt demnach nicht, um in jedem Fall ein erfolgreiches Tracking zu ermöglichen. Die zusätzliche Bestrafung der *filled*-Voxel hat einen leicht negativen Einfluss (Zeilen 17–20). Mit $b = 2$ (Zeilen 41–44) kann hingegen auch eine fehlende Gewichtung der *occluded*-Voxel kompensiert und ein erfolgreiches Tracking ermöglicht werden. Die Ergebnisse der äquivalenten Bestrafung von $a = b = 2$ der Zeilen 41–44 zeigen im Vergleich zu den Durchgängen der Zeilen 21–24 folgende Unterschiede: Ohne die positive Gewichtung der *occluded*-Voxel schneidet das Ellipsoid die Tischplatte, andernfalls nicht. Ansonsten lassen sich nur geringe Unterschiede feststellen, weshalb die Bestrafung der *filled*-Voxel sicherlich nicht notwendig ist, wenn bereits der Kollisionstest verwendet wird.

7.6.2 Verschiedene Situationen

Bisher wurden bei den Experimenten variierende Verdeckungsvolumina unterhalb des Tischs für die Teilsequenz B betrachtet. Im Folgenden werden weitere Teilsequenzen untersucht. Dazu wird die aufgezeichnete Gesamtsequenz für das Tracking von zwei Personen in die Teilsequenzen unterteilt, die in Abb. 7.26 aufgelistet sind. Das Filterverhalten wird für sämtliche dieser Teilsequenzen analysiert, wobei die Filterinitialisierung für jede Teilsequenz separat vorgenommen wird (mit den vier konstanten Seeds), um wieder eine entsprechende Vergleichbarkeit der Ergebnisse zu garantieren. Weiterhin werden zwei Parametersätze vergleichend betrachtet: Der Parametersatz 1 entspricht einer Gewichtung der *unknown*-Voxel mit $\upsilon = 1{,}0$ und nichtlinearer Verstärkung (Gain), wie in Formel (7.2) angegeben. Die *occluded*-Voxel werden dabei nicht gewichtet und die Voxelzustände *filled* und *empty* erhalten keine Bestrafung. Die Blocking-Methode findet Anwendung, der Kollisionstest hingegen nicht.

Start-Frame	Beschreibung der Teilsequenz	Vollständige Objektverdeckung	Parametersatz 1 S1	S2	S3	S4	Parametersatz 2 S1	S2	S3	S4
380	R, G betreten die Zelle, G geht in Hocke	-	✓	✓	✓	✓	✓	✓	✓	✓
620	R, G gehen aneinander vorbei	-	✓	✓	✓	✓	✓	✓	✓	✓
670	R steht vor dem Tisch (größere Artefakte)	-	✓	✓	✓	✓	✓	✓	✓	✓
750	G kniet am Roboter, R kommt dazu	-	✓	✓	✓	✓	✓	○	○	✓
870	G steigt auf den Tisch, R geht darunter	-	✓	✓	✓	✓	✓	✓	✓	✓
1235	G macht Richtungswechsel und passiert R	-	✓	✓	✓	✓	✓	✓	✓	✓
1320	R, G gehen aneinander vorbei (Regal)	-	○	○	○	○	○	✗	○	○
1500	R, G laufen einzeln herum	-	✓	✓	✓	✓	✓	✓	✓	✓
1730	R, G gehen aneinander vorbei (Laufband)	-	✓	✓	✓	✓	✓	○	✗	○
1820	R sitzend auf Tisch, G unter Tisch	-	✓	✓	✓	✓	✓	✓	✓	✓
1980	G sitzend auf Tisch, R am Laufband	-	✓	✓	✓	✓	✓	✓	✓	✓
2200	R, G gehen aneinander vorbei	-	✓	✓	✓	✓	✓	✓	✓	✓
2250	G verlässt kurz die Arbeitszelle									
2390	R, G laufen einzeln herum	-	✓	✓	✓	✓	✓	✓	✓	✓
2510	R unter dem Tisch durch, G steht daneben	-	✓	✓	✓	✓	✓	✓	✓	✓
2740	R, G laufen einzeln herum	-	✓	✓	✓	✓	✓	✓	✓	✓
3040	R, G gehen an 2 Seiten des Regals vorbei	-	✓	✓	✓	✓	✓	✓	✓	✓
3170	R, G verlassen kurz die Arbeitszelle									
3410	G legt sich unter den Tisch, R steht daneben	-	✓	✓	✓	✓	✓	✓	✓	✓
3780	R, G gehen aneinander vorbei	-	✓	✓	✓	✓	○	○	○	○
3890	R, G gehen aufeinander zu und zurück	-	✓	✓	✓	✓	✓	✓	✓	✓
4050	R, G gehen aneinander vorbei	-	✓	✓	✓	✓	✓	✓	✓	✓
4130	R, G laufen einzeln herum	-	✓	✓	✓	✓	✓	✓	✓	✓
4250	R, G stehen länger zusammen	-	✓	✓	✓	✓	✓	✓	✓	✓
4410	R verlässt die Arbeitszelle	-	✓	✓	✓	✓	✓	✓	✓	✓

Abb. 7.26: Tracking-Ergebnisse für verschiedene Teilsequenzen einer gesamten Videosequenz mit zwei Personen (R: rote Person, G: grüne Person). Für jede Teilsequenz ist der Start-Frame eingetragen. Die Videosequenz endet mit Frame 4499. Die Likelihood-Gewichtung erfolgt mit dem Parametersatz 1 (simple Gewichtung) und dem Parametersatz 2 (komplexere Gewichtung). Jeweils sind die Tracking-Ergebnisse für die Initialisierung mit vier verschiedenen Seeds eingetragen (S1 bis S4). Durch die real gegebenen Verdeckungsvolumina entstehen keine vollständigen Objektverdeckungen.

Beim Parametersatz 2 werden die *unknown*-Voxel ebenfalls mit $\upsilon = 1{,}0$ gewichtet und nichtlinear verstärkt. Zusätzlich werden die verdeckten *occluded*-Voxel mit $\psi = 0{,}2$ gewichtet, wodurch das Tracking durch Verdeckungsvolumina hindurch ermöglicht werden soll. Weiterhin erhalten die *filled*- und *empty*-Voxel eine Bestrafung, bei der die Exponenten der Bestrafungsterme auf $a = 2$ bzw. $b = 2$ gesetzt werden. Die Gewichtung entspricht der Formel (6.33). Der Kollisionstest ist für diesen Parametersatz aktiviert und die Blocking-Methode wird eingesetzt.

Verlassen eine oder beide Personen die Roboterarbeitszelle, so terminiert der Filter. Beim erneuten Betreten des Überwachungsraums wird wieder eine Filterinitialisierung gestartet. Bei den Teilsequenzen in den nachfolgenden Tabellen, in denen solche Situationen auftreten, ist keine Bewertung in der zugehörigen Zeile eingetragen, da eine Filterterminierung in diesen Fällen erwünscht ist.

Partielle Objektverdeckung

Im ersten Schritt werden wieder partielle Objektverdeckungen betrachtet, die bei dem gegebenen realen Verdeckungsvolumen unterhalb des Tischs entstehen (vgl. Abb. 7.1(a)). Die Tracking-Ergebnisse für die beiden Parametersätze 1 und 2 sind in der Tabelle von Abbildung 7.26 dargestellt. Da es zu keiner vollständigen Objektverdeckung kommt, gibt es auch keine Einträge in der entsprechenden Spalte. Mit dem Parametersatz 1 konnte in allen Situationen erfolgreich getrackt werden, mit einer Ausnahme: In der Teilsequenz mit dem Start-Frame 1320 gehen die beiden Personen beim Regal eng aneinander vorbei. Hierbei wird eine Person temporär von ihrem Filter verloren, aber final verfolgen beide Filter wieder jeweils eine Person. Das gleiche gilt bei dieser Teilsequenz für den Parametersatz B, wobei die Initialisierung mit Seed 2 (S2) dazu führt, dass eine Person gänzlich vom Filter verloren wird. Für drei weitere Teilsequenzen, mit den Start-Frames 750, 1320 und 3780 zeigt sich ein ähnliches Filterverhalten für den Parametersatz B. Betroffen sind dabei Situationen, in denen beide Personen miteinander interagieren. Eine Vermutung dazu ist, dass nach einem Merge-Ereignis die Bestrafung der *empty*-Voxel verhindert, dass die Partikel beider Filter weit genug gestreut werden. Dadurch könnten nach dem Split-Ereignis auf beiden Teilmessungen zu wenige Partikel liegen, die eine Verfolgung dieser ermöglichen. Im Effekt wird eine Person verloren und beide Filter tracken die gleiche Person (trotz Blocking-Methode). Ob die Vermutung zutrifft, müsste jedoch genauer analysiert werden.

Start-Frame	Beschreibung der Teilsequenz	Vollständige Objektverdeckung	Parametersatz 1				Parametersatz 2			
			S1	S2	S3	S4	S1	S2	S3	S4
380	R, G betreten die Zelle, G geht in Hocke	-	✓	✓	✓	✓	✓	✓	✓	✓
620	R, G gehen aneinander vorbei	✓	X	X	X	X	✓	✓	✓	✓
670	R steht vor dem Tisch (größere Artefakte)	-	O	O	✓	O	O	✓	O	O
750	G kniet am Roboter, R kommt dazu	-	O	O	O	O	✓	✓	✓	O
870	G steigt auf den Tisch, R geht darunter	-	✓	✓	✓	✓	✓	✓	✓	✓
1235	G macht Richtungswechsel und passiert R	✓	✓	✓	✓	✓	O	✓	✓	O
1320	R, G gehen aneinander vorbei (Regal)	-	O	O	O	O	X	O	X	O
1500	R, G laufen einzeln herum	✓	X	X	X	X	✓	✓	✓	✓
1730	R, G gehen aneinander vorbei (Laufband)	-	✓	✓	✓	✓	O	O	O	O
1820	R sitzend auf Tisch, G unter Tisch	✓	X	X	X	X	✓	✓	✓	✓
1980	G sitzend auf Tisch, R am Laufband	✓	X	X	X	X	✓	✓	✓	✓
2200	R, G gehen aneinander vorbei	✓	X	X	X	X	✓	✓	✓	✓
2250	G verlässt kurz die Arbeitszelle									
2390	R, G laufen einzeln herum	✓	X	X	X	X	✓	✓	✓	✓
2510	R unter dem Tisch durch, G steht daneben	✓	X	X	X	X	✓	✓	✓	✓
2740	R, G laufen einzeln herum	✓	X	X	X	X	O	O	✓	O
3040	R, G gehen an 2 Seiten des Regals vorbei	✓	X	✓	X	X	✓	✓	✓	✓
3170	R, G verlassen kurz die Arbeitszelle									
3410	G legt sich unter den Tisch, R steht daneben	✓	X	X	X	X	✓	✓	✓	✓
3780	R, G gehen aneinander vorbei	✓	X	X	X	X	✓	✓	✓	O
3890	R, G gehen aufeinander zu und zurück	✓	X	X	X	X	✓	✓	✓	✓
4050	R, G gehen aneinander vorbei	✓	X	X	X	X	✓	✓	✓	✓
4130	R, G laufen einzeln herum	-	✓	✓	✓	✓	O	✓	✓	✓
4250	R, G stehen länger zusammen	✓	✓	X	X	X	✓	✓	✓	✓
4410	R verlässt die Arbeitszelle	✓	X	X	X	X	✓	✓	✓	✓

Abb. 7.27: Tracking-Ergebnisse für verschiedene Teilsequenzen einer gesamten Videosequenz mit zwei Personen (R: rote Person, G: grüne Person). Für jede Teilsequenz ist der Start-Frame eingetragen. Die Videosequenz endet mit Frame 4499. Die Likelihood-Gewichtung erfolgt mit dem Parametersatz 1 (simple Gewichtung) und dem Parametersatz 2 (komplexere Gewichtung). Jeweils sind die Tracking-Ergebnisse für die Initialisierung mit vier verschiedenen Seeds eingetragen (S1 bis S4). Durch das Hinzufügen eines synthetischen Verdeckungsvolumens können vollständige Objektverdeckungen entstehen. Dies ist in der Teilsequenz dann der Fall, wenn in der Spalte „Vollständige Objektverdeckung“ ein Häkchen gesetzt ist.

Vollständige Objektverdeckung

Nach der Betrachtung partieller Objektverdeckungen werden nun vollständige Objektverdeckungen untersucht. Um diese hervorzurufen, wird ein größeres synthetisches Verdeckungsvolumen in den Überwachungsraum integriert. Dieses ist in Abb. 7.1(d) dargestellt. Die Tracking-Ergebnisse sind in der Tabelle von Abb. 7.27 zu finden. Kommt es bei einer betrachteten Teilsequenz mindestens in einer Situation zu einer vollständigen Objektverdeckung, so ist in der entsprechenden Spalte ein Häkchen gesetzt. Die Ergebnisse des Parametersatzes 1 lassen erkennen, dass in fast allen Durchgängen, bei denen eine vollständige Objektverdeckung auftritt, auch ein Track-Verlust erfolgt. Bei den Teilsequenzen mit den Start-Frames 670 und 750 findet zwar keine vollständige Objektverdeckung statt, dennoch sind die Ergebnisse negativer als in dem vorangegangenen Experiment zu den realen partiellen Objektverdeckungen, was auf eine entsprechende Wirkung des Verdeckungsvolumens auf die Filter schließen lässt. Mit dem Parametersatz 2 kann auch bei vollständigen Objektverdeckungen erfolgreich getrackt werden. Die Teilsequenz mit dem Start-Frame 1320 bereitet aber auch hier wieder Schwierigkeiten. Zwei Mal kommt es zu einem Tracking-Verlust. Bei anderen Teilsequenzen mit den Start-Frames 670, 750, 1235, 1730, 2740, 3780 und 4130 verliert ein Filter auch temporär das dynamische Objekt. Das berechnete gewichtete Schwerpunktellipsoid weicht in diesen Fällen sichtbar von der tatsächlichen Position des dynamischen Objekts ab. Für die Teilsequenz mit Start-Frame 670 erscheint dies überraschend. Bei den anderen Teilsequenzen hingegen lässt sich als Ursache wieder ein gewisser Grad an Interaktion zwischen den Personen ausfindig machen (räumliche Nähe), die bei den partiellen Objektverdeckungen bereits ähnliche Ergebnisse hervorriefen. Hierbei könnte eine geeignetere Blocking-Methode Abhilfe verschaffen.

7.6.3 Zusammenfassung

Die Herausforderung des Trackings in Teilsequenz B ist gegeben durch störende Artefakte. Diese können den Filter der grünen Person, die sich temporär unter dem Tisch befindet, auf sich „ziehen“. Bei der partiellen Objektverdeckung gelingt dennoch ein erfolgreiches Tracking, auch mit der einfachsten Parametrisierung, bei der lediglich die *unknown*-Voxel positiv gewichtet werden (Tabelle 7.20, Zeile 1). Für die vollständige Objektverdeckung hingegen ist diese Parametrisierung unzureichend.

Eine Gewichtung der *occluded*-Voxel mit $\psi = 0{,}2$ ist bei der partiellen Objektverdeckung kein Garant für ein erfolgreiches Tracking. Bei dem vergrößerten Verdeckungsvolumen, das eine vollständige Objektverdeckung mit sich bringt, zeigt sich damit allerdings ein gutes Folgen der Person unter den Tisch. Jedoch ist das gewählte Gewicht nicht

ausreichend groß, um den Filter von einem Drift auf das Artefakt abzuhalten. Dies kann durch eine Bestrafung der *empty*-Voxel erreicht werden, was auch implizit mit einer Bestärkung der *occluded*-Voxel einhergeht. Eine höhere Gewichtung der *occluded*-Voxel als mit $\psi = 0{,}2$ würde voraussichtlich einen ähnlichen Effekt hervorrufen wie die Bestrafung der *empty*-Voxel. Dies wurde für die Teilsequenz B jedoch nicht untersucht.

Bei dem größeren Verdeckungsvolumen tritt als positiver Nebeneffekt mit Gewichtung der *occluded*-Voxel eine Verhinderung der Diffusion des Schwerpunktellipsoids durch die Tischplatte auf, was oftmals durch die benachbarte Messung von *unknown*-Voxeln über dem Tisch hervorgerufen wird. Dies lässt sich auf die größere Menge an *occluded*-Voxeln und deren positive Gewichtung zurückführen. Auch für den Kollisionstest, ebenso wie für den bestrafenden Term der *filled*-Voxel, zeigt sich ein resultierender Abstand des Filters zum Tisch. Die gleichzeitige Anwendung von Kollisionstest und Bestrafung bringt jedoch wie erwartet keine weiteren ersichtlichen Vorteile, da sie im Grunde eine ähnliche Funktionalität realisieren. Weiterhin war erkennbar, dass der Filter ohne Bestrafung der *empty*-Voxel (und damit der *outside*-Voxel) etwas in den Boden ausweicht.

Zur Verhinderung des beschriebenen Filterdrifts auf das Artefakt konnte die Bestrafung der *empty*-Voxel den größten Beitrag liefern, sowohl bei der partiellen als auch bei der vollständigen Objektverdeckung. Die Bestrafung der *empty*-Voxel zieht das Schwerpunktellipsoid in das Verdeckungsvolumen zur Tischplatte hin und baut eine Barriere um die Cluster der anderen Voxelzustände auf, was insbesondere für die nicht-statische Menge der *unknown*-Voxel relevant ist. Jedoch wird damit auch die Streuung der Partikel abgeschwächt, was sich bei einer entsprechend hohen Bestrafung, beispielsweise für $b = 8$, zu negativen Effekten hin verschiebt. So kann der Filter der Person meist nicht mehr aus dem Verdeckungsvolumen heraus folgen, was verstärkt wird durch die positive Gewichtung der *occluded*-Voxel und den Einsatz des Kollisionstests. Dies zeigte sich auch am Regal der Roboterarbeitszelle: Der Filter wurde bei höherer Bestrafung temporär oder permanent vom Regal (von seinen *occluded*-Voxeln) „festgehalten“.

In dem letzten Experiment der vorliegenden Dissertation wurde das Tracking anhand verschiedener Teilsequenzen untersucht, um das generelle Verhalten der Filter beim Einsatz der vorgeschlagenen Likelihood-Gewichtung zu überpüfen. Hierfür wurden die Tracking-Ergebnisse zweier ausgewählter Parametersätze 1 und 2 gegenübergestellt. Bei ersterem erhalten lediglich die *unknown*-Voxel ein Gewicht, also die Voxel, die zur rekonstruierten Visuellen Hülle der dynamischen Objekte gehören. Demnach werden Verdeckungsvolumina nicht gewichtet. Weiterhin werden *filled*- und *empty*-Voxel auch nicht bestraft und ein Kollisionstest wird nicht angewandt. Der Parametersatz 2 zeichnet sich durch eine positive Gewichtung der *occluded*-Voxel sowie den Einsatz von

Bestrafungstermen für die *filled*- und *empty*-Voxel aus. Zudem wird der Kollisionstest im Schritt der Partikelprädiktion eingesetzt.

Die Likelihood-Gewichtung mit Parametersatz 1 zeigte bei den realen partiellen Objektverdeckungen im Vergleich zu Parametersatz 2 die besseren Ergebnisse. Der Parametersatz 2 war dabei nicht deutlich schlechter, lediglich in einzelnen Situationen, in denen es durch die räumliche Nähe der Personen (Interaktionen) zu einem Merge und Split der Teilmessungen kam, war dieser unterlegen. Die Blocking-Methode wurde bei beiden Parametersätzen angewandt, konnte das Datenassoziationsproblem aber insbesondere bei der komplexeren Gewichtung mit Parametersatz 2 nicht ausreichend lösen. Hierbei könnte die Bestrafung der *empty*-Voxel jedoch auch einen zu großen Einfluss ausgeübt haben, was an dieser Stelle eine Vermutung bleiben soll. In Betrachtungen, die sich dieser Dissertation anschließen, könnte dies näher untersucht werden und auch der Fokus gezielter auf Interaktionen zwischen den zu trackenden Personen gesetzt werden.

Im Anschluss an partielle Objektverdeckungen wurden vollständige Objektverdeckungen untersucht. Es zeigte sich, dass mit der Anwendung des Parametersatzes 1 in den meisten Durchgängen ein Tracking-Verlust auftrat. Die komplexere Gewichtung mithilfe des Parametersatzes 2 konnte hingegen einen Tracking-Erfolg in diesen Situationen gewährleisten und den dynamischen Objekten gut durch das große Verdeckungsvolumen folgen, das in den betrachteten Teilsequenzen zu vollständigen Objektverdeckungen führte.

7.7 Zusammenfassung

In diesem Kapitel wurden Details zur Implementierung sowie zur Durchführung und Auswertung der Experimente beschrieben. Anschließend wurden verschiedene Experimente durchgeführt, mit folgenden Ergebnissen: Die Behandlung der Verdeckungsvolumina als Pseudomessungen durch die positive Gewichtung der *occluded*-Voxel mit $0 < \psi < \upsilon$ eignet sich zur Reduktion von Tracking-Verlusten dynamischer Objekte, wenn diese vollständig verdeckt sind. Eine nichtlineare Verstärkung der *unknown*-Voxel führt dazu, dass sich ein Filter besser auf der Messung positioniert, wenn sich die Personen partiell in Verdeckungsvolumina befinden. Eine Bestrafung der *filled*-Voxel resultiert darin, dass das Schwerpunktellipsoid den statischen Objekten ausweicht. Dies kann zur Vermeidung von Zuständen eingesetzt werden, die real nicht möglich sind. Eine gleichzeitige Durchführung des Kollisionstests bei der Partikelprädiktion ist nicht notwendig. Der Kollisionstest kann alternativ eingesetzt werden und verhindert im Vergleich zur Bestrafung der *filled*-Voxel eine vollständige Diffusion des Filters durch statische Objekte, was wünschenswert ist. Eine Bestrafung der *empty*-Voxel kann genutzt werden, um

eine stärkeren Fokussierung der Partikelellipsoide auf der Messung zu erzielen (weg von leeren Voxeln). Diese Bestrafung sollte jedoch gering ausfallen, z. B. mit $b = 1$, da andernfalls die Streucharakteristik der Partikel und damit die Filterbewegung zu stark eingeschränkt wird.

Treten Rekonstruktionsartefakte auf, so können diese störend wirken und einen Tracking-Verlust herbeiführen. Experimentell wurde gezeigt, dass eine komplexere Gewichtungsfunktion dem entgegenwirken kann.

In dem letzten Experiment wurde das Tracking von zwei Personen in verschiedenen Teilsequenzen untersucht. Eine einfache Likelihood-Funktion (Parametersatz 1) war für die gegebenen partiellen Objektverdeckungen erfolgreich. Bei vollständigen Objektverdeckungen wurde eine komplexere Likelihood-Funktion (Parametersatz 2) benötigt, um in den meisten Fällen einen Tracking-Erfolg zu ermöglichen. In Merge- und Splitszenarien fielen die Ergebnisse bei komplexerer Likelihood-Funktion in Kombination mit der Blocking-Methode etwas schlechter aus als bei der einfachen Likelihood-Funktion, was näher untersucht werden könnte.

Kapitel 8

Schlussfolgerungen

Das abschließende Kapitel der vorliegenden Dissertation fasst die gewonnenen Erkenntnisse zusammen, ordnet sie ein und stellt Anknüpfungspunkte für weitere Arbeiten dar. In Abschnitt 8.1 wird eine Zusammenfassung der wesentlichen erarbeiteten Inhalte und Ergebnisse gegeben. Diese werden in Abschnitt 8.2 von verschiedenen Seiten beleuchtet und diskutiert. Es werden auch Alternativen zu bestimmten gewählten Vorgehensweisen aufgezeigt. Weiterentwicklungsmöglichkeiten des gewählten Ansatzes werden im Ausblick in Abschnitt 8.3 aufgeführt.

8.1 Zusammenfassung

Gegenstand der Arbeit war die Realisierung eines Tracking-Verfahrens zum Verfolgen von Personen (dynamische Objekte) innerhalb einer Roboterarbeitszelle mit statischen Objekten und verdeckten Volumina. Dafür wurde ein Multi-View-Kamerasystem mit unterschiedlichen Blickwinkeln auf den Überwachungsraum eingesetzt. Die Verwendung mehrerer Kameras bietet eine größere Raumabdeckung als eine Einzelkamera sowie die Möglichkeit, verdeckte Sichtbereiche zu kompensieren, was als sensorische Kompensation bezeichnet wird. Damit lassen sich verdeckte Volumina im Raum für das Gesamtsystem reduzieren und die Sichtbarkeitsgrade der sichtbaren Volumina erhöhen. Zudem kann eine 3D-Rekonstruktion durchgeführt werden, die eine dreidimensionale Visuelle Hülle der dynamischen Objekte im Überwachungsraum liefert. Diese wurde als Eingabe für das Personen-Tracking verwendet.

Für Verdeckungen als Hauptuntersuchungsaspekt des Personen-Trackings wurden in Kapitel 3 eigene Verdeckungsbegriffe und -definitionen eingeführt, da „Verdeckungen“ in der Literatur uneinheitlich, meist unspezifisch und häufig nur für die 2D-Bildebene beschrieben werden. Bei den Definitionen wird die Datenfusion mehrerer Kameras berücksichtigt und Betrachtungen im 3D-Raum einbezogen, angelehnt an die Vorgehensweise des Volumenverschnitts einer 3D-Rekonstruktion. Gleichzeitig sind die Beschreibungen dabei sensorunabhängig. In den Betrachtungen wird zwischen sensorisch nicht einsehbaren Volumina, genannt „Verdeckungsvolumina“ und tatsächlich

auftretenden Verdeckungen der betrachteten Objekte unterschieden, genannt „Objektverdeckungen“. Letztere treten erst dann auf, wenn sich Objekte in ein für das gesamte Multi-View-Kamerasystem nicht einsehbares Verdeckungsvolumen begeben. Der Erfolg eines Tracking-Verfahrens bei Verdeckungen hängt meist von der Größenordnung auftretender (partieller) Objektverdeckungen ab. Kommt es zu vollständigen Objektverdeckungen, so führen diese häufig zu einem Versagen des Trackers. Ebenfalls ist die Dauer von Objektverdeckungen sehr relevant. Während kurzzeitige Objektverdeckungen durch die Fortführung von Bewegungsprädiktionen oftmals noch kompensiert werden können, werden größere Objektverdeckungen, die über mehrere Frames bestehen, meist zum Problem. In dieser Dissertation wird eine Lösung für ein Personen-Tracking im 3D-Raum unter gegebenen partiellen und vollständigen Objektverdeckungen präsentiert, deren Dauer prinzipiell uneingeschränkt lang sein kann.

Objektverdeckungen sind auch für die 3D-Rekonstruktion der Visuellen Hülle (VH) ein Problem. Das Verfahren der VH verwendet Silhouettenbilder dynamischer Objekte, die meist durch ein Background-Subtraction-Verfahren erzeugt werden. Die Rückprojektion der Silhouetten in den 3D-Raum mit anschließender Verschneidung führt zu den gewünschten approximierten Geometrien. Werden Teile der dynamischen Objekte durch statische Objekte im Raum verdeckt, so ergeben sich fehlerhafte Silhouettenbilder. Ist die Silhouette in nur einem verwendeten Bild unvollständig, so ist das Rekonstruktionsergebnis aufgrund des Prinzips des Volumenverschnitts ebenso unvollständig. Dann können Teile der dynamischen Objekte in der rekonstruierten Hülle fehlen. Zur Vermeidung dieses Problems wird das Rekonstruktionsverfahren einer Vorarbeit in Kapitel 5 aufgegriffen und erweitert. Dieses Rekonstruktionsverfahren integriert Wissen von 3D-Modellen der statischen Objekte des Überwachungsraums. Für alle Kameras werden Tiefenbilder der statischen Szene generiert, die in den Rekonstruktionsprozess der VH einbezogen werden. Es wird dabei aus Sicherheitsbetrachtungen heraus Wert auf eine konservative Verarbeitungskette gelegt. Die damit erzeugte VH ergibt eine bessere Approximation der realen dynamischen Objekte als mit einem alternativen Verfahren vom Stand der Technik erreicht werden kann, bei dem die statischen Objekte in den Kamerabildern als Occlusion Masks maskiert und den Silhouettenbildern der dynamischen Objekte hinzugefügt werden.

Das Tracking-Verfahren dieser Dissertation verwendet 3D-Rekonstruktionsdaten als Eingabe. Zusätzlich zur Visuellen Hülle wird Wissen von den statischen Objekten und deren Verdeckungsvolumina herangezogen, um mit den resultierenden Objektverdeckungen im Verfahren umgehen zu können. Der Rekonstruktionsalgorithmus wurde deshalb so abgeändert, dass eine Voxelklassifikation nach konservativen und semantischen Überlegungen erfolgt und die zur Verfügung stehenden Informationen von der Raumbelegung und den gegebenen statischen Verdeckungsvolumina zu jedem betrach-

teten Zeitpunkt k von vier exklusiven Voxelmengen repräsentiert werden. Namentlich sind dies die *filled*-Voxel, welche sämtliche durch statische Objekte permanent belegte Raumbereiche repräsentieren sowie die *occluded*-Voxel, die sensorisch für das gesamte Kamerasystem verdeckt, jedoch initial leer sind und dementsprechend zur Laufzeit auch von dynamischen Objekten (Personen) okkupiert werden können. Weiterhin werden die *empty*-Voxel als unbelegt angenommen. Übrig bleiben die *unknown*-Voxel, welche der generierten Visuellen Hülle entsprechen. Für diese wird davon ausgegangen, dass sie zu den dynamischen (unbekannten) Objekten gehören. In dieser Menge können auch Artefakte, d. h. Pseudoobjekte bzw. leere Volumina, enthalten sein. Die verwendeten Sensoren lassen jedoch keine genauere Analyse bei der Rekonstruktion der Visuellen Hülle zu. Als alternatives Verfahren wurde auch das Voxel Carving zur Rekonstruktion einer Photohülle betrachtet, welches ein Rekonstruktionsprinzip verfolgt, das auf Farbinformationen und Annahmen zu Farbkonsistenzen basiert. Im Ergebnis erwies sich die Photohülle für die betrachtete Roboterarbeitszelle mit sieben Kameras jedoch als zu rechenaufwändig im Verhältnis zu dem geringen Nutzen hinsichtlich der Verbesserung der Objektapproximation.

Für das Personen-Tracking wird ein Partikelfilter eingesetzt [Choset et al., 2006]. Als Objektmodell kommt ein Ellipsoid zum Einsatz, das eine geringe Dimensionalität besitzt. Dies ist vorteilhaft zur Begrenzung der Laufzeit, da bei einer geringeren Dimensionalität auch weniger Partikel für die Abtastung des Zustandsraums benötigt werden als bei Objektmodellen höherer Dimensionalität. In den Experimenten wurde für jede Person ein eigener Partikelfilter mit jeweils 500 Partikeln initialisiert, da sich diese Anzahl als geeignet erwies. Zur Initialisierung wird zu jedem Zeitpunkt k eine 3D-Voxelsegmentierung vorgenommen. Die Segmente werden im Zusammenhang mit den bereits existierenden Filterinstanzen ausgewertet, um Segmente zu identifizieren, für die ein neuer Filter initialisiert werden muss, damit sämtliche Personen im Überwachungsraum getrackt werden. Die Terminierung eines Filters erfolgt, sobald das maximale Ellipsoidgewicht unter einer definierten Schwelle liegt. Alternativ können dafür auch die *unknown*- und *occluded*-Voxel herangezogen werden. Ist deren Anteil innerhalb des Ellipsoids maximalen Gewichts zu gering, so kann davon ausgegangen werden, dass der Filter nicht mehr ausreichend gut auf der Person lokalisiert ist. Zur Lösung des Datenassoziationsproblems in jedem Zeitschritt wird eine Blocking-Methode ähnlich derer in [Canton-Ferrer et al., 2011] eingesetzt, die ein gegenseitiges „Abstoßen“ der Filter im Zustandsraum realisiert. Damit wird eine virtuelle Barriere zwischen dem Einzugsbereich verschiedener Filter erzeugt, ohne dass eine feste Zuordnung der rekonstruierten Voxel zu den Filtern in jedem Zeitschritt k vorgenommen werden muss.

Zur Umsetzung des gewünschten Tracking-Verhaltens werden die Voxel abhängig von ihrem Zustand innerhalb der Likelihood-Funktion gewichtet. Der Menge an *occluded*-

Voxeln kommt eine besondere Bedeutung zu, da sie die nicht einsehbaren, aber initial leeren Volumina des Überwachungsraums repräsentiert und durch ihre positive Gewichtung das Verfolgen dynamischer Objekte durch Verdeckungsvolumina hindurch ermöglicht wird. Das positive Gewicht der *occluded*-Voxel bildet zusammen mit dem positiven Gewicht der *unknown*-Voxel den additiven Gewichtungsterm der Likelihood-Funktion. Diese „Veroderung“ bringt zum Ausdruck, dass nicht nur die Voxel der Visuellen Hülle positiv gewichtet werden, welche die Messung der Personen repräsentieren, sondern auch Teile der Person, die sich teilweise oder vollständig in einem Verdeckungsvolumen befinden. Das Gewicht der *occluded*-Voxel sollte dabei jedoch kleiner sein als das Gewicht der *unknown*-Voxel, die konstant mit 1 gewichtet werden, um ein Stagnieren des Filters in einem Verdeckungsvolumen zu verhindern. Für die gewählten Sequenzen zeigten sich Werte von 0,1 bis 0,3 generell für die *occluded*-Voxel als vorteilhaft, insbesondere wenn es sich um größere Verdeckungsvolumina handelt, bei denen vollständige Objektverdeckungen auftreten können. Diese Werte könnten aber beispielsweise auch lokal unterschiedlich gewählt werden, abhängig von der Größe sowie der konkreten Verdeckungssituation im Raum. Die Ergebnisse der Experimente zeigten, dass der Filter tendenziell beim Eintreten in ein größeres Verdeckungsvolumen dem Objekt mit einer Latenz folgt, wenn das Gewicht der *occluded*-Voxel klein gewählt wird oder er dem Objekt vorauseilt, wenn ein größeres Gewicht zum Einsatz kommt. Umgekehrt kann mit einem kleinen Gewicht ein zügiges Folgen des Filters der Person aus dem Verdeckungsvolumen heraus realisiert werden, während eine höhere Gewichtung hierbei eine Verzögerung des Filters verursacht.

Besteht ein Ellipsoid vollständig aus *unknown*-Voxeln, so ergibt sich durch die konstante Voxelgewichtung mit 1 für das Ellipsoid das maximale normierte Gewicht von 1. An den additiven Term wird eine nichtlineare Verstärkung der *unknown*-Voxel heranmultipliziert (Gain), der Ellipsoide mit einem großen Anteil an *unknown*-Voxeln gegenüber derer mit geringerem Anteil gewichtsmäßig hervorhebt. Damit werden größere Teile von Personen, die sich außerhalb der Verdeckungsvolumina befinden gegenüber den Verdeckungsvolumina selbst bestärkt. Dies soll das folgende gewünschte Filterverhalten hervorbringen: Der Filter soll nur dann in verdeckte Volumina diffundieren, wenn sich die Person auch tatsächlich dort hinein begibt. Umgekehrt wird ein möglichst zügiges Folgen der Person durch den Filter aus dem Verdeckungsvolumen heraus angestrebt. Als Ergebnis ist erkennbar, dass das gewünschte Filterverhalten bei betrachteten Werten von $\psi = 0,1$ bis $\psi = 0,3$ erreicht werden kann. Gleichzeitig ergibt sich eine durchschnittlich vertikalere Vorzugsrichtung des Filters auf den *unknown*-Voxeln, was positiv bewertet wird, da sich die Personen oft aufrecht im Raum bewegen und sich die Ellipsoidform dadurch etwas besser in die Voxeldaten einpasst. Richtig zur Geltung kommt der Mechanismus der nichtlinearen Verstärkung jedoch erst bei höheren Werten von ψ. Diese sind

möglich, ohne zu einer Stagnation des Filters in dem Verdeckungsvolumen zu führen. Die Eruierung von entsprechenden Anwendungsfällen lag außerhalb des Fokus dieser Dissertation.

Letztendlich ist es immer eine Frage des Verhältnisses zwischen den *occluded*- und den *unknown*-Voxeln, die sich in den Partikelellipsoiden befinden und der konkreten Werte ihrer Gewichtung, die über die resultierende Filterbewegung entscheiden. Zusätzliche Faktoren wie die Ausmaße des betrachteten Verdeckungsvolumens im Verhältnis zu dem Objekt, das verfolgt wird sowie die konkrete Bewegung der Person mit all ihren möglichen Dynamiken können verschiedene Effekte entsprechend verstärken oder abmildern. Das Filterverhalten kann weiterhin sowohl durch die Einbeziehung der zwei verbleibenden Voxelzustände *filled* und *empty* bei der Likelihood-Gewichtung beeinflusst werden als auch durch den vorgeschlagenen Kollisionstest. Die beiden Voxelzustände werden jeweils in Form bestrafender Terme mit dem additiven Term multiplikativ verknüpft, um eine Verundung mit dem positiven Teilgewicht zum Ausdruck zu bringen. Sind also neben den positiv gewichteten Voxelzuständen gleichzeitig Voxel der genannten Voxelzustände in den Partikelellipsoiden enthalten, so führt dies zu einer Reduktion der Gewichte, da ihnen eine bestrafende Semantik zugewiesen wird.

Die Bestrafung der *empty*-Voxel zeigte in einer speziellen Teilsequenz des Zweipersonen-Trackings ein erfolgreiches Tracking, weil damit verhindert werden konnte, dass der Filter aus dem Verdeckungsvolumen auf ein Artefakt driftet. Es zeigte sich generell eine länglichere Einpassung der Ellipsoide entlang der Hauptachse der Daten, was eine weitere Verstärkung des Effekts der nichtlinearen Gewichtung mit sich bringt. Weiterin zeigte sich, dass der Exponent b des Bestrafungsterms nicht größer als 1 gewählt werden sollte. Andernfalls sind negative Effekte erwartbar: Der Filter kann stärker auf Voxeln des Zustands *occluded* „hängenbleiben“, was sich am Beispiel des Regals in der Roboterarbeitszelle zeigte. Die Bestrafung der *empty*-Voxel kompensiert zudem teilweise die Streuung des Bewegungsmodells, weil Ellipsoide, die sich zu weit von der Messung entfernen, kein ausreichend hohes Gewicht mehr erhalten, um entsprechend häufig reproduziert zu werden. Die Streuung der Partikel im Raum wird damit implizit vermindert.

Reale Situationen lassen sich mitunter besser abbilden, wenn unplausible Zustände durch Randbedingungen ausgeschlossen werden, wie in diesem Fall durch die Berücksichtigung physischer Barrieren bei der Likelihood-Gewichtung. Die Bestrafung der *filled*-Voxel führte wie erwartet und gewünscht zu einer „Abstoßung“ des Filters durch gegebene statische Objekte. Nach Vorüberlegungen sollten die *filled*-Voxel ein stärkere Bestrafung erfahren als die *empty*-Voxel mit der Begründung der Unmöglichkeit, dass sich ein statisches nicht-leeres Objekt und ein dynamisches Objekt eine Volumeneinheit (Voxel) teilen.

Hingegen kann ein leerer Voxel auch das Ergebnis einer fehlerhaften Objektdetektion sein. Dennoch kann sich auch anders entschieden werden. So zeigten die Ergebnisse, dass auf eine Bestrafung der *filled*-Voxel verzichtet werden kann, wenn gleichzeitig bereits ein Kollisionstest eingesetzt wird. Bei dem Kollisionstest werden Ellipsoide entfernt, die zu weit in statische Objekte hineinragen und damit einen größeren Anteil an *filled*-Voxeln enthalten. Auch kann auf die Bestrafung der *filled*-Voxel verzichtet werden, wenn die statischen Objekte nur ein geringes Volumen besitzen, wodurch die Bestrafung dieser Voxel keinen nennenswerten Effekt hat, oder wenn die Filterbewegung dadurch so stark eingeschränkt wird, dass sich ein Filterverhalten ergibt, das zu einem Tracking-Verlust führen kann. Dies hängt mit den Geometrien der statischen Objekte zusammen und muss anwendungsabhängig entschieden werden. Gleiches gilt für den Kollisionstest.

Der Kollisionstest arbeitet mit Geometrien in der Form von „Kapseln“, die auch als R-Zylinder bezeichnet werden. Zwischen dem Kern eines betrachteten Partikelellipsoids und dem Kern eines zugehörigen propagierten Partikelellipsoids wird ein R-Zylinder aufgespannt, der auf Kollision mit den R-Zylindern der statischen Objekte im Raum getestet wird. Im Falle einer Kollision wird der propagierte Partikel durch einen Partikel neuer Konfiguration ersetzt. Wendet man den Kollisionstest an, so wird damit erreicht, dass die verbleibenden Partikel nur noch begrenzt weit in die statischen Objekte „hineinragen“ können. Mit dieser Vorgehensweise wird eine zu starke Bewegungseinschränkung des Filters verhindert, gleichzeitig wird aber das Durchdringen statischer Objekte durch den Filter vermieden. Bei den Experimenten zeigte sich die gewünschte Wirkungsweise des Kollisionstests am Tisch, der sich im Versuchsaufbau befand. Der Kollisionstest hat eine ähnliche Wirkung wie eine Blocking-Methode als Modellierung einer Barriere. Nur wird damit nicht verhindert, dass sich ein Filter auf die Teilmessung einer anderen Person verschiebt, wo sich im Idealfall bereits ein anderer Filter befindet. Stattdessen soll der Kollisionstest Bewegungseinschränkungen der Realität abbilden, die durch statische Objekte entstehen, indem eine Filterdiffusion auf ungültige und unplausible Zustandsbereiche verhindert wird. Der Kollisionstest löst demnach nicht das Datenassoziationsproblem das beim Tracking mehrerer dynamischer Objekte auftritt, er kann aber dennoch zu einer Verbesserung des Filterverhaltens führen.

Zusammenfassend wurde in dieser Dissertation gezeigt, dass ein erfolgreiches Tracking durch Verdeckungsvolumina möglich ist, selbst wenn es zu vollständigen Objektverdeckungen längerer Dauer kommt. Dies wird durch die Differenzierung verschiedener Voxelzustände mithilfe der Integration von Wissen zu statischen Objekten und ihren Verdeckungsvolumina im Zuge der Rekonstruktion einer Visuellen Hülle erreicht. Damit können verdeckte und initial leere *occluded*-Voxel positiv gewichtet und als Pseudomessung behandelt werden. Zusätzlich kann ein Kollisionstest vermeiden, dass die Partikel durch statische Objekte diffundieren. Alternativ können aber auch die *filled*-Voxel inner-

halb der Ellipsoide bestraft werden, auch wenn damit nicht garantiert werden kann, dass ein Filter durch ein statisches Objekt diffundiert. Eine Bestrafung der *empty*-Voxel kann ebenso einen Nutzen erbringen, was implizit zu einer Verstärkung der *occluded*-Voxel führt. Dies muss jedoch fallabhängig abgewogen werden.

8.2 Diskussion

Die Unterscheidung der Voxelzustände und ihre Gewichtung in der Likelihood-Funktion kann dazu führen, dass sich der Filter orts- und situationsabhängig etwas von der tatsächlichen Objektposition verschiebt. Dies geschieht bei der Bestrafung der *filled*-Voxel, welche Abstoßungseffekte hervorruft, ebenso wie mit der positiven Gewichtung der *occluded*-Voxel, die dazu führt, dass der Filter von dem Verdeckungsvolumen „angezogen“ wird, wenn außerhalb davon nur noch wenige *unknown*-Voxel rekonstruiert werden. Eine Bestrafung der *empty*-Voxel kann ebenfalls einen leichten Drift des Filters auf die anderen Voxelzustände bewirken. Auch der Kollisionstest verändert das Filterverhalten etwas, da er ein Ersetzen kollisionsbehafteter Partikel vornimmt. Wobei letzteres anscheinend einen weniger starken Effekt mit sich bringt als die Bestrafung der *filled*-Voxel, die sichtbar in den Experimenten eine leichte Filterverschiebung erkennen ließ. Dies wird jedoch in Kauf genommen, um die Möglichkeit zu bieten, Personen während vollständiger Objektverdeckungen hinweg weiterverfolgen zu können. Ebenso wird damit verhindert, dass der Filter durch statische Objekte diffundieren kann, was vor allen Dingen beim Mehrpersonen-Tracking hilfreich sein könnte, wenn unerwünschte Artefakte auftreten. Der konkrete Nutzen sollte jedoch noch in konkreten Anwendungsfällen evaluiert werden. Als Gegenmaßnahme zu solchen Verschiebungen erhalten die *unknown*-Voxel eine nichtlineare Verstärkung, um die Filter möglichst stark auf den Messungen dynamischer Objekte (den Voxeln der Visuellen Hülle) zu halten. Die Abweichung von der realen Objektposition wird voraussichtlich in den meisten Fällen gering ausfallen. Zudem kann ein Ellipsoidmodell die Objektposition auch nur annähern, weil es in Gänze nicht detailliert genug ist, um eine exakte Posenbestimmung vornehmen zu können.

Die virtuellen *outside*-Voxel außerhalb des Überwachungsraums wurden als leer angenommen und mit *empty*-Voxeln gleichgesetzt, was zu deren Gewichtung mit 0 oder einer Bestrafung des Gesamtgewichts eines Partikels führte. Bei einer Bestrafung der *outside*-Voxel wich der Filter dem Boden sichtbar aus, was für die betrachteten Teilsequenzen positiv bewertet wurde. Dennoch muss die generelle Bestrafung von *empty*-Voxeln abgewogen werden, weil damit die vom Bewegungsmodell vorgegebene Streucharakteristik des Filters eingeschränkt wird und der Filter sich stärker auf den Messungen fokus-

siert. Ebenso kann der Filter an Verdeckungsvolumina „hängen bleiben“ und abrupten Bewegungen möglicherweise weniger schnell folgen. Deshalb sollten die *outside*-Voxel vorzugsweise separat gewichtet und nicht mit den *empty*-Voxeln gleichgesetzt werden. Beispielsweise könnten die *outside*-Voxel eine Bestrafung erhalten, während die *empty*-Voxel mit 0 gewichtet werden. Für die Betrachtung der Grenzen des Überwachungsraums wäre auch ein Kollisionstest mit den Zellwänden oder statischen Objekten, die außerhalb liegen, denkbar. Dies erhöht jedoch den Berechnungsaufwand.

In dieser Dissertation wurde eine statische Anzahl von Partikeln verwendet, um eine Vergleichbarkeit der Ergebnisse zu gewährleisten. Die Verwendung einer größeren oder geringeren Anzahl als 500 Partikel bringt eine Veränderung des konkreten Filterverlaufs mit sich. Dennoch zeigte sich in nicht dargestellten Voruntersuchungen, dass die Gewichtsparameter auch bei einer Veränderung der Partikelanzahl zu denselben beschriebenen qualitativen Effekten führen, selbst wenn es bei einzelnen Tracking-Verläufen deutliche Unterschiede geben kann.

Befindet sich eine Person in einem größeren Verdeckungsvolumen, so können die Schätzergebnisse deutlich von der tatsächlichen Objektposition abweichen. Dies liegt daran, dass sich die Partikel innerhalb des Verdeckungsvolumens gleichmäßig ausbreiten, aufgrund der identischen Gewichtung aller *occluded*-Voxel. Je größer das Verdeckungsvolumen dabei ist, desto größer kann theoretisch auch die Abweichung des Schwerpunktellipsoids von der tatsächlichen Objektposition sein. Zudem kann es zu einem sichtbaren „Schwingen“ des Schwerpunktellipsoids innerhalb des Verdeckungsvolumens kommen, wenn sich die Person nicht zügig durch diese hindurch bewegt, sondern länger darin verweilt. Dies wird allerdings in Kauf genommen, da in den Verdeckungsvolumina keine Sensordaten gewonnen werden können, die eine Adaption des Bewegungsmodells ermöglichen. Das in Abschnitt 6.3.1 vorgestellte Bewegungsmodell führte in den betrachteten Experimenten zu einer ausreichend guten Personenverfolgung. Dies kann damit erklärt werden, dass die Bewegungen von Personen in einer Arbeitszelle eher wechselhafter Natur sind und mit einer eher geringen Geschwindigkeit einhergehen. Die nicht erfolgreichen Tracking-Durchgänge, bei denen sich die Person(en) außerhalb von Verdeckungsvolumina befand(en) sind eher auf das Datenassoziationsproblem, die Grenzen der verwendeten Blocking-Methode zurückzuführen sowie darauf, dass in dieser Dissertation keine Farbmerkmale in der Likelihood-Funktion eingesetzt wurden, die eine Differenzierung der Voxeldaten bei der Datenassoziation ermöglicht hätten. Die Likelihood-Funktion kann aber durch Farbmerkmale erweitert werden, ähnlich wie es in [Canton-Ferrer et al., 2011] vorgeschlagen wurde.

Die sensorische Abdeckung des Multi-View-Kamerasystems führte in der betrachteten Roboterarbeitszelle nicht zu einem solch großen Verdeckungsvolumen, dass damit

vollständige Objektverdeckungen bei den Videosequenzen auftraten. Der untersuchte Ansatz bietet sich jedoch insbesondere für größere Verdeckungsvolumina an, die im Effekt zu größeren partiellen oder vollständigen Objektverdeckungen führen. Um dies zu zeigen, wurden größere Verdeckungsvolumina künstlich eingefügt. Alternativ hätte auch die für die Rekonstruktion verwendete Kameraanzahl reduziert werden können. Damit hätte sich beispielsweise das Verdeckungsvolumen unterhalb des Tischs vergrößern lassen. Gleichzeitig hätte sich aber auch die Approximationsgüte der 3D-Rekonstruktion verschlechtert. Dies wäre ein möglicher Anknüpfungspunkt für weitere Betrachtungen, in denen die Grenzen des Verfahrens näher untersucht werden könnten, insbesondere auch in Abhängigkeit von variierenden Rekonstruktionsgüten. Hierfür wäre insbesondere eine Simulationsumgebung hilfreich, bei der die Rekonstruktionsdaten mithilfe einer Ground Truth ausgewertet und verglichen werden könnten. In realen Anwendungen des Personen-Trackings sollte die sensorische Abdeckung möglichst optimiert werden, um Verdeckungsvolumina klein zu halten. Dennoch ist eine gute sensorische Abdeckung wie in der SIMERO-Roboterarbeitszelle nicht in jedem Anwendungsfall möglich oder zu teuer, wodurch auch größere Verdeckungsvolumina auftreten könnten. Zudem kann es in industriellen Umgebungen sensorisch überwachte Inseln geben, die durch größere Verdeckungsvolumina voneinander getrennt sind, zwischen denen jedoch ein konsistentes Tracking von Personen erfolgen soll.

Bei den Experimenten mit zwei Personen traten mehrfach Probleme auf, die nicht mit den Objektverdeckungen in Zusammenhang stehen. Ergab sich zwischen den Personen ein geringer Abstand, was meist zu einer Fusion der zugehörigen 3D-Voxelsegmente führte (Merge-Ereignis), so erfolgte vereinzelt nach dem Auseinandergehen der Personen (Split-Ereignis) der Tracking-Verlust einer Person. Häufiger dauerte es nach dem Split jedoch eine gewisse Zeit bis eines der Schwerpunktellipsoide die gerade nicht fokussierte Person wieder „richtig“ verfolgte. Ein Filtertausch trat dabei ebenfalls mehrfach in den Experimenten der Gesamtevaluierung auf (auch wenn dies nicht gezählt wurde). Die Hinzunahme von Farbmerkmalen in der Likelihood-Funktion könnte diesen Problemen entgegenwirken. Dennoch sollte auch eine Verbesserung der Blocking-Methode in Betracht gezogen werden, da in den Bildersequenzen auch mehrfach beobachtet wurde, wie sich die Filter bei einem Merge übereinander positionierten, obwohl die Personen nebeneinander standen.

In den Experimenten wurde das vorgestellte Tracking-Verfahren für das Verfolgen von maximal zwei Personen untersucht. Die Vorgehensweise ist jedoch ebenso auf mehr als zwei Personen übertragbar, ohne dass sich dabei an dem Verfahren etwas ändern müsste. Das Datenassoziationsproblem kann jedoch in bestimmten Situationen schwieriger zu lösen sein, was bedeutet, dass dann ebenfalls Tracking-Verluste oder Filtertausche erwartbar sind. Zu beachten ist die Entstehung weiterer gegenseitiger Objektverde-

ckungen der dynamischen Objekte sowie zusätzliche störende Artefakte, welche die Filter beeinflussen können, wie es für die betrachtete Teilsequenz B gezeigt wurde. Unter Zuhilfenahme einer Simulationsumgebung wären auch hierfür Untersuchungen interessant, welche die Grenzen der Verwendung von 3D-Rekonstruktionsdaten beim Mehrpersonen-Tracking aufzeigen würden.

Als Objektmodell für das Tracking kam ein Ellipsoidmodell zum Einsatz. Möchte man ein komplizierteres Modell wie ein Skelettmodell verwenden, so kann die vorgestellte Likelihood-Funktion prinzipiell auch für einzelne Körperteile eingesetzt werden, sofern diese ebenfalls mit Volumenmodellen modelliert sind und deren Voxelbelegung in der Likelihood-Funktion gewichtet werden soll. Solche Modelle besitzen allerdings eine deutlich höhere Dimensionalität, was den Suchraum und die Berechnungsdauer stark vergrößert. Auch müssten die Einzelgewichte der Körperteile miteinander kombiniert werden. In einer studentischen Arbeit [Munder, 2015] wurde in diesem Kontext (ohne 3D-Rekonstruktion) untersucht, wie bei einem Skelettmodell mit verdeckten Körperteilen umgegangen werden könnte. Dabei ging es insbesondere um die Initialisierung von Posen für den Fall, dass zum Zeitpunkt der Posenbestimmung kein Wissen über eine Vorgängerpose zur Verfügung steht. Die einzelnen Körperteile wurden durch unterschiedlich farbige Stoffe für das Multi-View-Kamerasystem sichtbar gemacht, um die Detektion der Körperteile in den Bildern zu erleichtern. Bei den Untersuchungen zeigte sich mit welchen Herausforderungen die Posenschätzung verbunden ist, wenn der Suchraum entsprechend groß ist und gleichzeitig Informationen bei verdeckten Körperteilen fehlen, wodurch sich nicht alle Ambiguitäten auflösen lassen. Im Stand der Forschung von Kapitel 4 wurden alternative Verfahrenskategorien für das Ziel der Pose Estimation vorgestellt. In neueren Ansätzen für Multi-View-Kamerasysteme wird auf eine explizite 3D-Rekonstruktion verzichtet. Techniken des Deep Learning sind stark in den Fokus gerückt. Für die Pose Estimation in komplexen Situationen mit mehreren Personen und statischen Objekten sollten deshalb solche Verfahren in Erwägung gezogen werden, insbesondere wenn zukünftig mehr geeignete Trainingsdatensätze zur Verfügung stehen.

Die Bildersequenzen wurden mit passiven Farbkameras aufgezeichnet. Es ist empfehlenswert für Fortsetzungsarbeiten den aktuellen Entwicklungsstand alternativer Sensoren jeweils neu zu bewerten. Könnte man nahezu ideale Tiefendaten erzeugen (vgl. Abschnitt 5.5), so würden die Artefakte in den Rekonstruktionen entfallen, die für passive Einzelkameras entstehen. Der vorgestellte Ansatz könnte für Tiefenkameras ebenso zur Anwendung kommen. Auch eine Kombination verschiedener Sensortypen könnte für das Gesamtsystem Vorteile erbringen.

8.3 Ausblick

Verschiedene Anknüpfungspunkte für weiterführende Untersuchungen wurden in Abschnitt 8.2 bereits erwähnt. Einige Themen werden nun konkreter behandelt.

Bei der Likelihood-Gewichtung wurden alle Voxel, die zu einer Voxelmenge gehören und demnach den gleichen Voxelzustand besitzen mit dem gleichen konstanten Gewicht versehen. Es wurde nicht zwischen verschiedenen Größen der Verdeckungsvolumina oder statischen Objekten unterschieden. Die Verwendung lokal unterschiedlicher Gewichte könnte näher untersucht werden. Weiterhin wurden im Rahmen dieser Arbeit ausschließlich statische Objekte als Hindernisse und Verursacher von Verdeckungen betrachtet. Die gewichteten Verdeckungsvolumina veränderten sich damit ebenfalls nicht. Lohnenswert erscheint eine Betrachtung wie sich die vorgeschlagene Likelihood-Gewichtung auf das Filterverhalten auswirkt, wenn sich die Voxelzustände und damit auch die bekannten Verdeckungsvolumina dynamisch verändern. In einer Simulationsumgebung könnten dazu auch sehr unterschiedliche Verdeckungsvolumina mit verschiedenen Ausmaßen und Formen betrachtet werden. Die Datenauswertung ließe sich in einer solchen Simulation effizienter bewerkstelligen, weil mit einer gegebenen Ground Truth für die Position und Pose der Personen Ergebnisse stärker quantifiziert werden könnten.

Für einen realen Einsatz müsste es jedoch möglich sein, diese dynamischen Verdeckungsvolumina (deren *occluded*-Voxel) auch zu jedem Zeitpunkt zu ermitteln, ebenso wie die *filled*-Voxel dieser Objekte. Herausfordernd ist dabei, dass dynamische Gegenstände auch Änderungen in den Bildern hervorrufen, die beim Background Subtraction zu detektierten Pixeln führen, was eigentlich nur für die zu verfolgenden Personen erwünscht ist. Statische Objekte wären damit zumindest teilweise auch in den *unknown*-Voxeln der Visuellen Hülle enthalten. Zusätzliches Wissen, z. B. von zyklischen Bewegungen und dem Aussehen dynamischer Gegenstände, könnte dabei hilfreich sein, um ein entsprechendes Verfahren umzusetzen.

Die Likelihood-Funktion könnte um eine Bewertung kolorierter Oberflächenvoxel (nach deren Einfärbung) erweitert werden, wie es in [Canton-Ferrer et al., 2011] erfolgt. Die Fusion von Farben im 3D-Raum ist jedoch mit Informationsverlusten behaftet, weil Farben bei ihrer Rückprojektion aus den Kamerabildern in den 3D-Raum für den gesamten zugehörigen Sichtkegel eines Pixels angenommen werden müssen. Wo sich aber die tatsächliche Objektoberfläche in dem Sichtkegel befindet, die zur Farbentstehung geführt hat, ist für (dynamische) Objekte in passiven Kameras nicht bekannt. Abweichungen zu den rekonstruierten Oberflächenvoxeln sind zu erwarten. Für die Einfärbung einer Visuellen Hülle basierend auf der Farbfusion mehrerer Kamerabilder im 3D-Raum sind dahingehend Annahmen zu treffen, die Fehler mit sich bringen. Aber

auch wenn die Ergebnisse fehlerbehaftet sind, so könnten Farbmerkmale zur Reduktion von Datenassoziationsproblemen beitragen. Zudem ist es auch möglich, Farbmerkmale nur aus den Kamerabildern heranzuziehen, ohne die Visuelle Hülle selbst zu kolorieren. Die Filter sollten sich damit nach einer Merge-Split-Situation wahrscheinlicher auf den richtigen Objekten repositionieren können. Filtertausche sollten in Gänze seltener auftreten. Dies setzt allerdings voraus, dass sich das Aussehen der zu trackenden Personen in den Bildern ausreichend voneinander unterscheidet.

Für das Tracking wurde ein Partikelfilter gewählt, da bei den gegebenen Beobachtungen der Visuellen Hülle mit sehr unterschiedlichen Güten und Ambiguitäten zu rechnen ist. Dies bestätigte sich beispielsweise in Situationen, in denen rekonstruierte Artefakte zu einer temporären Aufteilung der Partikel eines Filters auf mehrere 3D-Voxelsegmente der *unknown*-Voxel (Visuelle Hülle) führten. Mit dem Partikelfilter konnte bei geeigneter Parametrisierung auch in solchen Situationen das Tracking aufrechterhalten werden. Die beste Schätzung (repräsentiert durch das Schwerpunktellipsoid aller Partikel) entfernte sich dabei jedoch mitunter deutlich von der tatsächlichen Position der Person. Hierfür könnte man ein Verfahren des Partikel-Clusterings untersuchen, bei welchem temporär mehrere beste Schätzungen betrachtet und beispielsweise bei Anwendung einer Blocking-Methode zur Lösung des Datenassoziationsproblems berücksichtigt werden.

In den Experimenten wurde ein konstantes Ellipsoidvolumen für zwei Personen ähnlicher Größe (ca. 170 cm) vorgegeben. Alternativ könnte ein Automatismus zur Abschätzung des eingenommenen Volumens unterschiedlicher Personen integriert werden, welcher bei der Filterinitialisierung zum Tragen käme oder sich im Verlauf des Trackings auf die Personen durch eine statistische Analyse der Rekonstruktionsdaten adaptiert.

Für einen realen Einsatz des Gesamtsystems könnte weiterhin die Objektdetektion optimiert werden, um entstehende Fehler in der 3D-Rekonstruktion zu reduzieren, die auf Störungen der Objektdetektion zurückzuführen sind. In [Ascenso et al., 2020] werden leistungsfähige Algorithmen für die Silhouettenextraktion miteinander verglichen und im Hinblick auf deren Einsatz für die Rekonstruktion einer Visuellen Hülle bewertet. Als Ergebnis werden die Resultate der Algorithmen „FgSegNet v2“ (Background Subtraction), „DeepLabv3+ JFT“ (semantische Segmentierung) sowie „Djelouah 2013“ (Multi-View-Segmentierung) hervorgehoben, deren nähere Betrachtung sinnvoll erscheint.

Optimierungspotential besteht auch bezüglich der Laufzeit für die gesamte Verarbeitungskette. Laufzeitbetrachtungen wurden in dieser Dissertation nur für das Tracking-Verfahren vorgenommen. Für den Tracking-Algorithmus selbst sowie den Kollisionstest wurde auf eine effiziente Umsetzung geachtet. Der betrachtete Zustandsraum geringer Dimensionalität ist insbesondere für den Einsatz von Partikelfiltern sehr wichtig.

Denn einerseits hat der Partikelfilter eine lineare Laufzeit in der Anzahl an Partikeln. Andererseits sollte für eine vorgegebene Ortsauflösung/Genauigkeit die Partikelanzahl exponentiell mit der Dimension des Zustandsraums wachsen. Die 3D-Rekonstruktion macht ebenfalls einen Großteil der gesamten Laufzeit je Frame aus. Verschiedene Verfahren für echtzeitfähige 3D-Rekonstruktionen wurden in dieser Dissertation genannt (vgl. Abschnitt 2.3) und könnten mit dem vorgestellten Tracking-Verfahren kombiniert werden.

In der Einleitung wurden Verfahrensklassen der Sensordatenverarbeitung vorgestellt. Die betrachtete Zustandsschätzung könnte ein Baustein zur Realisierung darauf aufbauender Komponenten wie eine trajektorienbasierte Mustererkennung sein, mit dem Ziel, semantisches Wissen zu den Aktivitäten im Überwachungsraum zu gewinnen und Anomalien zu detektieren. Zudem könnte beispielsweise für die Evaluierung von 3D-Rekonstruktionsdaten durch Plausibilisierungsfunktionen nach [Kuhn, 2012] die betrachtete Zustandsschätzung eine zusätzliche Plausibilisierungsfunktion realisieren, um in Gänze mehr Sicherheit bei der Bestimmung von Raumbelegungen zu erreichen.

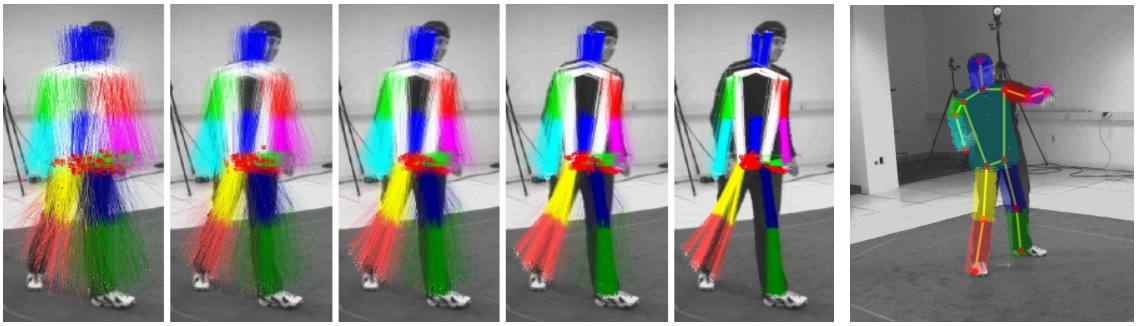
Kapitel 9

Anhang

9.1 Beispiele rekonstruktionsbasierter Tracking-Verfahren

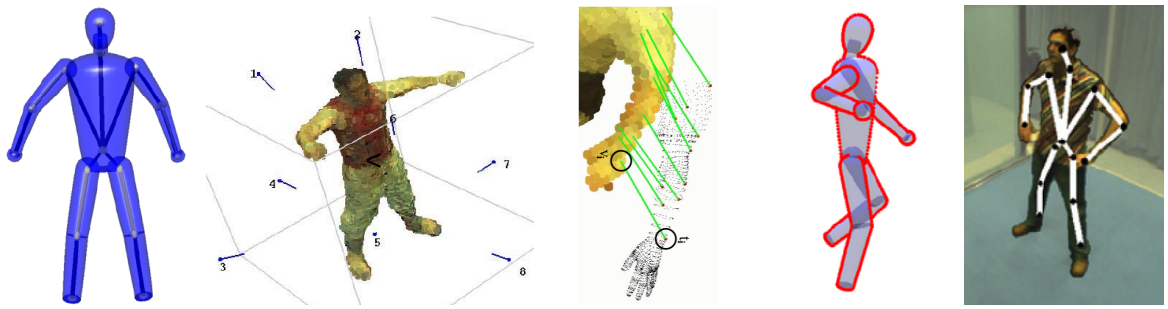
In diesem Abschnitt werden exemplarisch und bewertungsfrei ausgewählte Tracking-Ansätze beschrieben, um die Vorgehensweisen bei der Zustands- bzw. Parameterschätzung zu veranschaulichen. Alle Verfahren verwenden dabei Rekonstruktionsdaten einer Visuellen Hülle als Eingabe.

In dem Ansatz von [Canton-Ferrer et al., 2009], dargestellt in Abb. 9.1, wird von jedem Partikel eine Instanz einer Skelettmodellparametrisierung repräsentiert (ähnlich wie in [Caillette, 2006]). Zum Einsatz kommt ein Annealed-Partikelfilter-Schema, um ein Stagnieren in lokalen Minima zu vermeiden und um die Anzahl der Partikel verringern zu können. Abb. 9.1(a) zeigt solch einen Annealing-Verlauf. Zur Verdeutlichung sind die Gliedmaßen unterschiedlich eingefärbt. Als 3D-Likelihood wird die Übereinstimmung zwischen der Parametrisierung und einer voxelbasierten Visuellen Hülle berechnet. Dazu wird der Körper durch abgeschnittene Kegel modelliert, sodass



(a) (b)

Abb. 9.1: Ansatz und entnommene Bilder aus [Canton-Ferrer et al., 2009, S. 2,4]. Jedes Partikel repräsentiert die Parametrisierung eines Skelettmodells. Zur Schätzung wird ein Annealed-Partikelfilter verwendet. Der Verlauf eines Annealing-Prozesses ist in (a) dargestellt. Zur besseren Veranschaulichung sind die Körperteile unterschiedlich eingefärbt. Das Ergebnis einer Schätzung wird in (b) gezeigt. Das Kamerabild ist von den rekonstruierten Voxeln sowie der geschätzten Pose des Skelettmodells überlagert.



(a) (b) (c) (d) (e)

Abb. 9.2: Ansatz und entnommene Bilder aus [Kehl et al., 2005, S. 3,4], [Kehl und Gool, 2006, S. 6,7,15]. Zu sehen ist das verwendete Körpermodell (a) und die Rekonstruktion einer kolorierten Visuellen Hülle (b). Für die Abstandsbestimmung zwischen einem parametrisierten Modell und den Voxeldaten in der Likelihood-Funktion werden Punkte auf dem Oberflächenmodell gesampelt, denen Voxel zugewiesen werden (c). Durch die Projektion bestimmter Punkte können die Occluding Contours (rot) in den Kamerabildern bestimmt werden (d), was einen Umgang mit Objektselbstverdeckungen ermöglicht. Eine geschätzte Pose ist in (e) zu sehen.

ein Volumenverschnitt zwischen Rekonstruktion und Körpermodell berechnet werden kann. 3D-Oberflächendistanzwerte gehen ebenso in die Likelihood-Funktion ein. In Abb. 9.1(b) ist das Tracking-Ergebnis für ein Frame zu sehen. Das Kamerabild wird mit den rekonstruierten Voxeln sowie dem Skelettmodell der geschätzten Parametrisierung überlagert.

Ein weiteres Verfahren zur rekonstruktionsbasierten Pose Estimation ist das aus [Kehl und Gool, 2006]. In Abb. 9.2(a) ist das Körpermodell, bestehend aus einem Skelettmodell und einem Oberflächenmodell, das aus Superellipsoiden zusammengesetzt ist, zu sehen. Für das Tracking wird eine Stochastic-Meta-Descent-Optimierung verwendet, um die Pose zu finden, welche am besten die Kamerabilder „matcht“. Das stochastische Sampling macht das Verfahren robust gegenüber lokalen Minima und führt zu einem verringerten Berechnungsaufwand. Aus den Kameradaten wird eine kolorierte Visuelle Hülle generiert, wie in Abb. 9.2(b) gezeigt. Für die Einpassung des Modells in die Rekonstruktionsdaten werden gesampelte Modellpunkte verwendet, so wie in Abb. 9.2(c) veranschaulicht. In der Likelihood-Funktion wird die Summe der Distanzen der Modellpunkte zu den entsprechenden rekonstruierten Voxeln gebildet. Durch die Farbinformationen kann der Aufwand bei der Suche nach Voxel-Punkt-Korrespondenzen reduziert werden. Weiterhin werden auch Konturen ausgewertet. Hierfür erfolgt eine Projektion der sichtbaren Punkte des Körpermodells in die Kameras, was die sogenannten „Occluding Contours“ ergibt, wie in Abb 9.2(d) dargestellt. Die Übereinstimmung dieser Konturen mit detektierten Kanten der aufgezeichneten Bilder fließt ebenfalls in die Likelihood-Gewichtung ein. Dadurch können Selbstverdeckungen in den Kamerabil-

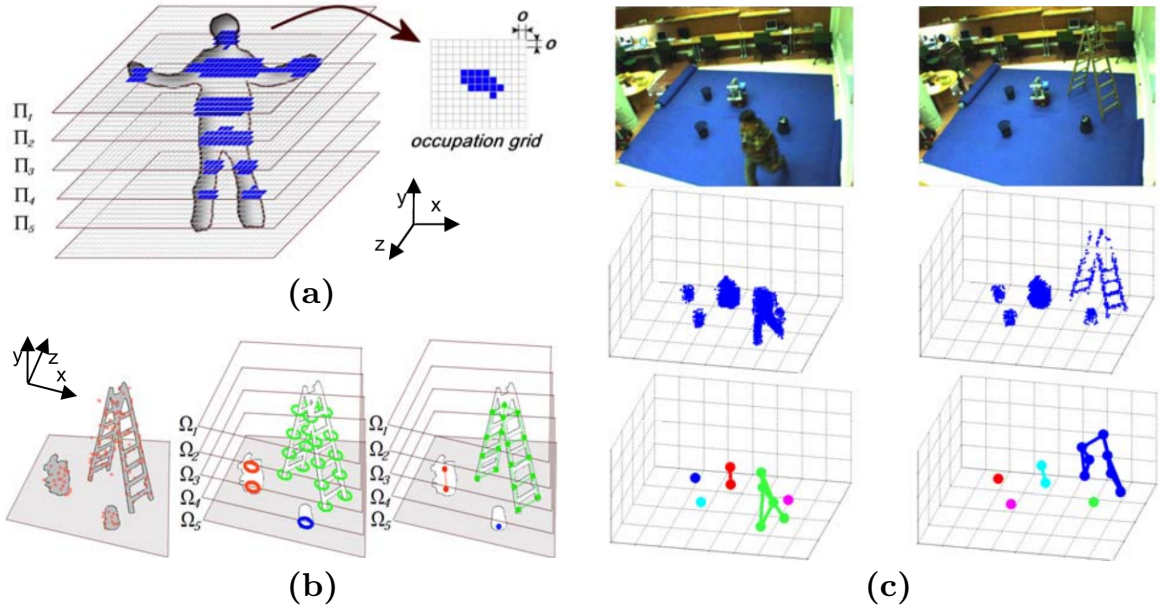


Abb. 9.3: Ansatz und entnommene Bilder aus [Marrón et al., 2009, S. 2,4,5]. Bei dem Verfahren werden als Messung Occupancy Grids verwendet. Diese werden aus Schnittebenen Π_i von einer Visuellen Hülle gewonnen (a). Durch den Zustandsraum wird ebenfalls eine diskrete Menge von Schnittebenen Ω_j gelegt. Bei der Zustandsschätzung werden alle Partikel eines Partikelfilters der jeweils nahesten Schnittebene Ω_j zugeordnet und anschließend geclustert (b). Die Cluster-Zentroiden aller Schnittebenen werden miteinander verbunden, wodurch Gruppierungen von Zentroiden entstehen, welche die einzelnen Objekte repräsentieren. Die Ergebnisse zweier Frames sind in (c) dargestellt.

dern behandelt und die Ambiguitäten reduziert werden. Ein Tracking-Ergebnis ist in Abb. 9.2(e) zu sehen.

Einen Ansatz zum Verfolgen verschiedener statischer und dynamischer Objekte findet man in [Marrón et al., 2009], dargestellt in Abb. 9.3. Bei diesem Verfahren kommen keine Objektmodelle zum Einsatz, sondern es wird ein anderes Konstrukt gewählt, um die Lokalisierung der Objekte im Raum zu bestimmen. Für das Beobachtungsmodell wird eine Visuelle Hülle rekonstruiert. Allerdings wird diese nicht wie bei den anderen Verfahren gänzlich in Form einer Voxeldatenstruktur verarbeitet. Sondern es wird eine diskrete Menge von Schnittebenen Π_i durch die Hülle gelegt, so wie in Abb. 9.3(a) gezeigt. Diese Ebenen werden ebenfalls diskretisiert, sodass die gemessene Raumbelegung in Form von „Occupancy Grids“ kodiert wird. Für alle Objekte im Raum wird nur ein einzelner Partikelfilter verwendet. Die Objekte werden von unterschiedlichen Modalitäten der approximierten A-Posteriori-Dichte repräsentiert. Dazu muss entsprechend sichergestellt werden, dass die Partikel diese Modalitäten auch ausreichend gut abbilden. Hierfür wird als Algorithmus ein Extended-Partikelfilter mit Clustering-Prozess (XPFCP) vorgeschlagen. Vor dem Reinitialisierungsschritt erfolgt dabei eine „Measurement Equalization“, bei der die Partikel umverteilt werden, sodass Redundanzen entfernt werden und die Par-

(a) (b) (c)

Abb. 9.4: Ansatz und entnommene Bilder aus [Canton-Ferrer et al., 2011]. Für das Beobachtungsmodell wird die Visuelle Hülle rekonstruiert (a) und nachträglich koloriert (b). Dargestellt sind die besten Schätzungen der Ellipsoidmodelle zweier Partikelfilter (c).

tikel besser die A-Posteriori-Dichte repräsentieren. Für die Lokalisierungsschätzung der Objekte wird ein Clustering der gewichteten Partikel vorgenommen. Der Zustandsraum wird dazu entlang der Höhenkoordinate diskretisiert, wie in Abb. 9.3(b) zu sehen. Alle Partikel werden dann auf die nächstgelegene Ebene Ω_j projiziert. Anschließend wird für alle Partikel einer Ebene ein k-Means-Clustering vorgenommen. Als Ergebnis erhält man eine Menge von Clustern mit Zentroiden in jeder Ebene. Im letzten Schritt werden die Zentroiden aller Ebenen in einem Boolschen und bidirektionalen Konnektivitätsprozess unter Auswertung der Euklidischen Distanz miteinander verbunden. Jede verbundene Menge von Zentroiden soll ein einzelnes Objekt repräsentierten. Die Ergebnisse zweier Frames sind in Abb. 9.3(c) dargestellt.

Das letzte Verfahren, das konkret vorgestellt werden soll, ist das Verfahren aus [Canton-Ferrer et al., 2011]. Damit soll das Verfolgen mehrerer Personen in einer Umgebung mit statischen Objekten vorgenommen werden können. Für jede Person wird ein eigener Partikelfilter eingesetzt. Das Objektmodell ist jeweils ein Ellipsoid. Für das Beobachtungsmodell wird eine Visuelle Hülle rekonstruiert, so wie in Abb. 9.4(a) dargestellt. Diese Visuelle Hülle wird nachträglich koloriert, gezeigt in Abb. 9.4(b). Die Likelihood-Funktion besteht aus zwei Teilen, die additiv verknüpft sind. In dem ersten Teil wird die Überlappung zwischen Ellipsoid und belegten Voxeldaten berechnet. Im zweiten Teil wird für das Ellipsoid ein Histogramm erzeugt, welches mit einem Referenzhistogramm der Person verglichen wird. In Abb. 9.4(c) sind die besten Schätzungen (Ellipsoide) zweier Partikelfilter in ein Kamerabild gerendert. Alternativ zu dem Partikelfilteransatz wird in [Canton-Ferrer et al., 2011] auch ein spezielles oberflächenbasierten Sampling vorgestellt. Letzteres ist für diese Arbeit jedoch weniger von Interesse, da in betrachteten verdeckten Verdeckungsvolumina keine Informationen zu den Objektoberflächen erfasst werden können.

9.2 Konzept der Photohülle

Die folgenden Darstellungen zur Photohülle wurden in [Ober-Gecks et al., 2016] veröffentlicht und werden in ähnlicher Weise wiedergegeben.

Shape-from-Silhouette-Verfahren (SfS) zur Rekonstruktion Visueller Hüllen (VH) basieren auf der Verschneidung rückprojizierter Silhouettenbilder in den 3D-Raum (vgl. Abschnitt 5.3). Dabei werden Farb- und andere Bildmerkmale nicht berücksichtigt, die jedoch das Potential besitzen, eine feinere Rekonstruktion mit besserer geometrischer Objektapproximation als die VH zu liefern. Es existieren verschiedene Ansätze zum nachträglichen Einfärben einer VH. Diese beeinflussen das geometrische Rekonstruktionsergebnis selbst jedoch nicht.

Verfahren zur Rekonstruktion einer Photohülle beruhen auf einem anderen Prinzip als dem SfS und beziehen Farbmerkmale in den Rekonstruktionsprozess mit ein. Solche Verfahren werden auch als **Color Reconstruction** [Seitz und Dyer, 1999] oder als **Space Carving** [Kutulakos und Seitz, 2000] bezeichnet. Der Begriff Space Carving bezieht sich auf den eigentlichen Prozess des „Zurechtschneidens“ des Raumes. Da typischerweise eine Voxeldatenstruktur verwendet wird, ist das Verfahren auch als **Voxel Carving** bekannt. Das Rekonstruktionsergebnis selbst wird Photohülle genannt.

Die prinzipielle Vorgehensweise bei der Rekonstruktion ist folgende: Beginnend mit einem vollständig belegten Voxelraum werden solange Voxel iterativ entfernt (gecarvt) bis sich die Geometrie der Szenenobjekte als Resultat ergibt. Die Entscheidung für

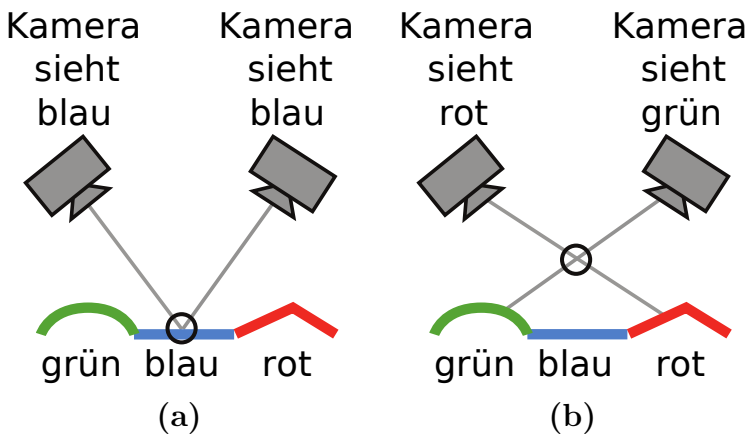


Abb. 9.5: Prinzip eines Farbkonsistenztests betrachtet für zwei Kameras und eine Voxelposition (kleiner schwarzer Kreis). Die Kameras erfassen einen gleichen Farbwert (a). Es wird angenommen, dass sich das Voxel auf der Oberfläche eines Objekts befindet. Das Voxel wird nicht entfernt (gecarvt). Erfassen die Kameras unterschiedliche Farbwerte (rot und grün), so wird angenommen, dass es sich bei dem Voxel nicht um ein Oberflächenvoxel handeln kann (b). Das Voxel wird entfernt (als „leer“ markiert). Abbildungen entnommen aus [Ober-Gecks et al., 2016, S. 9], angelehnt an [Slabaugh et al., 2001, S. 6].

das Entfernen von Voxeln wird basierend auf ihren assoziierten Farbinformationen in den Bildern getroffen. Für jedes Voxel wird überprüft, ob es zu einer Objektoberfläche gehört. Nach einer zugrundeliegenden Annahme ist das dann der Fall, wenn das betrachtete Voxel in allen Kamerapixeln, auf die es tatsächlich abbildet (für die es nicht verdeckt ist), die gleiche Farbe erzeugt, sprich eine **Farbkonsistenz** (engl. Color Consistency) vorliegt. Die Überprüfung der Hypothese einer Farbkonsistenz wird **Farbkonsistenztest** (FT) genannt. Hierfür wird ein **Farbkonsistenzkriterium** benötigt, welches die Übereinstimmung der Farben aller betrachteten Pixel berechnet. Mögliche Konsistenzkriterien sind in [Slabaugh et al., 2004] zu finden. Sind die Farben zu unähnlich, so wird das entsprechende Voxel als inkonsistent klassifiziert und entfernt, mit der Annahme, dass es nicht auf einer Objektoberfläche liegen kann. Andernfalls bleibt das Voxel, zumindest für diesen Iterationsschritt, in der Rekonstruktion erhalten. Das Prinzip des FTs wird in Abb. 9.5 veranschaulicht.

Um einen FT für ein Voxel durchführen zu können, müssen vorab die Pixel aus der Menge seiner Projektionspixel ermittelt werden, für welche das Voxel auch tatsächlich sichtbar ist. Dieser Vorgang wird als **Sichtbarkeitstest** (engl. Visibility Test) bezeichnet. Ein Pixel gehört dann dazu, wenn sich aus Sicht des Voxels auf dem Strahl in Richtung Pixel kein anderes belegtes Voxel befindet, so wie in Abschnitt 3.2.1 für die Punktverdeckungen beschrieben. Durch das iterative Abarbeiten des Voxelraums von außen nach innen wird die Anzahl der durch den Sichtbarkeitstest bestimmten Pixel für die noch nicht entfernten Voxel nur zunehmen, aber nicht abnehmen, da einmal entfernte Voxel nicht wieder hinzugefügt werden dürfen. Mit dem Entfernen eines jeden Voxels muss der Sichtbarkeitstest für alle Voxel erneut ausgeführt werden, da sich die Sichtbarkeit der verbleibenden Voxel verändert haben kann und dies bei den folgenden FTs entsprechend berücksichtigt werden muss.

9.2.1 Eigenschaften

Es gibt bei der Anwendung von Space-Carving-Algorithmen zwei Schlüsselfaktoren, welche die Qualität der rekonstruierten Objekte beeinflussen [Slabaugh et al., 2004]. Der erste ist die Berechnung der Voxelsichtbarkeiten. [Steinbach et al., 2000] zeigen, dass bessere Rekonstruktionsergebnisse zu erwarten sind, wenn die Projektionen der Voxel in die Bilder präzise berechnet werden. Ein approximativer Ansatz, der beispielsweise lediglich die Projektion des Voxelzentrums verwendet, führt zu sichtbar schlechteren Ergebnissen. Dies wird in [Slabaugh et al., 2004] bestätigt. Der zweite Faktor ist der konkrete Farbkonsistenztest, der eingesetzt wird, um die Ähnlichkeit der Farben für ein Voxel in den Bildern zu bewerten. Wenn ein Voxel eigentlich zum Objektmodell gehört, aber durch den FT als inkonsistent klassifiziert und aus der Rekonstruktion entfernt

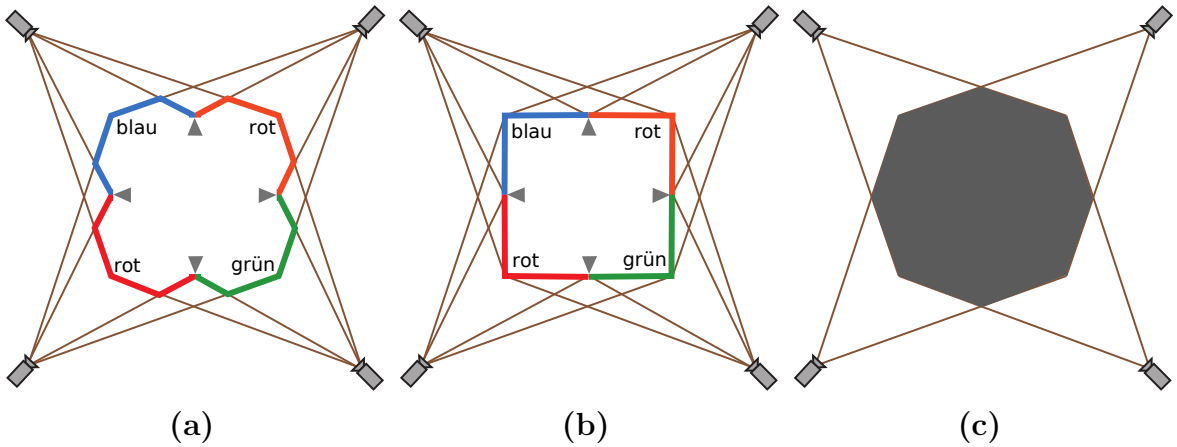


Abb. 9.6: Ambiguitäten der Photohülle bei Bewertung der Photointegrität. Das Kriterium der Photointegrität kann für verschiedene Objekte zugleich erfüllt sein. Die Objekte in (a) und (b) erzeugen die gleichen Bilder aus denen einen Photohülle rekonstruiert wird. Demnach muss die erzeugte Photohülle nicht dem gewünschten Objekt aus (a) oder (b) entsprechen. Nichtsdestotrotz kann die Photohülle prinzipbedingt eine bessere Approximation von Objektgeometrien ermöglichen als die Visuelle Hülle (c). Abbildungen entnommen aus [Ober-Gecks et al., 2016, S. 10], in Anlehnung an [Kutulakos und Seitz, 2000, S. 208].

wird, so ist der FT für dieses Voxel zu streng. Solch ein Voxel wird fälschlicherweise als transparent (leer) markiert, was die Sichtbarkeitstests verbleibender Voxel beeinflusst und damit wiederum inkorrekte Entscheidungen bei deren FTs nach sich ziehen kann. Ist ein FT hingegen zu schwach und wird ein Voxel, das nicht zum Modell gehört, als konsistent klassifiziert, so entsteht ein überflüssiges Voxel. Dieses „versperrt“ in den nachfolgenden Sichtbarkeitstests fälschlicherweise die Sicht auf die Objektoberfläche und führt zu einer verfrühten Konvergenz des Algorithmus. Ein gewähltes Farbkonsistenzkriterium kann zu beiden Effekten an unterschiedlichen Stellen der Rekonstruktion führen, wodurch im Ergebnis einerseits Löcher und andererseits störende überflüssige Voxel entstehen. Beide Fehlerarten beeinflussen die geometrische Approximation negativ.

Bei den FTs wird zwischen monotonen und nicht-monotonen Farbkonsistenzkriterien unterschieden [Slabaugh et al., 2004]. Wenn ein Voxel für eine Menge an Projektionspixeln, für die es „sichtbar“ ist, inkonsistent ist und dies ebenso für jede beliebige Obermenge an „sichtbaren“ Projektionspixeln gilt, so wird das betrachtete Farbkonsistenzkriterium als monoton bezeichnet. Mit monotonen Farbkonsistenzkriterien sollen Voxel niemals fälschlicherweise verworfen werden, die im finalen Modell konsistent wären. Umgekehrt dürfen Voxel zunächst als konsistent und erst in einer späteren Iteration als inkonsistent klassifiziert werden. Bei einem monotonen Konsistenzkriterium kann – unter Verwendung korrekter Sichtbarkeitsinformationen zu dem Voxel – garantiert werden, dass man eine Photohülle erhält, die ein eindeutiges photokonsistentes Modell darstellt [Kutulakos und Seitz, 2000]. Im Gegensatz dazu kann der Algorithmus bei Verwendung eines nicht-

monotonen Farbkonsistenzkriteriums (mit gleichen Schwellenwerten) zu verschiedenen Rekonstruktionen (Photohüllen) konvergieren, abhängig davon, in welcher Reihenfolge die Voxel verarbeitet werden. Obwohl die Verwendung monotoner Kriterien sehr sinnvoll erscheint, können auch nicht-monotone Kriterien zu guten Rekonstruktionsergebnissen führen. Ein einfaches nicht-monotones Kriterium ist die Standardabweichung der Farbwerte der sichtbaren Projektionspixel eines Voxels. Diese wird im FT mit einem festen Schwellenwert verglichen. Ein Überblick zu monotonen und nicht-monotonen Farbkonsistenzkriterien wird gegeben in [Slabaugh et al., 2004].

In [Kutulakos und Seitz, 2000] wird beschrieben, dass die Photohülle – die Menge aller photokonsistenten Voxel – die knappste Approximation der wahren Geometrie ermöglicht, die alleinig aus der Verwendung von Farbbildern abgeleitet werden kann. Dennoch hängt das Ergebnis stark von dem verwendeten Farbkonsistenzkriterium ab und zusätzlich noch von den gesetzten Schwellenwerten. Somit können auch Ergebnisse entstehen, die weit von einer guten Approximation der Geometrie entfernt liegen. Zur qualitativen Bewertung einer rekonstruierten Photohülle kann das Kriterium der **Photointegrität** (engl. Photo Integrity) eingesetzt werden. Die Projektion der Photohülle in die Kameras soll dabei möglichst genau die originalen Eingabebilder, unter Berücksichtigung der Auflösung und Quantisierung, reproduzieren. Ein Nachteil dieses Maßes ist, dass die Photointegrität durch verschiedene Objektgeometrien gleichermaßen erfüllt werden kann und damit Ambiguitäten mit der Rekonstruktion einhergehen (vgl. Abb. 9.6(a) und (b)). Die Verwendung einer größeren Kameramenge kann dabei helfen, die rekonstruierte Geometrie näher an die reale heranzubringen.

Mit der Photohülle möchte man eine bessere geometrische Approximation erzielen als mit der VH. Ob dies möglich ist, hängt jedoch von verschiedenen Parametern ab. Neben den gewählten Kameraperspektiven spielt die Texturierung der Objekte eine große Rolle. Eigene Ergebnisse hierzu werden in Abschnitt 9.3 diskutiert. Zudem beeinflusst die Voxelgröße die Rekonstruktion. Je größer die Voxel sind, umso häufiger treten fälschlicherweise berechnete Farbinkonsistenzen auf, wodurch Teile von Objekten in der Rekonstruktion fehlen können. Auch kann eine schlechte Kamerakalibrierung die FTs negativ beeinflussen, weil damit teilweise falsche Projektionspixel für die Voxel herangezogen werden.

Als Einschränkung ist weiterhin anzumerken, dass die Objekte als diffus reflektierend angenommen werden, also Lambertschen Strahlern entsprechen sollen. Denn nur so können Oberflächenpunkte oder -teile aus verschiedenen Kameraperspektiven auch zu gleichen Farbwerten führen (wenn man von produktionsbedingten Unterschieden der Sensoren und häufig fehlenden Farbkalibrierungen absieht). Es gibt verschiedene Ansätze, welche versuchen, die genannte Annahme aufzuweichen, da diese in der Realität oft

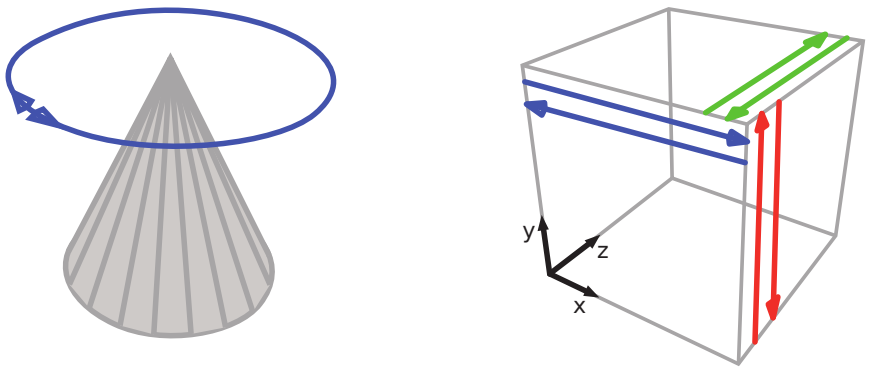
nicht eingehalten werden kann. Eine weitere implizite Annahme besteht bei den meisten Space-Carving-Algorithmen darin, dass die Pixelauflösung größer als die Voxelauflösung ist, ein Voxel also auf eine Anzahl an Pixeln in mindestens zwei Kameras projiziert.

Zur Berechnung Visueller Hüllen wurden in dieser Dissertation spezielle Algorithmen vorgestellt, die eine konservative Abschätzung bezüglich einer oberen Grenze des Objektvolumens treffen, sofern man ideale Silhouettenbilder und eine ideale Kamerakalibrierung als gegeben annimmt (vgl. Abschnitt 5.3 und 5.4). Solch eine Konservativität ist jedoch beim Space Carving aufgrund ihrer Parameterabhängigkeit nur schwer zu gewährleisten.

Trotz genannter Nachteile kann das Space Carving geeignet sein, um die tatsächlichen Objektgeometrien gut zu approximieren. Zudem bringt das Ergebnis einer Photohülle eine inhärente Kolorierung mit sich (Visuelle Hüllen werden nachträglich eingefärbt). Die Anwendung eines Space-Carving-Algorithmus für ein Überwachungsszenario wurde daher für diese Dissertation näher untersucht.

9.2.2 Literaturübersicht

Den Flaschenhals beim Voxel Carving bilden die notwendigen Sichtbarkeitstests jeder Iteration. Ein naiver Ansatz wird in [Slabaugh et al., 2001] vorgestellt. Beginnend mit einem gefüllten Voxelraum werden in jeder Iteration der Hauptschleife alle verbliebenen Voxel auf Farbinkonsistenz hin geprüft und entsprechend aus der Rekonstruktion ent-



(a) (b)

Abb. 9.7: Voxelsichtbarkeiten: Ordinal Visibility Constraint und ein Sweeping-Sichtbarkeitstest. Das Ordinal Visibility Constraint wird erfüllt, wenn die Kameras mit Blick auf das Objekt (hier: Kegel) um das Objekt rotierend angeordnet sind (a). Dadurch wird kein expliziter Sichtbarkeitstest benötigt und der Voxelraum kann in nur einer Iteration abgearbeitet werden. Das Objekt kann aber nicht von allen Seiten rekonstruiert werden. Hingegen wird die Sichtbarkeit der Voxel in den PVSC und FVSC-Algorithmen in jeder Iteration durch ein Sweeping entlang der Achsen (in sechs Richtungen) des Voxelraums bestimmt (b). Abbildungen entnommen aus [Ober-Gecks et al., 2016, S. 11] und [Zwicker, 2013, S. 28].

fernt, bis keine Farbinkonsistenzen mehr auftreten. Jedes betrachtete Voxel wird zur Ermittlung seiner Projektionspixel in sämtliche Kameras (erneut) projiziert. Anschließend wird der Sichtbarkeitstest ausgeführt. Für die Teilmenge der Projektionspixel, in denen das Voxel sichtbar ist, erfolgt anschließend der Farbkonsistenztest. Ist die Farbe eines Voxels inkonsistent, so wird es verworfen.

Der beschriebene Algorithmus hat sehr hohe Berechnungskosten. Diese können reduziert werden durch eine Begrenzung der Kamerapositionen, genannt **Ordinal Visibility Constraint** [Seitz und Dyer, 1999], veranschaulicht in Abb. 9.7(a). Alle Kameras müssen hierbei hinter einer **Parting Plane** (Trennebene) platziert werden, sodass die Voxel in nur einer Iteration in einer festen Reihenfolge abgearbeitet werden können. Nachteilig daran ist die unvollständige Rekonstruktion aufgrund unzureichender Perspektiven. Im Vergleich dazu führen Algorithmen des **Partial Visibility Space Carving** (PVSC) und des **Full Visibility Space Carving** (FVSC) [Slabaugh et al., 2004] aufeinanderfolgend entlang aller drei Koordinatenachsen einen „Sweep“ durch, jeweils in die positive und negative Richtung, ausgehend von den externen Grenzen (vgl. Abb. 9.7(b)). Durch die Sweeps werden die aktiven Kameras ermittelt, in denen das jeweils betrachtete Voxel sichtbar ist. Nur die aktiven Kameras werden dann in den Farbkonsistenztest des Voxels einbezogen. Auch diese Ansätze gehen von dem Ordinal Visibility Constraint aus und verzichten auf einen explizit berechnenden Sichtbarkeitstest. Das **Generalized Voxel Coloring** (GVC) [Culbertson et al., 1999] hingegen ermöglicht eine exakte Berechnung für beliebige Kameraplatzierungen. Die Sichtbarkeit der Voxel wird mithilfe einer **Surface Voxel List** (SVL) verwaltet (vgl. Abb. 9.8). Der **GVC-IB**-Ansatz projiziert in jeder Iteration alle Voxel der SVL in die Kameras und speichert das naheste und damit sichtbare Voxel für jedes Pixel in einem **Item Buffer** (IB) ab. Die Farbkonsistenz jedes Voxels aus der SVL wird anschließend durch die Verwendung der zugeordneten Pixel aus dem Item Buffer bestimmt. Sobald ein Voxel gecarvt und damit aus der SVL entfernt wird, werden seine noch nicht gecarvten Nachbarn zur SVL hinzugefügt. Der Algorithmus terminiert, wenn alle Voxel in der SVL farbkonsistent sind. Ein Nachteil ist, dass die Voxel der SVL permanent in die Kameras projiziert werden müssen. Die Berechnungszeit kann durch die Verwendung von **Sorted Linked Lists** verbessert werden, wie bei dem **GVC-LDI**-Ansatz mit **Layered Depth Images** (LDI) [Culbertson et al., 1999], wodurch allerdings die algorithmische Komplexität und der Speicherverbrauch entsprechend wachsen.

Eine Parallelisierung des Voxel Carvings wird erschwert durch die interativ zu wiederholenden exakten Sichtbarkeitstests. Mit der ursprünglich gegeben Verarbeitungsdauer eines Frames von mehreren Minuten, wie in [Slabaugh et al., 2004] beschrieben, war das Voxel Carving lange Zeit nicht für die Verarbeitung von Bildersequenzen attraktiv. Moderne Grafik-Hardware liefert jedoch neue Möglichkeiten zur Beschleunigung der

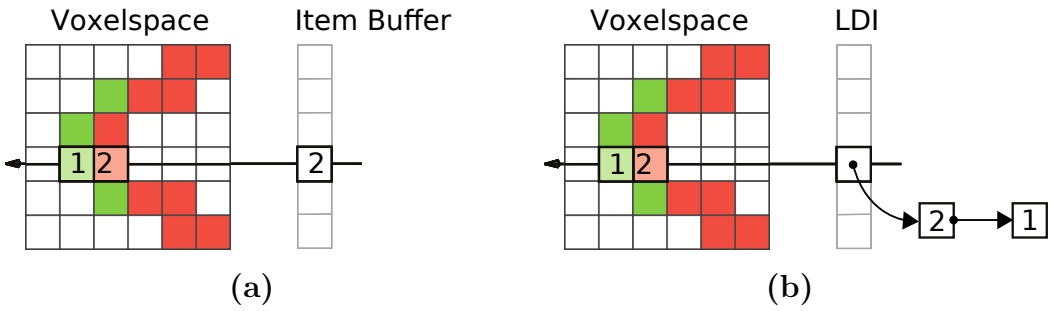


Abb. 9.8: Voxelsichtbarkeit: Illustration der Datenstrukturen für eine Surface Voxel List (SVL), die in den GVC-Algorithmen zum Einsatz kommt. Gezeigt wird ein Item-Buffer in (a) und das Prinzip der Layered Depth Images (LDI) in (b). Abbildung entnommen aus [Ober-Gecks et al., 2016, S. 12], angelehnt an [Culbertson et al., 1999, S. 5].

Berechnung, wie bereits in [Nitschke, 2006] gezeigt wurde. In [Prock und Dyer, 1998] wird die Verwendung des Texture Mappings von Grafikkarten für die schnelle Berechnung von Voxel-Pixel-Projektionen vorgeschlagen. Weiterhin wird empfohlen, einen **Coarse-to-Fine-Ansatz**, beispielsweise durch den Einsatz einer Octree-Datenstruktur, zu verwenden sowie die zeitliche Kohärenz aufeinanderfolgender Bilder bei Bildersequenzen auszunutzen. Viele Verfahren wurden entwickelt, um die Verarbeitung der Photohülle (und auch der Visuellen Hülle) zu beschleunigen, die diese grundlegenden Ideen umsetzen. In [Sainz et al., 2002] wird ein Texture Mapping in Kombination mit einem Octree eingesetzt sowie ein **Multiple Sweep Space Carving** ähnlich zu dem PVSC-Ansatz. In einem anderen Ansatz wird ein Raycasting verwendet, um die Berechnungszeit der Voxel-Pixel-Projektionen zu verbessern [Batchelor et al., 2005]. Dabei kommen 20 bis 30 Kameras zum Einsatz, was eine Berechnungszeit von 700 Sekunden mit der älteren Hardware ergibt. Viele andere Ansätze zeigen Lösungen, welche auf dem Rendering neuer Perspektiven einer Szene fokussieren, z. B. durch das Erfüllen des Ordinal Visibility Constraints, ohne jedoch eine explizite geometrische Rekonstruktion vorzunehmen (siehe [Zwicker, 2013] für Details).

Eine andere Herangehensweise wird in [Nitschke, 2006] propagiert. Eine Grafikkarte wird eingesetzt, um segmentierte Bilder zu erzeugen und anschließend eine VH (via „Vertex Shader“) sowie eine Photohülle (via „Fragment Shader“) zu berechnen. Ein Multi-Sweep-Ansatz, der ein Raycasting mit „Early Ray Termination“ vornimmt, wird verwendet, um die Voxelsichtbarkeiten zu bestimmen, wobei jedoch nur das Voxelzentrum in die Bilder projiziert wird. Für einen Voxelraum mit $94 \times 94 \times 113$ Voxeln und acht FireWire-Kameras mit einer Auflösung von 1024×768 Pixeln kann eine Framerate von 33 fps erzielt werden.

Trotz der verbesserten Laufzeiten genannter Verfahren des Voxel Carvings lassen die Ansätze mindestens einen der folgenden Aspekte außer Acht, der für das betrachtete Überwachungsszenario der Dissertation von Bedeutung ist:

1. Die explizite Berücksichtigung statischer verdeckender Objekte im Überwachungsraum beim Einsatz eines Background-Subtraction-Verfahrens zur Erzeugung einer Hülle, die möglichst alle Objekte im Raum vollständig enthält.

2. Die Projektion gesamter Voxel-Volumina in die Bilder zur Verwendung vollständiger Farbinformationen (aller zugehörigen Projektionspixel), anstatt z. B. nur die Farbinformationen des Projektionspixels vom Voxelzentrum [Nitschke, 2006] oder andere Vereinfachungen zu verwenden [Ladikos et al., 2008].

3. Die Berechnung exakter Voxelsichtbarkeiten für die Farbkonsistenztests, um beliebige Kameraplatzierungen beim Voxel Carving zu ermöglichen.

Das Verfahren der Photohülle in Abschnitt 9.2.3 realisiert diese Anforderungen und bietet als GPU-basierte Implementierung (Abschnitt 9.2.4) die Möglichkeit für ein Überwachungsszenario oder zur schnellen Verarbeitung und Darstellung von hochauflösenden Bildersequenzen eingesetzt zu werden.

9.2.3 Algorithmen der Photohülle

Wie in Abschnitt 9.2.2 beschrieben, erfüllt von den recherchierten Voxel-Carving-Verfahren keines die folgenden Anforderungen der Aufgabenstellung und ist gleichzeitig schnell genug für das Überwachungsszenario: Die Durchführung einer vollständigen Projektion der Voxel-Volumina in die Bilder, die Berechnung exakter Voxel-Sichtbarkeiten, sowie der Umgang mit verdeckenden statischen Objekten bei der unvollständigen Segmentierung durch ein Background Subtraction.

In [Zwicker, 2013] wurde zur Erfüllung dieser Anforderungen ein GPU-basiertes Verfahren entworfen, welches das GVC-IB aus [Culbertson et al., 1999] zugrundelegt. Das GVC-IB wurde ausgewählt, weil darin bereits vollständige Projektionen der Voxelvolumina vorgenommen werden und eine exakte Berechnung der Voxelsichtbarkeiten erfolgt. Im Folgenden wird das GVC-IB im Detail dargelegt. Anschließend werden die notwendigen Anpassungen des Verfahrens für das Überwachungsszenario und die beschleunigten Algorithmen für eine Berechnung auf der GPU präsentiert (vgl. Abschnitt 9.2.4).

Das GVC-IB-Verfahren wird in Algorithmus 9.1 gezeigt und arbeitet wie folgt: Zu Beginn jeder Iteration werden die Oberflächenvoxel bestimmt, die in einer dynamischen

Algorithmus 9.1 GVC-IB-Verfahren [Culbertson et al., 1999] mit Likelihood-Ratio-Test (LRT) als Farbkonsistenzkriterium

```
1:  procedure GVC-IB(V, C, I_col, τ)
2:      V_PH ← V
3:      L ← DETERMINESVL(V_PH)        ▷ determine initial surface visibility list (SVL)
4:      repeat
5:          for all v_i ∈ L do                                  ▷ for all voxels of the SVL
6:              for all c_j ∈ C do                                  ▷ for each camera
7:                  Ψ_{v_i,c_j} ← VISIBILITY(v_i,c_j)          ▷ gather projection pixels
8:                                                                    that view the voxel
9:              end for
10:             Ψ_{v_i} = ∪_{c_j∈C}(Ψ_{v_i,c_j}).
11:             μ_{v_i} ← COMPUTEMEANCOLOR(Ψ_{v_i}, I_col)
12:             val ← COMPUTELRT(μ_{v_i}, Ψ_{v_i})          ▷ compute value for color
13:                                                                 consistency test
14:             if val < τ then                                     ▷ color is consistent
15:                 COLOR(v_i) ← μ_{v_i}      ▷ maintain voxel and add pixel color to voxel
16:             else                                            ▷ color is not consistent
17:                 V_PH ← V_PH \ {v_i}                             ▷ carve voxel
18:                 L ← UPDATESVL(L)                  ▷ remove voxel from SVL and
19:                                                                 add its neighbours
20:             end if
21:         end for
22:     until no voxel is carved
23:     return V_PH
24: end procedure
```

Surface Visibility List (SVL) gespeichert werden, wie in Abb. 9.8(a) gezeigt. Anfangs wird die SVL mit den äußeren Voxeln des Voxelraums initialisiert (Zeilen 2 und 3). Die SVL wird am Ende jeder Iteration aktualisiert (Zeile 18). Für alle Voxel einer aktualisierten SVL wird ein Sichtbarkeitstest durchgeführt (Zeile 7), bei welchem die Voxel der SVL in die Kameras projiziert werden. Dabei wird für jedes Projektionspixel in einem Item-Buffer das Voxel gespeichert, auf welches das Pixel freie Sicht hat. Dies ist für ein Voxel im Vergleich zu anderen Voxeln der SVL dann der Fall, wenn es sich auf dem Sichtstrahl des Pixels befindet und zugleich den kürzesten Abstand zu diesem aufweist (vgl. Abschnitt 3.2.1). In diesem Zusammenhang sei die Funktion $\text{visibility}_{c_j} : V \rightarrow 2^{|P_{c_j}|}$ definiert, gegeben die Pixelmenge P_{c_j}. Dabei liefert $\text{visibility}_{c_j}(v_i)$ die Teilmenge Ψ_{v_i,c_j} aller Projektionspixel Φ_{v_i,c_j} für ein Voxel v_i in der Kamera c_j zurück (mit $\Psi_{v_i,c_j} \subset \Phi_{v_i,c_j}$), welche einen Eintrag in dem Item Buffer aufweisen und damit freie Sicht auf v_i haben. Die Menge an Pixeln aller Kameras, die eine freie Sicht auf ein Voxel v_i haben, ergibt sich somit zu $\Psi_{v_i} = \cup_{c_j \in C}(\Psi_{v_i,c_j})$, wobei gilt: $\Psi_{v_i} \subset \Phi_{v_i}$. Nach Durchführung des Sichtbarkeitstests erfolgt für jedes Oberflächenvoxel v_i aus der SVL ein Farbkonsistenztest. Hierfür werden nun die ermittelten Pixel Ψ_{v_i} herangezogen, da nur

diese auch die Farbe eines Voxels v_i erfassen können. Mithilfe des Konsistenztests wird überprüft, ob das Voxel zur Oberfläche des Objekts gehört. Es wird davon ausgegangen, dass dies der Fall ist, wenn alle Pixel, in denen das Voxel sichtbar ist, die gleiche Farbe aufweisen. Das Voxel bleibt dann in der SVL gespeichert (Zeile 15). Im anderen Fall wird das Voxel entfernt (Zeile 17) und in der SVL durch die nächsten benachbarten potentiellen Oberflächenvoxel ersetzt (Zeile 18).

Es gibt verschiedene Farbkonsistenzkriterien, welche eine Übereinstimmung der Pixelfarben berechnen, um die Farbkonsistenz eines Voxels zu bestimmen. Der **Likelihood Ratio Test** (LRT) aus [Seitz und Dyer, 1999] sei durch eine Funktion $\mathrm{lrt} : \mathbb{R} \to \{0, 1\}$ definiert, so wie in Formel (9.1) angegeben.

$$\mathrm{lrt}_{v_i}\left(\Psi_{v_i}\right) := \begin{cases} 1 & \text{if} \quad (m-1) \cdot \sigma^2_{v_i}(\Psi_{v_i}) < \tau \\ 0 & \text{otherwise} \end{cases} \tag{9.1}$$

Der LRT verwendet die Varianz der Farben $\sigma^2_{v_i} \in \mathbb{R}^3$, die zu den sichtbaren Pixeln Ψ_{v_i} des Voxels v_i gehören sowie die Kardinalität $m = |\Psi_{v_i}|$ dieser Pixel. Die Entscheidung, ob ein Voxel konsistent ($\mathrm{lrt}_{v_i}(\Psi_{v_i}) = 1$) oder inkonsistent ($\mathrm{lrt}_{v_i}(\Psi_{v_i}) = 0$) ist, wird anhand eines Schwellenwerts τ getroffen. Dieser wird experimentell ermittelt und bleibt für die gesamte Rekonstruktion konstant. Die Kolorierung der konsistenten Voxel (die nicht entfernt werden) erfolgt mit dem Mittelwert $\mu_{v_i} \in \mathbb{R}^3$ aus den Farben der sichtbaren Pixel. Der Mittelwert wird auch für die Berechnung der Standardabweichung benötigt.

Der LRT ist durch den Term $(m-1)$ monoton (vgl. Abschnitt 9.2). Nachteilig am LRT ist, dass Voxel, die für viele Pixel sichtbar sind, mit größerer Wahrscheinlichkeit entfernt werden als Voxel, die für wenige Pixel sichtbar sind. Dennoch kann der LRT gute Ergebnisse liefern. In einem experimentellen Benchmark-Test mit dem Tempel und Stegosaurus aus dem „Middlebury-Datensatz“ [Scharstein, 2006] wurde der LRT mit anderen Kriterien, wie dem „Adaptive Standard Deviation Test“ (ASDT), dem „Histogram Test“ und dem „Maximum Distance Test“ verglichen (siehe [Zwicker, 2013] und [Slabaugh et al., 2004]). Mit dem LRT konnten die besten qualitativen Ergebnisse bei gleichzeitig guter Rechenzeit erzielt werden.

Photohülle mit verdeckenden statischen Objekten

Das GVC-IB-Verfahren wird typischerweise für statische Szenen eingesetzt, wobei alle Objekte in der Rekonstruktion enthalten sein sollten. Dabei wird implizit vorausgesetzt, dass entweder eine Segmentierung der Objekte in den Bildern gegeben ist oder aber die Hintergründe aus Sicht aller Kameras so verschieden sind, dass bei den äußeren Voxeln auch tatsächlich Farbinkonsistenzen entstehen, die zur Entfernung dieser Voxel führen.

Andernfalls terminiert der Algorithmus zu früh und die zu rekonstruierenden Objekte im Inneren des Voxelraums werden nicht freigelegt.

In [Zwicker, 2013] wurde eine Segmentierung vorgenommen, um eine erfolgreiche Rekonstruktion zu ermöglichen. Gegeben die Silhouettenbilder der Objekte, können so alle Voxel, deren Projektion nicht innerhalb der Silhouetten liegen, verworfen werden. Die verbleibenden Voxel, die zur Rekonstruktion der Photohülle verwendet werden, repräsentieren dabei nichts anderes als eine Visuelle Hülle, die als Eingabe zur Rekonstruktion einer verfeinerten Photohülle dient und eine Obermenge davon darstellt.

Für das gegebene Überwachungsszenario mit dynamischen Objekten, die in Echtzeit rekonstruiert werden sollen, kommt allerdings nur eine automatische Segmentierung in Betracht, was mit einem Background Subtraction umgesetzt wird. Es wurde bereits die Problematik erläutert, dass damit generierte Silhouetten aufgrund von Verdeckungen durch statische Objekte unvollständig sein können. Möchte man vermeiden, dass Teile der Objekte in der Rekonstruktion fehlen, so kann für die Initialisierung der Photohülle eine konservative VH mit Verdeckungsbehandlung verwendet werden (vgl. Formel 5.5). Zeilen 1 und 2 in Algorithmus 9.1 müssen entsprechend angepasst werden, so dass initial nicht der gesamte Voxelraum sondern nur die VH verwendet wird. In dem nächsten Abschnitt 9.2.4 wird dies entsprechend so dargestellt.

9.2.4 Beschleunigte Algorithmen

Das im letzten Abschnitt 9.2.3 vorgestellte Verfahren des Generalized Voxel Coloring with Item Buffer (GVC-IB) aus [Culbertson et al., 1999] kann prinzipiell zur Verarbeitung von Bildersequenzen eingesetzt werden, wenn die Vorverarbeitung, d. h. die Eingabe einer entsprechenden VH, stimmt. In dem Verfahren werden die Voxelsichtbarkeiten möglichst exakt berechnet und vollständige Voxel-Projektionen in die Bilder für eine hohe Genauigkeit durchgeführt. Allerdings erfüllt des Verfahren nicht den Anspruch einer Online-Fähigkeit, die für ein Überwachungsszenario benötigt wird. Hierzu wurden beschleunigte Algorithmen für die GPU zur Berechnung einer VH und einer Photohülle entworfen (vgl. [Zwicker, 2013] und [Ober-Gecks et al., 2014b]). Eine überarbeitete und erweiterte Darstellung dazu ist in [Ober-Gecks et al., 2016] zu finden und wird im Folgenden wiedergegeben.

Einen wesentlichen Beitrag zur Beschleunigung von Rekonstruktionsalgorithmen kann die Anwendung einer GPU-unterstützten Rendering-Methode (z. B. Raycasting) leisten, um effizient die Voxel-Projektionen in den Kameras zu berechnen und bei der Rekonstruktion für jedes Voxel die Pixel zu bestimmen, für die es sichtbar ist. Zuvor wurden vollständige Voxel-Pixel-Korrespondenzen beispielsweise in einer Lookup-Tabelle ge-

speichert [Kuhn und Henrich, 2009] und ähnlich zu [Ladikos et al., 2008] enkodiert. Zur Umsetzung von Sichtbarkeitstests kam eine SVL mit Item-Buffer zum Einsatz, die nun entfallen kann. Stattdessen wird das Rendering in Kombination mit einer Transferfunktion umgesetzt, wodurch in den Pixeln der gerenderten Bilder jeweils der Index des nahesten belegten Voxels kodiert wird. Das Rendering ermöglicht es, auch höhere Auflösungen des Voxelraums und der Bilder zu verarbeiten.

Zur Umsetzung des Renderings wurde der OpenGL-Standard des Industriekonsortiums Khronos ausgewählt. Es wurde Wert auf einen offenen Standard gelegt und eine gründliche Abwägung zwischen OpenGL und seiner Alternative OpenCL getroffen. Während sich OpenCL für die Berechnung allgemeiner wissenschaftlicher Aufgaben eignet und die Möglichkeit bietet, eine höhere Genauigkeit zu erzielen, ist der Einsatz von OpenGL primär für Aufgaben mit Nähe zur Grafikerzeugung gedacht. Da letzteres auch für die Rekonstruktion einer Photohülle zutrifft, bei der viele Voxelprojektionen in Kamerabilder vorgenommen werden und OpenGL zudem für eine sehr schnelle Verarbeitung optimiert wurde, fiel die Wahl auf OpenGL. Weitere Details hierzu sind [Zwicker, 2013] zu entnehmen. Ob die Genauigkeit und Geschwindigkeit der Berechnungen für die gegebenen Ziele ausreicht, wird in Abschnitt 9.3 beschrieben. Verschiedene GPU-unterstützte Rendering-Methoden des Texture Mappings und Raycastings wurden untersucht und miteinander verglichen.

Die Übertragung des GVC-IB auf die GPU wurde neben den beschleunigten Sichtbarkeitstests durch die folgenden Konzepte ermöglicht: Eine inkrementelle Berechnung der Standardabweichung für das Konsistenzkriterium Likelihood Ratio Test (LRT) [Seitz und Dyer, 1999] sowie die Integration des **Anytime-Konzepts** [Dean und Boddy, 1988], um einer Terminierung des Algorithmus nach einer definierten maximalen Berechnungsdauer T zu erreichen. Die Umsetzung der Abbruchbedingung für die Schleife innerhalb des GVC-IB erfordert bei der verwendeten Version OpenGL 4.3 eine Rückmeldung der Grafikkarte, die schwierig umzusetzen war.

Ein Pseudocode für den Überblick über das gesamte beschleunigte Rekonstruktionsverfahren wird in Algorithmus 9.2 gegeben. Zunächst werden in einem Offline-Schritt für alle Kameras Referenzbilder $I_{\text{col}}^{\text{ref}}$ aufgezeichnet, die für das Background Subtraction benötigt werden (Zeile 2). Anschließend werden diese Bilder zusammen mit weiteren Datenstrukturen auf die GPU transferiert (Zeile 3): der Voxelraum V, zwei Kopien des Voxelraums V_{occupied} und V_{free} (die für die VH benötigt werden), die Kameras C, die statischen Objekte O, ein Schwellenwert τ für die Farbkonsistenztests, sowie eine zeitliche obere Verarbeitungsgrenze T zur Berechnung der Photohülle. Ebenfalls im Offline-Modus werden die Tiefenbilder I_{dep} von den statischen modellierten Objekten

Algorithmus 9.2 Main-Funktion für die beschleunigte Rekonstruktion einer Photohülle zur Online-Verarbeitung von Bildersequenzen

```
1:  procedure MainFunction(V, V_occupied, V_free, C, O, τ, T)
2:      I_col^ref ← captureImages(C)            ▷ capture set of reference images on host
3:      copyToGPU(V, V_occupied, V_free, C, O, τ, T)
4:      I_dep ← createDepthImagesFromObstacles(C, O)      ▷ project obstacles
5:                                                                  into cameras
6:      while SystemStop = false do                          ▷ do for every time frame
7:          I_col ← captureImages(C)          ▷ capture current camera images on host
8:          I_bin ← BackgroundSubtraction(I_col, I_col^ref)          ▷ compute silhouette
9:                                                                  images on host
10:         copyToGPU(I_col, I_bin)
11:         V_VH ← visualHullConservative(V, V_occupied, V_free, C, I_bin, I_dep) ▷ visual
12:                                                                 hull
13:         V_PH ← parallelGVC-IB(V_VH, C, I_dep, τ, T,)          ▷ refine to photo hull
14:         copyToHost(V_PH)                                 ▷ pull back from GPU to host
15:                          ▷ use reconstructed photo hull in further processing steps
16:     end while
17: end procedure
```

auf der GPU gerendert (Zeile 4). Diese werden zur expliziten Berücksichtigung statischer Verdeckungsvolumina bei der Rekonstruktion benötigt.

Zu Beginn jeder Iteration werden Farbbilder I_{col} synchron von allen Kameras aufgezeichnet (Zeile 7). Diese werden zusammen mit den Referenzbildern $I_{\text{col}}^{\text{ref}}$ von dem Background-Subtraction-Verfahren verarbeitet (Zeile 8). Die resultierenden Binärbilder I_{bin} werden mit den Farbbildern I_{col} auf die GPU transferiert (Zeile 10). Die Silhouettenbilder werden anschließend verwendet, um zusammen mit den Tiefenbildern eine konservative VH zu berechnen, die Verdeckungen berücksichtigt (Zeile 11). Hierfür wurde die spezielle VH aus Abschnitt 5.4.3 ebenfalls für den Einsatz auf der GPU neu entworfen (vgl. [Zwicker, 2013]). Im nächsten Abschnitt 9.2.4 wird der Algorithmus dazu vorgestellt. Dieser benötigt die zwei temporären Kopien des Voxelraums V_{occupied} und V_{free}. Die Vorberechnung der VH für die anschließende Rekonstruktion einer Photohülle wird durchgeführt, da die gleichfarbigen Hintergründe der Roboterarbeitszelle (vgl. Abb. 1.2) andernfalls durch das Auftreten unerwünschter Farbkonsistenzen zu einer verfrühten Terminierung des Algorithmus führen würden. Anschließend wird die Photohülle nach dem beschleunigten GVC-IB Verfahren berechnet (Zeile 13). Warum die Tiefenbilder auch für das Voxel Carving verwendet werden, wird an späterer Stelle erklärt. Zum Schluss wird das Rekonstruktionsergebnis auf die CPU transferiert (Zeile 14).

In den nächsten beiden Abschnitten werden nun die beschleunigten Rekonstruktionsalgorithmen vorgestellt.

Beschleunigte konservative Visuelle Hülle mit verdeckenden statischen Objekten

Eine konservative VH, die Verdeckungen von statischen Objekten behandelt, wurde in Abschnitt 5.4.3 präsentiert. In diesem Abschnitt wird eine Adaption dieses Algorithmus zur Berechnung auf der Grafikkarte vorgestellt, zu finden in den Algorithmen 9.3 und 9.4.

Als Eingabe für den Algorithmus 9.3 dienen neben dem Voxelraum V zwei gleich große Voxelräume V_{free} und V_{occupied} (welche die Eigenschaften des originalen Voxelraums aufweisen), sowie die Kameras C, die Silhouettenbilder I_{bin} und Tiefenbilder von den statischen Objekten I_{dep}. Jeder Kamera wird eine ID zugewiesen, gegeben durch eine Funktion $\text{id} : C \to \{1, \ldots, |C|\}$ mit $\text{id}(c_j) = j$. Die Verarbeitung erfolgt sequentiell für eine Kamera nach der anderen, sortiert nach aufsteigender $\text{id}(c_j)$, aber parallel für alle Pixel einer Kamera. Für jedes Pixel werden in einer sequentiellen Abarbeitung sämtliche Voxel behandelt, die sich in dem Kegel seiner Rückprojektion befinden (Zeile 10). Dies wird ermöglicht durch ein iteratives Rendering. Begonnen wird mit dem nahesten Voxel (Algorithmus 9.3, Zeile 5). Der Prozess, der für jedes Pixel und jedes Voxel ausgeführt wird (Algorithmus 9.3, Zeile 9), wird in Algorithmus 9.4 dargestellt.

Jedes Voxel wird initialisiert mit 0 und soll eine einzelne Kamera-ID nach der Verarbeitung enthalten. Diese ID erhält man durch die Funktion $\text{value}_z : V \to \{0\} \cup \{1, \ldots, |C|\}$ über die Iterationsschritte z, mit dem Wert $\text{value}_0(v_i) = 0$. Der Wert der Iteration $z + 1$ leitet sich wie in Algorithmus 9.4 gezeigt, ab. Im Wesentlichen erhält jedes Voxel

Algorithmus 9.3 Verfahren für eine beschleunigte konservative VH mit Verdeckungsbehandlung

```
 1: procedure VISUALHULLCONSERVATIVE(V, V_occupied, V_free, C, I_bin, I_dep)
 2:     for from k ← ID(c)= 1 to ID(c)= |C| step 1 do  ▷ sequentially with ascending
 3:                                                                  camera id
 4:         parallel for all p_l ∈ P_{c_j} do                        ▷ parallel for all pixels
 5:             v ← GETFIRSTVOXELPROJECTINGTOPIXEL(p_l, V)           ▷ render closest
 6:                                                                  voxel
 7:             while v ∈ V do
 8:                 v_occupied ← GETVOXELOCC(v), v_free ← GETVOXELFREE(v)
 9:                 PROCESSVOXELINPIXEL(v_occupied, v_free, p_l, k, I_bin, I_dep)
10:                 v ← GETNEXTVOXELPROJECTINGTOPIXEL(v, p_l, V)
11:             end while
12:         end parallel for
13:     end for
14:     V_VH ← ANALYZEVOXELVALUES(V_occupied,V_free)    ▷ gather all occupied voxels
15:     return V_VH
16: end procedure
```

Algorithmus 9.4 Verfahren zur parallelen Verarbeitung der Pixeldaten

```
1: procedure processVoxelInPixel(v_occupied, v_free, p_l, c_j, I_bin, I_dep)
2:     if value(v_occupied) ≥ value(v_free) ∨ value(v_free) = id(c_j) then
3:         if (getImageValue(p_l, I_bin, c_j) = 1 ∨                ▷ if pixel is foreground
4:             |r_{v_i} − r_{c_j}| + d_v ≥ getImageValue(p_l, I_dep, c_j) ) then    ▷ if voxel is
5:                                                                                  occluded
6:             value(v_occupied) ← id(c_j)
7:         else                                  ▷ if voxel is visible and pixel is background
8:             value(v_free) ← id(c_j)
9:         end if
10:    end if
11: end procedure
```

für jedes Projektionspixel einen Eintrag von mindestens einer Kamera-ID entweder im Voxelraum V_{free} oder im Voxelraum V_{occupied}. Der Pixel-Parallelismus wird sichergestellt durch die spezielle Konstruktion der Bedingung in Zeile 2 in Algorithmus 9.4. Mögliche Mischungen von Schreib- und Leseoperationen auf dem Voxelraum spielen keine Rolle, da nach der Verarbeitung einer Kamera (äußere Schleife), eine „Memory Barrier“ angewendet wird, welche die Synchronisierung gewährleistet.

Alle Projektionspixel eines Voxels, die Teil einer Objektsilhouette sind oder die aufgrund von Verdeckungen nicht das Voxel „sehen“ können, erzeugen einen Eintrag in V_{occupied} (Algorithmus 9.4, Zeile 6). Alle anderen Projektionspixel generieren einen Eintrag in V_{free} (Zeile 8). Um ein Voxel entfernen zu können und somit als transparent zu markieren, genügt es, wenn das Voxel in einer Kamera vollständig als Hintergrund klassifiziert wird und dabei für keins seiner Projektionspixel in dieser Kamera verdeckt ist. Die Bedingung der Zeile 2 in Algorithmus 9.4 ist so konstruiert, dass sie für die nachfolgend bearbeiteten Kameras niemals wahr werden kann, wenn einmal eine Kamera gefunden wurde, für die das Voxel vollständig als Hintergrund klassifiziert wurde. Die finale VH entspricht allen Voxeln, die als belegt (occupied) klassifiziert wurden. Die Klassifikation jedes Voxels nach der Verarbeitung kann ausgelesen werden, indem seine Einträge in V_{occupied} und V_{free} in einem Nachbearbeitungsschritt miteinander verglichen werden, wie in Tabelle 9.9 gezeigt (vgl. Algorithmus 9.3, Zeile 14).

Bei den beschriebenen Algorithmen 9.3 und 9.4 sind alle Objekte, also auch die verdeckenden statischen Objekte, Teil der Rekonstruktion. Damit bleiben Teile der Rekonstruktion über die Bildersequenz hinweg gleich. Diese konstanten Teile werden bei der nachfolgenden Berechnung einer Photohülle ignoriert, um eine bessere Laufzeit zu erzielen. Die konstanten Teile können einmalig in einem Offline-Schritt berechnet werden und nachträglich zu den Online-Rekonstruktionsergebnissen einfach hinzugefügt werden (nicht in Algorithmus 9.2 dargestellt).

Bei der Rekonstruktion wird sich im Folgenden auf die veränderlichen Teile der Umgebung, d. h. die sich bewegenden Objekte, konzentriert. Aus diesem Grund wird der Algorithmus 9.4 zur Verarbeitung der Pixel so abgewandelt wie in Algorithmus 9.5 angegeben. In Zeile 9 des Algorithmus 9.3 muss entsprechend die Methode PROCESSVOXELINPIXELIGNORINGOCCLUSIONS aus Algorithmus 9.5 aufgerufen werden. Der Unterschied besteht darin, dass verdeckte Pixel nun vollständig ignoriert werden (Zeilen 2 und 3). Die resultierenden Werte der zwei Voxelräume V_{free} und V_{occupied} müssen resultierend anders interpretiert werden, so wie in Tabelle 9.10 vermerkt.

Zusammengefasst wird die Rekonstruktion in einen statischen Teil (der offline generiert wird) und einen sich veränderlichen Teil (der online rekonstruiert wird) aufgeteilt. Bei dieser Modifikation muss beachtetet werden, dass nur beide Teile zusammengenommen die Garantie erbringen, dass alle Objekte vollständig in der Rekonstruktion enthalten sind. Während bei dem statischen Teil der Rekonstruktion davon ausgegangen werden kann, dass darin auch alle statischen Objekte enthalten sind, so gilt dies für den dynamischen Teil der Rekonstruktion von den sich bewegenden Objekten nicht. Dies liegt daran, dass der statische Teil der Rekonstruktion auch verdeckte leere Bereiche enthalten kann, die sensorisch nicht einsehbar sind (Sichtbarkeitsgrad von 0, vgl. Abschnitt 3.2.1). Darin können sich temporär auch Objekte aufhalten, die nicht zu den statischen Objekten gehören. Dieser Aspekt muss bei der Interpretation der Rekonstruktionsergebnisse mit den gegebenen Algorithmen entsprechend berücksichtigt werden.

Zur Offline-Generierung einer Rekonstruktion der statischen Objekte können die Tiefenbilder miteinander verschnitten werden, was eine Tiefenhülle ergibt und damit die bestmögliche Approximation der Geometrie der statischen Objekte für die gegebenen Kameraperspektiven. Verwendet man das Ergebnis anschließend zusammen mit den Referenzbildern als Eingabe für das GVC-IB-Verfahren (Algorithmus 9.1), so erhält man eine Kolorierung dieser Rekonstruktion. Alternativ könnten auch Occlusion Masks von den statischen Objekten generiert werden (vgl. Abschnitt 5.3.3), die als Eingabe zur Berechnung einer VH (ohne Verdeckungsbehandlung) dienen. Diese VH kann dann wiederum durch den GVC-IB verbessert und koloriert werden. Die zweite Variante ist allerdings ungenauer als die Verwendung der Tiefeninformationen, die sowieso zur Verfügung stehen.

Beschleunigte Photohülle

Als Sichtbarkeitstest sei wieder die Funktion $\text{visibility}_{c_j} : V \to 2^{|P_{c_j}|}$ definiert, gegeben die Pixelmenge P_{c_j} einer Kamera c_j. Für ein Voxel v_i liefert $\text{visibility}_{c_j}(v_i)$ die Teilmenge $\Psi_{v_i,c_j} \subset \Phi_{v_i,c_j}$ seiner Projektionspixel zurück, die freie Sicht auf dieses Voxel haben und somit auch seine Farbe erfassen können.

$\text{value}(v_{\text{occupied}}) < \text{value}(v_{\text{free}})$
Die Kamera mit der $\text{id}(c_j) = \lvert C \rvert$ oder eine zuvor ausgewertete Kamera enthält nur Projektionspixel, die als Hintergrund klassifiziert wurden. Es genügt, wenn eine Kamera ein Voxel vollständig als Hintergrund sieht, um diesen zu entfernen. Damit muss das Voxel leer (free) sein und kann entfernt werden.
$\text{value}(v_{\text{occupied}}) \geq \text{value}(v_{\text{free}})$
Für die zuletzt ausgewertete Kamera $\text{id}(c_j) = \lvert C \rvert$ wurde mindestens ein verdeckter oder ein Vordergrund-Projektionspixel vermerkt. Dies bedeutet, dass von den zuvor ausgewerteten Kameras keine gefunden wurde, bei der die Projektionspixel vollständig als Hintergrund klassifiziert wurden. Das Voxel ist damit belegt (occupied) und Teil der Rekonstruktion.

Abb. 9.9: Interpretation der finalen Werte in den Datenstrukturen V_{free} und V_{occupied} für jedes Voxel zur Bestimmung der belegten Voxel für die Rekonstruktion.

Die SVL wird beim GVC-IB verwendet, um die Anzahl an Voxelprojektionen zu verringern. Allerdings ist die Implementierung solch einer dynamischen Liste auf der GPU nur schwer umzusetzen. Zudem können Elemente damit nur auf eine sequentielle Art und Weise eingefügt oder gelöscht werden. Deshalb wird die SVL ersetzt durch eine effiziente GPU-Rendering-Technik (vgl. Abschnitt 9.3). Durch das Rendering ist es möglich, in jeder Iteration all die aktuell belegten Voxel V_{PH} des Voxelraums V in die Kameras zu projizieren. Hierfür wird auf jedes Voxel eine Visibility-Transfer-Funktion angewendet, wie in Formel (9.2) dargestellt, ähnlich einer Volumen-Rendering-Methode, die in anderen Anwendungsgebieten zum Einsatz kommt. Diese wird nun näher erläutert.

Es sei $\text{idx} : V \to \mathbb{N}^3$ eine Funktion, die für die Gridposition eines Voxels v_i ein Tripel von Indizes $(\text{idx}(v_i)^{(1)}, \text{idx}(v_i)^{(2)}, \text{idx}(v_i)^{(3)})$ zurückgibt. Weiterhin sei durch die Funktion $\text{image}_{\text{vis},c_j} : P_{c_j} \to \mathbb{N}^3 \times \{0, 1\}$ die Abbildung der Pixel auf virtuelle Pixeldaten für jede Kamera c_j gegeben. Die Bilder aller Kameras des Multi-View-Kamerasystems lassen sich

Algorithmus 9.5 Verfahren zur parallelen Verarbeitung der Pixeldaten. Verdeckungen werden ignoriert und statische Objekte nicht rekonstruiert

```
1: procedure PROCESSVOXELINPIXELIGNORINGOCCLUSIONS(v_i, p_l, c_j, I_bin, I_dep)
2:     if |r_{v_i} − r_{c_j}| + d_v ≥ GETIMAGEVALUE(p_l, I_dep, c_j) then
3:         return
4:     end if
5:     if VALUE(v_occupied) ≥ VALUE(v_free) ∨ VALUE(v_free) = ID(c_j) then
6:         if GETIMAGEVALUE(p_l, I_bin) = 1 then
7:             VALUE(v_occupied) ← ID(c_j)
8:         else
9:             VALUE(v_free) ← ID(c_j)
10:        end if
11:    end if
12: end procedure
```

$\text{value}(v_{\text{occupied}}) < \text{value}(v_{\text{free}})$
Die Kamera mit der $\text{id}(c_j) = \lvert C\rvert$ oder eine zuvor ausgewertete Kamera enthält nur Projektionspixel, die als Hintergrund klassifiziert wurden sowie möglicherweise verdeckte Pixel. Eine Kamera, die dies erfüllt, genügt, um das Voxel zu entfernen. Das Voxel wird als leer (free) markiert.
$(\text{value}(v_{\text{occupied}}) \geq \text{value}(v_{\text{free}})) \wedge (\text{value}(v_{\text{occupied}}) > 0)$
Mindestens ein Vordergrund-Projektionspixel wurde für die zuletzt ausgewertete Kamera $\text{id}(c_j) = \lvert C\rvert$ vermerkt. Damit sieht keine vorhergehende Kamera das Voxel als leer. Das Voxel wird als belegt (occupied) markiert.
$(\text{value}(v_{\text{occupied}}) = 0) \wedge (\text{value}(v_{\text{free}}) = 0)$
Das Voxel ist in allen Projektionspixeln aller Kameras verdeckt. Verdeckte Voxel werden nicht rekonstruiert. Deshalb ist das Voxel leer (free).

Abb. 9.10: Interpretation der finalen Werte in den Datenstrukturen V_{free} und V_{occupied} für jedes Voxel zur Bestimmung der belegten Voxel für die Rekonstruktion. Statische Objekte sind nicht Teil der Rekonstruktion

damit als Visibility Images (VI) $I_{\text{vis}} = \{\text{image}_{\text{vis},c_1}, \ldots, \text{image}_{\text{vis},c_{|C|}}\}$ zusammenfassen. Der Wert jedes Pixels in einem Visibility Image wird bestimmt durch eine Funktion $\text{transfer}_{p_l} : V \rightarrow \mathbb{N}^3 \times \{0,1\}$, welche einen Voxelindex auf die Farbkanäle des Bilds abbildet und einen Belegungswert von $\{0,1\}$ hinzufügt (vgl. Formel (9.2)). Dieser gibt an, ob ein Voxel belegt (= 1) oder frei ist (= 0). Letzterer Wert wird in dem Alpha-Kanal des Bilds gespeichert.

$$\text{transfer}_{p_l}(v_i) = \begin{cases} (\text{idx}(v_i)^{(1)}, \text{idx}(v_i)^{(2)}, \text{idx}(v_i)^{(3)}, 1) & \text{if} \quad v_i \in V_{\text{PH}} \quad \wedge \quad p_l \in \Phi_{v_i,c_j} \quad \wedge \\ & \lvert r_{v_i} - r_{c_j}\rvert + d_v < \text{image}_{\text{dep},c_j}(p_l) \\ (0,0,0,0) & \text{otherwise} \end{cases} \tag{9.2}$$

In Formel (9.2) werden auch Verdeckungen berücksichtigt: Nur Voxel, die sich vor den Oberflächen der statischen Objekte befinden, werden gerendert (Bedingung: $\lvert r_{v_i} - r_{c_j}\rvert + d_v < \text{image}_{\text{dep},c_j}(p_l)$). Diese Einschränkung ist notwendig, da als Eingabe für das Voxel Carving nur der Teil der VH verwendet wird, der den dynamischen Teil der Szene repräsentiert, um die Laufzeit zu verbessern. Möchte man hingegen alle Objekte rekonstruieren (vollständige VH als Eingabe), die statische und sich bewegende Objekte enthält, so darf diese Bedingung nicht gelten. Aufgrund der Abarbeitung des Voxelraums von außen nach innen treten bei der Eingabe einer vollständigen VH keine zusätzlichen Verdeckungen auf, die berücksichtigt werden müssten. Es wird die gesamte Szene rekonstruiert.

Entscheidend für die Anwendung der Transferfunktion ist, dass das Rendering stets für jedes Pixel beendet wird, nachdem das naheste sichtbare belegte Voxel gezeichnet wurde.

Algorithmus 9.6 GVC-IB-Verfahren für die GPU

```
1:  procedure parallelGVC-IB(V_VH, C, I_col, I_dep, τ, T, P_{c_j})
2:      V_PH ← V_VH
3:      for all v_k ∈ V_PH do
4:          occupation(v_k, true)                ▷ initialize all input voxels as occupied
5:      end for
6:      s ← time                                                    ▷ get current time
7:      while Time − s < T do  ▷ reconstruct for defined duration (anytime concept)
8:          I_vis ← renderVisibilityImages(V_PH, I_dep)             ▷ render closest
9:                                                                  occupied voxels
10:         for all c_j ∈ C do                                      ▷ for all cameras
11:             parallel for all p_l ∈ P_{c_j} do          ▷ for all pixels in visibility image
12:                 v_i ← getImageValue(p_l, I_vis, c_j) ▷ get voxel that is visible in pixel
13:                 μ^n_{v_i} ← updateMeanColor(v_i, I_col, c_j, p_l)   ▷ update color mean
14:                 σ^n_{v_i} ← updateStandardDeviation(v_i, μ^n_{v_i})  ▷ update standard
15:                                                                 deviation
16:                 if ((n − 1)/n) · (n · (σ_{v_i})^2) < τ then     ▷ if color is consistent
17:                     setColor(v_i, μ^n_{v_i})                    ▷ set new voxel color
18:                 else                                            ▷ color is not consistent
19:                     occupation(v_i, false)                      ▷ carve voxel by setting its
20:                                                                 state to not occupied
21:                 end if
22:             end parallel for
23:         end for
24:     end while
25:     V_PH ← AnalyzeVoxelValues(V_PH)                 ▷ gather all occupied voxels
26: end procedure
```

Dies wird mit einem sortierten Rendering, beispielsweise einem Raycasting realisiert, so wie in Algorithmus 9.7 dargestellt. Nach dem Rendering in jeder Iteration kodieren die Farbwerte jedes Pixels p_l eines Visibility Images die Koordinaten des nahesten belegten Voxels, sofern vorhanden (siehe Algorithmus 9.6, Zeile 8). Das entspricht den Einträgen eines Item Buffers nach der Projektion einer SVL in die Kameras.

In Algorithmus 9.6 ist das beschleunigte GVC-IB-Verfahren dargestellt. Wie beschrieben, dient die VH als Eingabe (Zeile 2), durch welche auch die initiale Belegung jedes Voxels festgelegt wird (Zeile 4). Für jedes Voxel, das entfernt wird, wird dieser Eintrag entsprechend geändert (Zeile 19). Damit wird gewährleistet, dass es in der nächsten Iteration nicht wieder in die Visibility Images gerendert wird (Zeile 8). Die Pixel einer jeden Kamera werden parallel abgearbeitet (Zeile 11). Dadurch kann der LRT (Zeile 16), permanent für alle Voxel nach jeder Aktualisierung der Farbinformationen (Zeilen 13 und 14) durch zugehörige Projektionspixel ausgeführt werden. Die Variable n ist im jeweiligen Voxel gespeichert und wird bei einer Aktualisierung inkrementiert. Sie müsste

Algorithmus 9.7 Verfahren zum Rendern der Visibility Images mittels Raycasting

```
1: procedure RENDERVISIBILITYIMAGES(V, V_VH, C, I_dep, I_vis)
2:     for all c_j ∈ C do
3:         parallel for all p_l ∈ P_{c_j} do
4:             color ← 0
5:             V_ray ← GETVOXELSALONGRAY(p_l, V)
6:             while color = 0 do          ▷ stop after rendering the first occupied voxel
7:                 v_i ← GETNEXTVOXEL(V_ray)
8:                 color ← GETVALUEFROMTRANSFERFUNCTION(v_i, p_l, c_j, V_PH, I_dep)
9:             end while
10:            SETIMAGEVALUE(p_l, I_vis, c_j, color)        ▷ pixel contains next possible
11:                                                              surface voxel
12:        end parallel for
13:    end for
14: end procedure
```

daher indiziert werden (n_{v_i}), dies wurde jedoch aus Gründen der Übersichtlichkeit ausgelassen. Für die Aktualisierung der Werte n, $\mu^n_{v_i}$ und $\sigma^n_{v_i}$ eines Voxels v_i durch die Projektionspixel wird eine Synchronisierung benötigt, um eine „Race Condition“ zu vermeiden. Dies wird durch die Verwendung von „Spin Locks“ realisiert. Aufgrund der großen Pixelanzahl kann aber dennoch ein paralleles Abarbeiten ermöglicht werden.

Der Likelihood Ratio Test als Farbkonsistenzkriterium verwendet die Standardabweichung $\sigma_{v_i} \in \mathbb{R}^3$. Nicht gecarvte belegte Voxel werden mit dem mittleren Wert $\mu_{v_i} \in \mathbb{R}^3$ koloriert (Zeile 17). Dieser wird aus den Farben der Projektionspixel Ψ_{v_i} aller Kameras, für welche das Voxel v_i sichtbar ist, berechnet. Die Farbe jedes Pixels $p_l \in \Psi_{v_i}$ sei gegeben als $u_k = \text{image}_{\text{col},c_j}(p_l)$. Der Farbwert μ_{v_i} eines Voxels sowie die assoziierte Standardabweichung σ_{v_i} werden inkrementell berechnet, um dynamische Datenstrukturen zu vermeiden.

Die originale Definition des Mittelwerts benötigt alle $n = |\Psi_{v_i}|$ Elemente u_k im Voraus wie in Formel (9.3) angegeben.

$$\mu_{v_i} = \frac{1}{n} \sum_{k=1}^{n} u_k \tag{9.3}$$

Dies kann ersetzt werden durch die inkrementelle Gleichung in Formel (9.4).

$$\mu^n_{v_i} = \frac{1}{n}(u_n - \mu^{n-1}_{v_i}) + \mu^{n-1}_{v_i} \tag{9.4}$$

Die originale Definition der Standardabweichung kann Formel (9.5) entnommen werden.

$$\sigma^n_{v_i} = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (u_k - \mu_{v_i})^2} \tag{9.5}$$

Sein inkrementelles Äquivalent lässt sich nach [Finch, 2009] durch Formel (9.6) ausdrücken.

$$n \cdot (\sigma_{v_i}^n)^2 = (n-1) \cdot (\sigma_{v_i}^{n-1})^2 + n \cdot (n-1) \cdot (\mu_{v_i}^n - \mu_{v_i}^{n-1})^2 \tag{9.6}$$

Es seien die Funktionen $f : \mathbb{R} \to \mathbb{R}$ und $g : \mathbb{R} \to \mathbb{R}$ definiert. Wie aus der Formel ersichtlich wird, benötigt die Berechnung von $n \cdot (\sigma_{v_i}^n)^2$ die Evaluierung eines monotonen Ausdrucks der Form $f(n) = f(n-1) + X$ mit $X > 0$. Ähnlich dazu ist die Funktion $g(n) = 1 - 1/n = (n-1)/n$ dann monoton, wenn gilt: $n > 0$. Das Produkt dieser beiden Funktionen $g(n) \cdot f(n)$ ist damit ebenso monoton. Eine Kombination der Formel (9.6) mit der Definition für den Konsistenztest $\mathrm{lrt} : \mathbb{R} \to \mathbb{R}$, welcher gegeben ist als $\mathrm{lrt} = (n-1) \cdot (\sigma_{v_i})^2$, ergibt eine Funktion, die der Form $g(n) \cdot f(n)$ folgt. Somit kann lrt inkrementell berechnet werden, unter Verwendung des Terms aus Formel (9.7), der monton in n ansteigt:

$$\mathrm{lrt}(n) = \left(\frac{n-1}{n} \right) \cdot \left(n \cdot (\sigma_{v_i}^n)^2 \right) \tag{9.7}$$

Aufgrund dieser Adaption kann der Farbkonsistenztest parallel für alle Pixel in jeder Iteration ausgeführt werden. Immer wenn gilt $\mathrm{lrt}(n) > \tau$ kann das Voxel sofort entfernt (gecarvt) werden, da der Wert der Funktion $\mathrm{lrt}(n)$ aufgrund der Monotonie nicht mehr kleiner werden kann. Ein wiederholtes Entfernen beeinflusst das Ergebnis nicht.

Neben der inkrementellen Berechnung des LRTs bestand eine weitere Herausforderung bei der Übertragung des GVC-IB auf die GPU darin, die Schleifenbedingung „until no voxel is carved“ (Algorithmus 9.1, Zeile 22) umzusetzen, da diese eine Rückmeldung von der Grafikkarte benötigt. Das konnte zumindest zur Zeit der Realisierung nur durch einen „Trick“ umgesetzt werden [Zwicker, 2013], [Owens et al., 2007]. Zudem variiert die Berechnungszeit der Photohülle abhängig vom Szeneninhalt. Deshalb wird für eine Online-Anwendung eine Grenze für eine maximale Berechnungsdauer benötigt. Dazu kommt das Konzept des Anytime-Algorithmus zum Einsatz [Dean und Boddy, 1988]. Nach einem Initialisierungsschritt können Anytime-Algorithmen zu jedem Zeitpunkt unterbrochen werden und dabei stets ein gültiges Ergebnis liefern. Die Qualität des Ergebnisses verbessert sich als Funktion über die Zeit. Für die Rekonstruktion einer Photohülle wird eine obere Grenze T für die Verarbeitungszeit festgelegt, in welcher mindestens eine Iteration durchgeführt werden kann. Damit entspricht das Rekonstruktionsergebnis im schlechtesten Fall einer kolorierten Visuellen Hülle und im besten Fall einer Photohülle, die eine gute geometrische Approximation der wahren Szene darstellt. Es muss allerdings angemerkt werden, dass die erreichbare Qualität einer Photohülle nicht nur von den berechneten Iterationen abhängt, sondern ebenfalls von den Eigenschaften der zu rekonstruierenden Objekte sowie der Anzahl und gewählten Perspektiven der Kameras.

9.3 Bewertung der Photohülle für den Einsatz im Überwachungsszenario

Im Folgenden werden die wichtigsten Experimente aus [Zwicker, 2013] dargestellt. Weitere Experimente und ausführlichere Ergebnisse zur Berechnungsdauer und Qualität der rekonstruierten Photohüllen sind [Zwicker, 2013] zu entnehmen. Zur Evaluierung des beschriebenen Algorithmus wurden Bildersequenzen der SIMERO-Roboterarbeitszelle (vgl. Abb. 1.2) sowie Aufnahmen einer mit Blender erstellten Simulationsumgebung von [Stoychev, 2013] eingesetzt.

Umgebung

Die Experimente wurden mit einer Grafikkarte des Typs AMD Radeon HD 7970 mit 3 GB RAM durchgeführt. Das Betriebssystem OpenSUSE 12.3 wurde eingesetzt sowie der proprietäre AMD-Treiber „fglrx“ (Catalyst) in der Version 12.104. Der Prozessor war vom Typ AMD FX-8350 Octacore mit 4 GHz und 32 GB RAM. Aufgrund der intensiven Nutzung der Grafikkarte spielt die Kernanzahl jedoch keine Rolle. Die Auflösung der Kamerabilder mit 640×480 Pixeln war bei allen Experimenten gleich.

Analyse der Rendering-Verfahren

Für die präsentierten Algorithmen wurde eine schnelle und effiziente Berechnung der vollständigen Voxel-Pixel-Projektionen benötigt. Diese werden mithilfe einer GPU-basierten Rendering-Methode umgesetzt. Dafür wird der Voxelraum unter Verwendung der kalibrierten Kameraparameter in alle Bildebenen gerendert (inklusive der Verzerrungen). Jedes Pixel kodiert anschließend in seinen RGB-Werten die drei Koordinaten des sichtbaren Voxels.

Für das effiziente Rendering existieren zwei Gruppen von Ansätzen, das Texture Mapping sowie das Raycasting. Beim Texture Mapping wird die Fähigkeit der Grafikkarten zur Texturierung von Oberflächen genutzt [Meissner et al., 1999], [Fernando und NVIDIA Corporation, 2004]. Zu rendernde Volumina, wie in dieser Dissertation der Voxelraum, werden dabei als Stapel von Bildern (Ebenen) interpretiert. In jedes Bild werden die Voxel gerendert, die der jeweiligen Ebene zugeordnet sind. Die Bilder werden anschließend als Textur auf lokal versetzte, parallele Polygone gelegt und die Polygone dann aus Sicht der Ziel-Kamera gerendert (vgl. Abb. 9.11). Durch die Aktivierung von „Blending“ werden Ebenen die weiter weg sind, in leeren oder semitransparenten Pixeln sichtbar. Für den Betrachter entsteht so ein räumlicher Bildeindruck. Die dafür verwendeten Ebenen

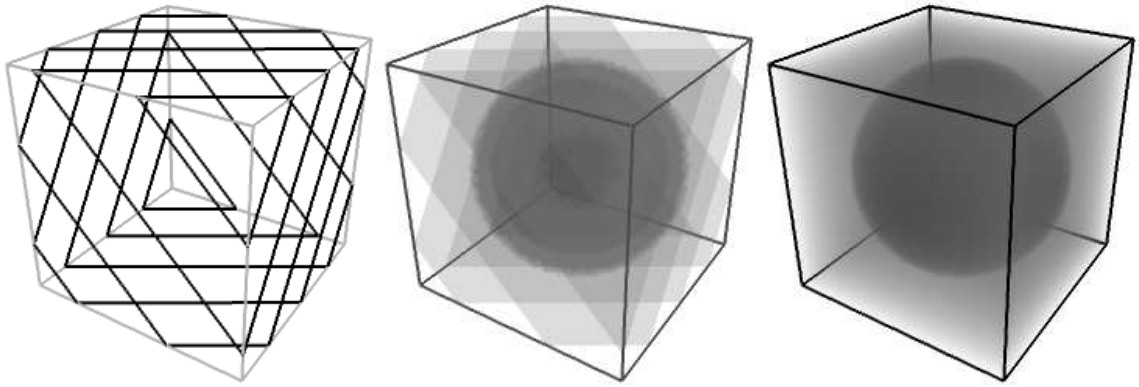


Abb. 9.11: Rendering eines Voxelraums mit Texture Mapping. Die Rendering-Ebene ist parallel zu den dargestellten Ebenen ausgerichtet. Abb. entnommen aus [Meissner et al., 1999].

werden als „Proxy Geometry“ bezeichnet. Die Qualität des Renderings hängt stark vom Abstand zwischen den einzelnen Ebenen sowie der Orientierung aller Ebenen in Bezug auf die Rendering-Ebene ab. Größere Distanzen führen zu Aliasing-Effekten und falschen Voxel-Pixel-Abbildungen. Um Aliasing-Effekte zu vermeiden, werden Ebenen parallel zu jeder Koordinatenachse eingeführt, so wie in Abb. 9.12 dargestellt.

Beim Raycasting wird das sichtbare Voxel eines Pixels folgendermaßen bestimmt: Für jedes Pixel wird der rückprojizierte Sichtstrahl solange verfolgt, bis dieser ein belegtes Voxel im 3D-Raum schneidet. Diese Vorgehensweise wird als „Ray Marching“ bezeichnet. Die Ausmaße eines Sichtkegels werden dabei vernachlässigt. Ein betrachteter Sichtstrahl wird mit gleicher Schrittweite abgetastet und auf Schnitt mit den Volumina getestet. Zur Beschleunigung des Ray Marchings existieren unterschiedliche Optimierungen, wie z. B. das „Empty Space Skipping“ (vgl. [Kruger und Westermann, 2003]). Aliasing-Effekte, die durch das diskrete Ray Marching verursacht werden, lassen sich mit einer Optimierung vermeiden (vgl. [Amanatides und Woo, 1987]): Die Schnittpunkte des Sichtstrahls mit den Voxeln werden dabei analytisch berechnet.

Bei den Experimenten wurde das Texture Mapping mit einem „Standard-Raycasting“, einem „Amanatides Raycasting“, sowie einer „Compute-Shader-Implementierung“ des

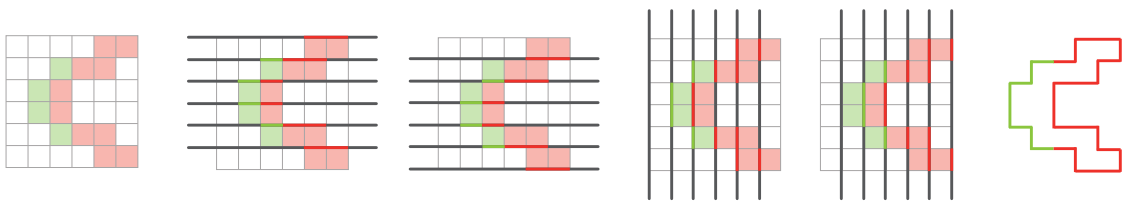


Abb. 9.12: Exaktes Texture Mapping eines Voxelraums. Von links nach rechts: belegte Voxel, gerendert in die Richtungen: $+y$, $-y$, $-x$, $+x$, finales Bild der Voxel. Abbildung entnommen aus [Zwicker, 2013, S. 51].

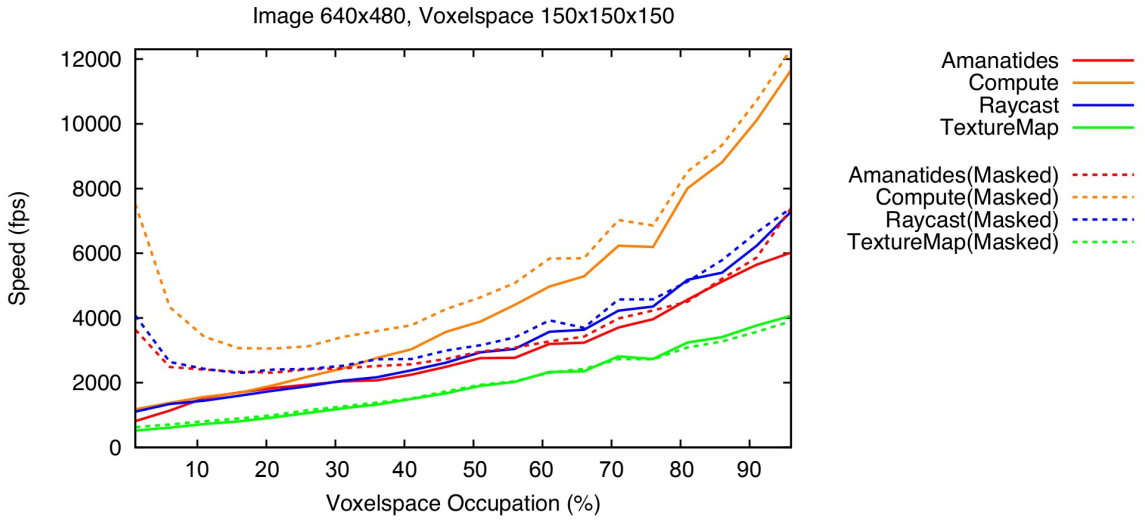


Abb. 9.13: Berechnungsdauer unterschiedlicher Rendering-Methoden bei einer variierenden Voxelraumbelegung. Die gestrichelten Linien „Masked“ zeigen die Ergebnisse für die Verwendung segmentierte Bilder an. Abbildung entnommen aus [Zwicker, 2013, S. 74].

Amanatides-Algorithmus in OpenCL verglichen. Zur Analyse der Berechnungsdauer dieser Rendering-Ansätze wurde in [Zwicker, 2013] ein vom Rekonstruktionsverfahren unabhängiger Test durchgeführt. Dabei wurden unterschiedliche Voxelräume in die Kamerabilder projiziert. Die Voxelraumauflösung variierte von 100^3 bis 350^3 Voxeln, in 50^3-Voxel-Schritten. Die Belegung des Voxelraums wurde ebenfalls variiert, zwischen 1 % bis 100 % der Gesamtanzahl an Voxeln. Dazu wurden Kugeln zufällig im Voxelraum platziert und die dadurch belegten Voxel aufsummiert, solange bis die gewünschte Gesamtanzahl belegter Voxel erreicht war. Abbildung 9.13 zeigt ein Ergebnis, das mit seiner Charakteristik als stellvertretend für alle Ergebnisse dieses Experiments betrachtet werden kann. Die durchgezogenen Linien zeigen die Ergebnisse aus der Verwendung unsegmentierter Bilder. Diese sollen zunächst erläutert werden. Erkennbar schneidet das Texture Mapping am schlechtesten ab (grüne Linie). Obwohl es am meisten von den gegebenen Hardware-Funktionen für „Vertices“ (Knoten, Eckpunkte) Gebrauch macht, ergibt sich viel Mehraufwand aus der Projektion nicht-sichtbarer Ebenen in die Kameras. Das Standard-Raycasting mit fester Schrittweite (blaue Linie) sowie die Optimierung von Amanatides (rote Linie) erzeugen ähnliche Ergebnisse. Die Kosten für die Schnittpunktberechnung zwischen Sichtstrahlen und Voxeln – zur Reduktion der Anzahl an Schritten beim Amanatides-Verfahren – stehen den Kosten gegenüber, welche für die größere Anzahl an Iterationen beim Standard-Raycasting benötigt werden. Die Verwendung der Compute-Shader-Implementierung der Amanatides-Methode (orangene Linie) führte zu den besten Frameraten. Dies bedeutet, dass die Verwendung der gesamten OpenGL-Pipeline in den Raycasting-Methoden, welche auf dem Vertex- und Fragment-Shader basieren nicht sinnvoll ist, weil dadurch ein zu großer

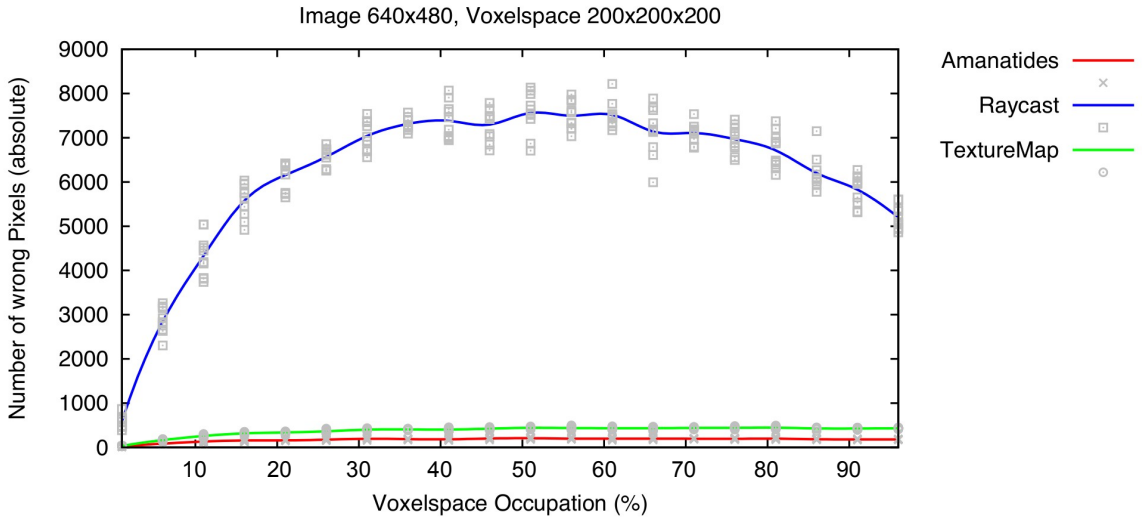


Abb. 9.14: Absolute Anzahl falsch berechneter Pixel im Vergleich zu Ground-Truth-Daten eines „naiven Renderings“. Abbildung entnommen aus [Zwicker, 2013, S. 77].

Mehraufwand entsteht. Ein weiteres Ergebnis, das aus Abb. 9.13 hervorgeht, ist die steigende Framerate mit einer Vergrößerung der Voxelraumbelegung. Dies gilt für alle Methoden gleichermaßen und resultiert aus den für jedes Pixel im Mittel kürzeren Ray-Marching-Distanzen bis zum ersten Schnitt mit einem belegten Voxel.

Die Verwendung segmentierter Silhouettenbilder wurde ebenfalls untersucht, da diese für die Rekonstruktion einer Visuellen Hülle verwendet werden. Die gestrichelten Linien in Abb. 9.13 mit der Bezeichnung „Masked “ repräsentieren die Ergebnisse. Die Dauer der Silhouettenbilderzeugung ist nicht in den Messungen enthalten. Wie erwartet, profitieren alle Raycasting-Methoden vom Auslassen der Pixel, die sich außerhalb der Silhouetten befinden. Dies führt insbesondere bei kleineren Voxelraumbelegungen zu einer deutlichen Reduktion der Berechnungsdauer, da hierbei fast keine Pixel abgearbeitet werden müssen. Die Effizienz des Texture Mappings kann dadurch jedoch nicht gesteigert werden, weil auch bei segmentierten Bildern immer alle Ebenen gerendert werden.

Neben der Berechnungsdauer wurde auch die Qualität der Rendering-Methoden untersucht. Ein „naives“ Rendering wurde zur Generierung von Ground-Truth-Daten verwendet, welches keine numerischen Approximationen vornimmt und damit Ergebnisse höchster Qualität liefert. Für kleine Voxelraumauflösungen wurde dies auf der CPU validiert. In dem Experiment wurde im Vergleich zum naiven Rendering die absolute Anzahl falscher Pixel gemessen (vgl. Abb. 9.14) sowie die mittlere Abweichung der tatsächlichen Voxelposition von der idealen, die in jedem Pixel kodiert wird (vgl. Abb. 9.15). Dazu wurde für jedes Voxel die Manhattan-Distanz zwischen den gerenderten Ist- sowie den Soll-Voxelkoordinaten von der Ground-Truth berechnet. Eine Messung für einen Voxelraum der Größe 200^3 wird in den Abbildungen 9.14 und 9.15 gezeigt.

Da beide Implementierungen der Amanatides-Methode zum selben Ergebnis führen, ist jeweils nur ein Graph dafür dargestellt.

Die Standard-Raycasting-Methode (blaue Linie) erzeugte die schlechtesten Ergebnisse. Das Voxel-Rendering wurde für 2,5 % aller Pixel falsch berechnet (vgl. Abb. 9.14). Selbst eine Ray-Marching-Schrittweite von 1/5 der kürzesten Voxelseite resultierte in vielen falschen Voxel-Pixel-Zuordnungen mit Distanzen von durchschnittlich 8 Voxeln (vgl. Abb. 9.15). Dies kann die Carving-Entscheidung im Algorithmus der Photohülle beeinflussen, da der Photokonsistenztest mit falschen Farben für diese Voxel durchgeführt werden könnte, wenn die Pixel-Voxel-Korrespondenz nicht stimmt. Die besten Ergebnisse wurden mit dem Amanatides-Rendering erzielt (rote Linie). Mit im Schnitt nur 0,08 % falschen Pixeln (vgl. Abb. 9.14) können die Fehler hierbei vernachlässigt werden. Zudem ist die durchschnittliche Abweichung von 2 bis 3 Voxeln (vgl. Abb. 9.15) für den Amanatides-Algorithmus signifikant geringer als für das Raycasting. Das Texture Mapping (grüne Linie) schnitt bezüglich der Gesamtanzahl an Falschzuordnungen von Pixeln (vgl. Abb. 9.14) etwas schlechter ab als der Amanatides-Algorithmus. Dies ist jedoch subjektiv bewertet in den erzeugten Bildern nicht sichtbar. Die auf der Grafikkarte ausgeführte Berechnung dient hierbei auch lediglich zur Optimierung der Geschwindigkeit, nicht der Genauigkeit. Die durchschnittliche Standardabweichung war beim Texture Mapping etwas besser als beim Amanatides-Algorithmus (vgl. Abb. 9.15), was jedoch vernachlässigt werden kann.

Zusammenfassend lässt sich aus den Experimenten zu den Rendering-Methoden festhalten, dass mit der Compute-Shader-Implementierung der Amanatides-Methode insgesamt

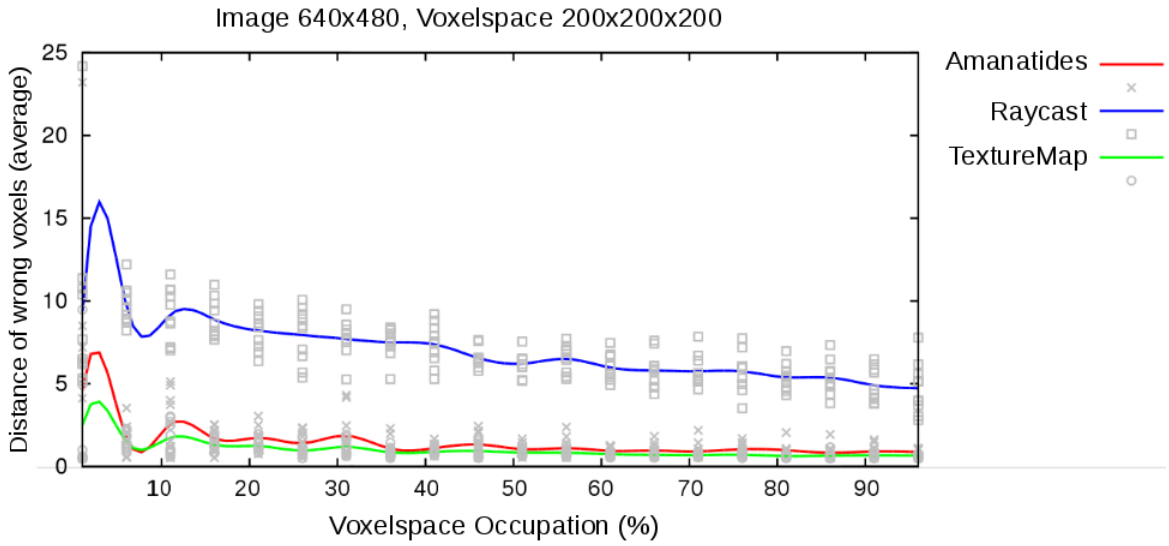


Abb. 9.15: Durchschnittliche Manhattan-Distanz zwischen den gerenderten Ist-Voxelkoordinaten und den Soll-Voxelkoordinaten eines „naiven Renderings“. Abbildung entnommen aus [Zwicker, 2013, S. 77].

die besten Ergebnisse erzielt werden konnten. Es liefert die beste Berechnungszeit (Rendering von 2000 Bildern pro Sekunde) und zeigt auch bei den qualitativen Untersuchen mit die besten Ergebnisse (ähnlich denen des Texture Mappings). Aus diesen Gründen wurde die Compute-Shader-Implementierung in den nachfolgenden Experimenten eingesetzt.

Analyse der Photohülle

In dem nachfolgend beschriebenen Experiment aus [Zwicker, 2013] wurde die vollständige Verarbeitungskette der Rekonstruktion anhand einer Bildersequenz der SIMERO-Roboterarbeitszelle (vgl. Abb. 1.2) untersucht. Wie beschrieben, stand ein Überwachungssystem mit sieben kalibrierten Kameras der Auflösung 640×480 Pixel des Typs „Unibrain Fire-i400“ zur Verfügung. Der Öffnungswinkel der gewählten Objektive betrug $110\,^\circ$ in der Diagonale. Jede Kamera war verbunden mit einem separaten Vorrechner, der die Bilder aufzeichnete und diese via Gigabit-Netzwerk an den Hauptrechner übertrug. Es wurden Bildersequenzen mit bis zu 4260 Frames untersucht.

Zwei bis drei einfarbig gekleidete Personen bewegten sich in der SIMERO-Zelle, so wie in Abb. 9.16 gezeigt. Die Kleidung verbesserte die Vordergrundsegmentierung durch das Background Subtraction. Die statischen Objekte im Raum wurden durch Dreiecksnetze dreidimensional modelliert und, wie beschrieben, zur Erzeugung synthetischer Tiefenbilder verwendet. Diese werden in den Rekonstruktionsprozess integriert, um einen Umgang mit den statischen Verdeckungsvolumina zu ermöglichen. Die räumliche Ausdehnung des Voxelraums wurde so gewählt, dass die Roboterarbeitszelle damit vollständig ausgefüllt ist. Die Decke, die Wände, der Boden, sowie die Kameras wurden vom Voxelraum jedoch nicht berührt und deshalb auch nicht modelliert. Zusätzlich zu den aufgezeichneten Bildern der Realwelt diente eine virtuelle Bildersequenz der Bewertung der Rekonstruktionsqualität.

Zunächst wurde anhand einiger Tests eine Obergrenze von 800 ms für die zur Verfügung stehende Berechnungzeit der Photohülle festgelegt. Das Überschreiten dieser Grenze führte in jeder weiteren Iteration zu einer lediglich geringen Anzahl verworfener Voxel, sodass eine weitere Verbesserung der Rekonstruktionsqualität als vernachlässigbar betrachtet werden konnte. Anschließend wurde die Anzahl an Iterationen untersucht, welche für die vorgegebene Berechnungszeit ausgeführt werden konnte. Bei diesem Experiment wurde die Kameranzahl von 4 auf 7 erhöht sowie die Voxelraumgröße von 100^3 auf 300^3 Voxel, mit einer Schrittweite von 50^3 Voxeln. Die Ergebnisse werden in Abb. 9.17 gezeigt.

Bei einer Erhöhung der verwendeten Kameraanzahl um 75 Prozent (von 4 auf 7)

reduzierte sich die Anzahl der durchgeführten Iterationen von 650 auf 500, was 23 Prozent weniger Iterationen entspricht. Eine Erhöhung der Voxelraumauflösung von 200^3 auf 250^3 Voxel führte zu einer Verringerung der Anzahl an Iterationen um 28 Prozent, von 350 auf 250. Überraschenderweise beeinflusst die Kameraanzahl die Ergebnisse ähnlich wie die Voxelraumauflösung für die gewählten Parameterkombinationen, obwohl die Berechnung der Photokonsistenz iterativ je Bild erfolgt und hauptsächlich von der Kameraanzahl abhängen sollte.

In dem Experiment zeigte sich weiterhin, dass ein großer Teil der Berechnungszeit (bis zu 50 % bei 300^3 Voxeln) für die Vorbereitung und das Zurücksetzen der Datenstrukturen benötigt wurde. Dies wird in Abb. 9.18 für die gewählten 800 ms Verarbeitungszeit dargestellt. Nach jedem Frame müssen die Voxeldatenstrukturen V_{free} und V_{occupied} genullt werden. Zudem wird ein Reset am Ende jeder Iteration für den Zähler benötigt, der für die inkrementelle Berechnung des LRT-Farbkonsistenztests verwendet wird. Diese Durchführung betrifft jedes Voxel, weshalb die Kameraanzahl hierbei irrelevant ist. Eine Erhöhung der Berechnungszeit in Abhängigkeit von der Größe der Voxelraumauflösung geht aus Abb. 9.18 hervor. Der verwendete OpenGL-Standard der Version 4.2 ermöglichte keine Verbesserung der Vorbereitung und Zurücksetzung der Datenzugriffe. Aber für neuere Versionen von OpenGL (Version 4.4 [Segal und Akeley, 2013] oder Version 4.5) sollte eine effektivere Handhabung zur Verfügung stehen. Die Berechnungszeit für das Rendering sowie die Aktualisierung der Farbkonsistenz für einzelne Voxel erhöht sich, wie erwartet, mit der Voxelraumauflösung (vgl. „rendering“ in Abb. 9.18). Diese

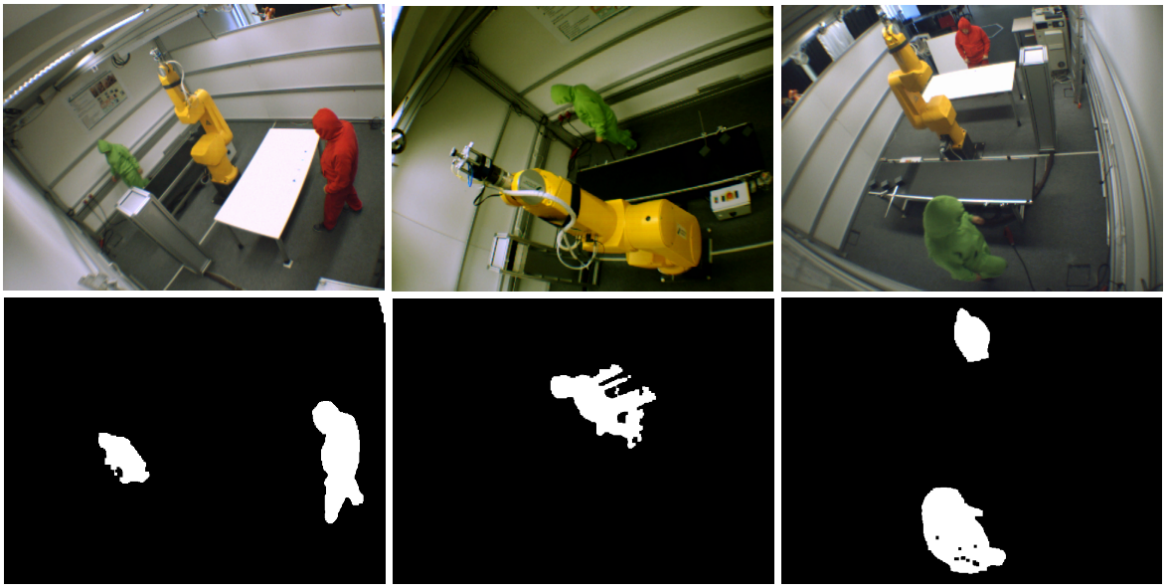


Abb. 9.16: Zwei Personen in der SIMERO-Roboterarbeitszelle aus Sicht dreier Kameraperspektiven. Farbbilder (oben) und Silhouettenbilder generiert durch ein Background Subtraction (unten). Die Ergebnisse des Background Subtraction sind im Allgemeinen nicht ideal. Abbildungen entnommen aus [Zwicker, 2013, S. 78].

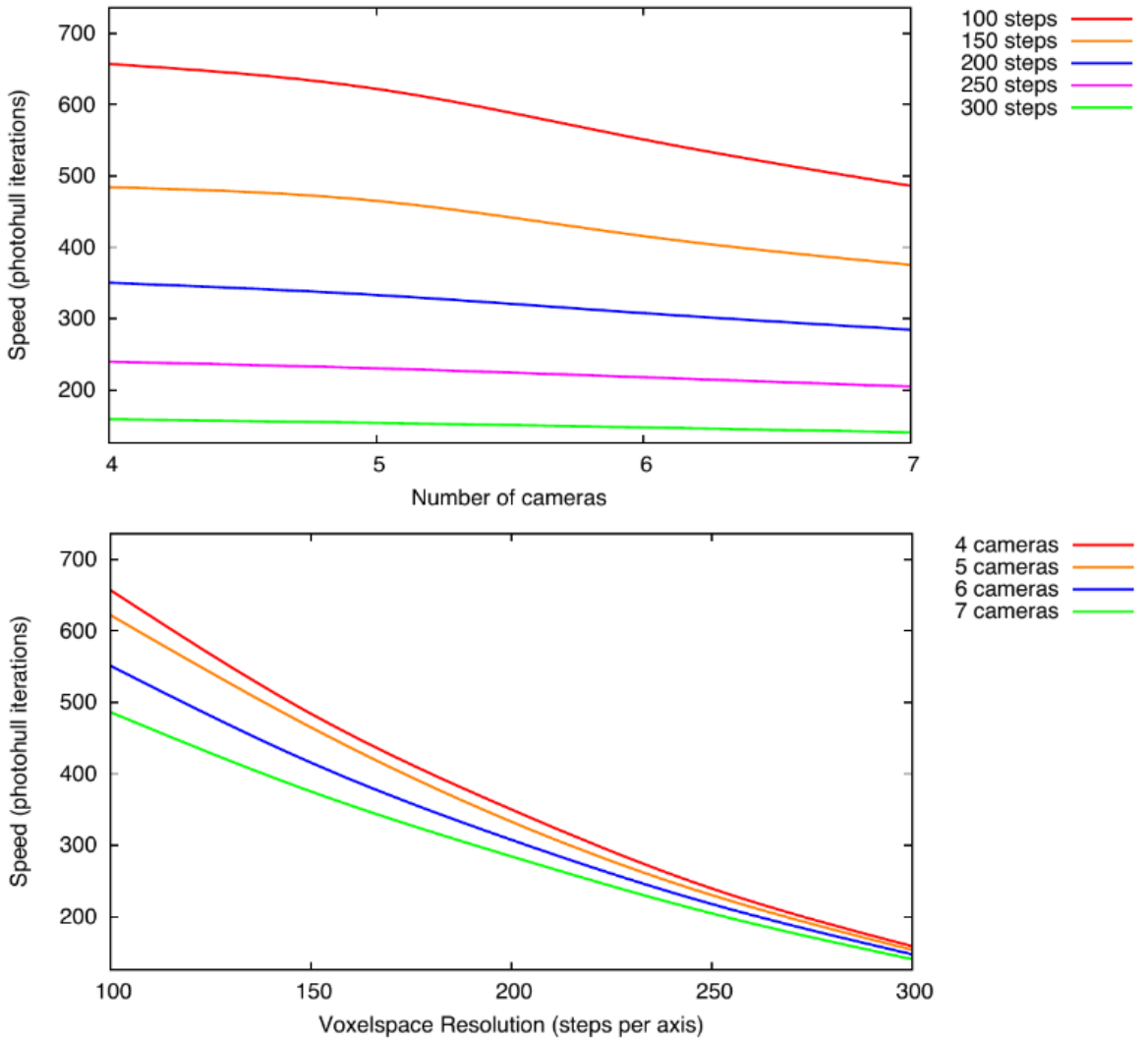
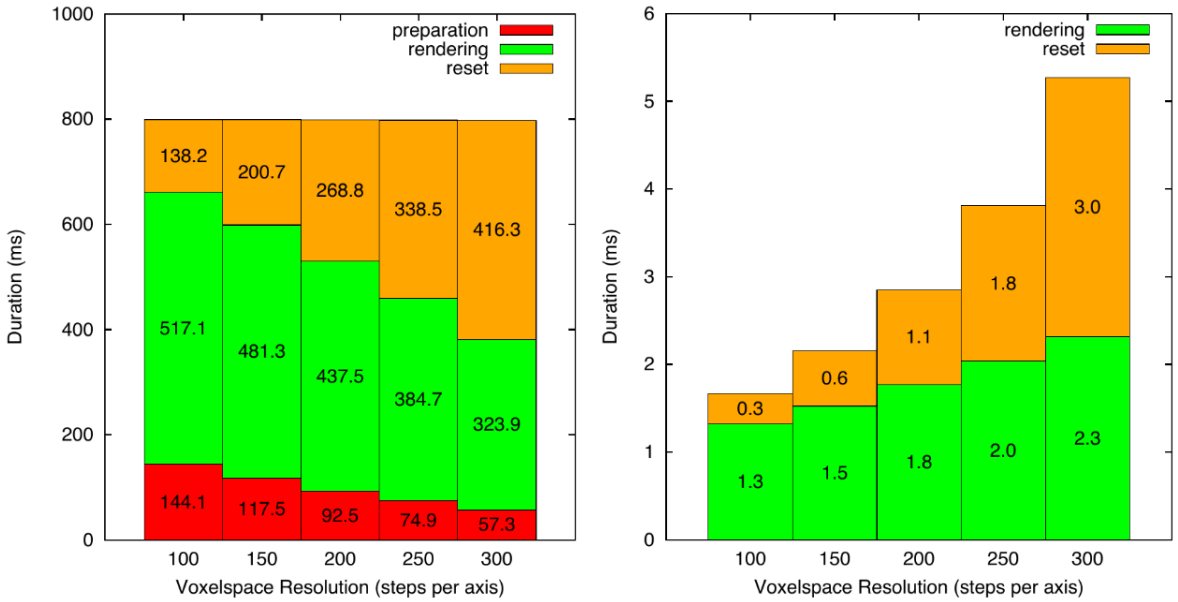


Abb. 9.17: Iterationen der Photohülle für eine variierende Anzahl an Kameras (oben) und variierende Voxelraumauflösungen (unten). Abbildungen entnommen aus [Zwicker, 2013, S. 82].

Korrelation kann durch die Verwendung des Amanatides-Raycasting erklärt werden, da die Schrittweite für die Bestimmung der Schnittpunkte zwischen Sichtstrahlen und Voxeln auf die Voxelgröße und damit auf die Voxelraumauflösung adaptiert ist.

Ein weiteres Experiment in [Zwicker, 2013] zielte auf die Anwendbarkeit des Verfahrens in der realen Arbeitszelle von ca. 40 m^3 mit 7 Kameras und einer gewählten Voxelraumauflösung von 200^3 Voxeln ab. Die Voxelseitenlängen waren jeweils 19, 12 und 21,5 mm lang. Das Ziel bestand darin, die höchste Framerate herauszufinden, die noch zu akzeptablen Ergebnissen führt. Zunächst wurde dafür ein optimierter Schwellenwert von $\tau = 7{,}5$ für den LRT bestimmt. Für die gegebenen 800 ms Berechnungszeit konnte damit die subjektiv beste Qualität erzeugt werden. Anschließend wurde die Berechnungszeit schrittweise reduziert und Bilder der erzeugten Photohülle subjektiv ausgewertet. Als Ergebnis zeigte sich, dass die meisten Voxel bereits innerhalb der ersten 200 ms entfernt



(a) (b)

Abb. 9.18: Dauer der einzelnen Berechnungsschritte der Photohülle (Preparation, Rendering, Reset) für variierende Voxelraumauflösungen. Durchschnittliche Zeiten (in ms) über alle Kameras und alle Iterationen (a), durchschnittliche Zeiten über alle Kameras für eine einzelne Iteration (b). Abbildungen entnommen aus [Zwicker, 2013, S. 83].

wurden und die nachfolgenden kleinen Verbesserungen ignoriert werden konnten, ohne einen großen Qualitätsverlust zu erzeugen.

Die erreichbaren Berechnungszeiten werden in Abb 9.19 gezeigt. Die Werte sind über die ersten 200 Frames für jede Sequenz gemittelt. Für das gegebene Anwendungsszenario konnte eine Rekonstruktionsrate von mehr als 4 fps erzielt werden. Die variierende Personenanzahl von 1 bis 3 in den Videosequenzen führte zu keinem signifikanten Unterschied hinsichtlich der Berechnungszeit.

Sequence	**Persons**	**Visual Hull**	**Photo Hull**		**Total**	
			Iteration	**Total**	**Time**	**fps**
Simulation	1	22 ms	8.2 ms	200 ms	222 ms	4,5
Real-world 1	2	29 ms	5.1 ms	200 ms	229 ms	4,3
Real-world 2	3	28 ms	8.1 ms	200 ms	228 ms	4,4

Abb. 9.19: Berechnungszeiten der beschleunigten Visuellen Hülle und Photohülle gemittelt über die ersten 200 Frames für jede Sequenz. Die gesamte Berechnungszeit für die Photohülle wurde auf 200 ms begrenzt. Damit konnte eine Online-Rekonstruktionsrate von mehr als 4 Frames pro Sekunde (fps) für das SIMERO-Szenario erreicht werden. Tabelle entnommen aus [Zwicker, 2013, S. 85].

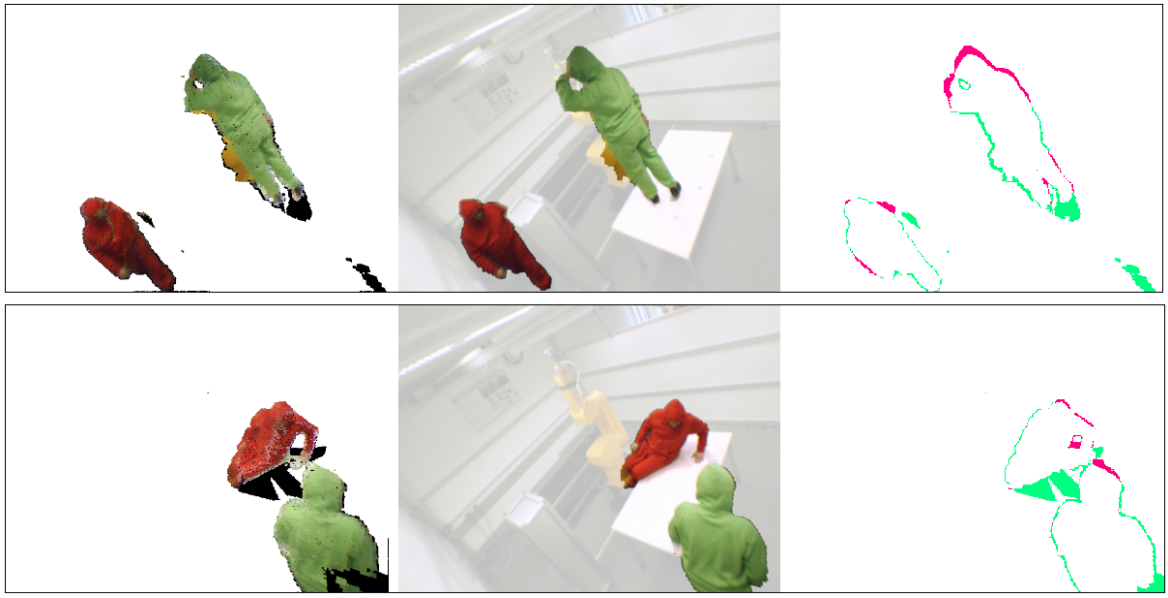


Abb. 9.20: Photohülle von Personen in der SIMERO-Roboterarbeitszelle. Rekonstruktionsergebnisse (links), Kamerabilder mit detektierten Silhouetten des Background Subtractions (mittig), Analyse der Objektkonturen (rechts). Rote Konturpixel markieren Regionen, die in der Rekonstruktion fehlen. Grüne Konturpixel markieren Bereiche, die zusätzlich rekonstruiert wurden. Solche Störungen entstehen z. B. durch Quantisierungsfehler oder eine ungenaue Kamerakalibrierung. Abbildungen entnommen aus [Zwicker, 2013, S. 86].

In einem abschließenden Experiment wurde die Qualität der Photohülle untersucht. Die Photointegrität von Seitz und Dyer [Seitz und Dyer, 1999] fordert, dass die Projektion der Rekonstruktion in die Kameras die Originalbilder reproduziert. Aus den Abbildungen 9.20 und 9.21 geht hervor, dass dies hinreichend erfüllt ist. Zu sehen sind jedoch auch Störungen, die aufgrund von Diskretisierungsfehlern, einer ungenauen Kamerakalibrierung, ungenau modellierten statischen Objekten, Fehlern beim Background Subtraction oder aufgrund anderer Ursachen entstanden sind. Solche Effekte treten in der Simulationssequenz weniger stark auf, wie aus Abb. 9.21 hervorgeht.

Eine weitere Voraussetzung, die erfüllt sein muss, um eine hohe Qualität der Rekon-

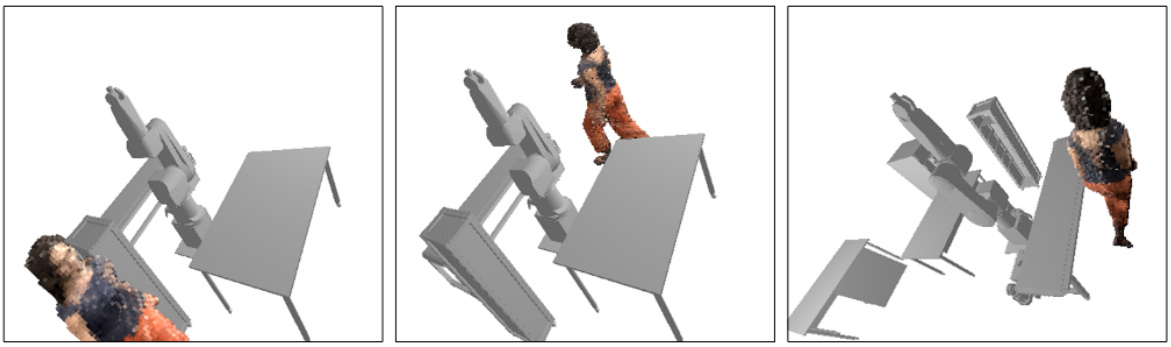


Abb. 9.21: Photohülle einer Person aus unterschiedlichen Perspektiven (Simulationssequenz). Abbildung entnommen aus [S. 88] [Zwicker, 2013].

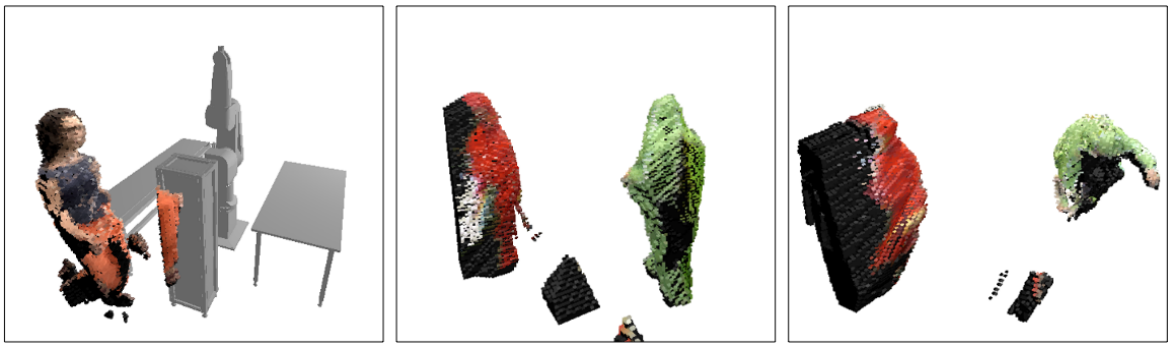


Abb. 9.22: Photohülle mit Rekonstruktionsstörungen an den Rändern der Roboterarbeitszelle, wo der Sichtbarkeitsgrad unzureichend ist. Abbildungen entnommen aus [Zwicker, 2013, S. 89].

struktion der Photohülle zu erreichen, ist der hohe Sichtbarkeitsgrad in jedem Punkt der Szene, genannt „Broad Viewpoint Coverage“ (vgl. [Seitz und Dyer, 1999]). Da nur sieben Kameras verwendet wurden, waren die Rekonstruktionsergebnisse an den Rändern des Überwachungsraums und in anderen Bereichen mit geringem Sichtbarkeitsgrad unzureichend. Dies ist in Abb. 9.22 zu sehen. In der Mitte der Roboterarbeitszelle sind die Ergebnisse hingegen besser (vgl. Abb. 9.23).

Schlussfolgerungen

In [Zwicker, 2013] wurde ein Voxel-Carving-Algorithmus für den Einsatz zur Online-Rekonstruktion von Überwachungsszenarien untersucht. Die Umgebungen von Interesse zeichnen sich durch die Präsenz verdeckender statischer Objekte aus sowie durch eine eher geringe Anzahl zur Verfügung stehender Kameras. Der Fokus wurde auf die Beschleunigung des verwendeten Algorithmus gelegt, mit dem Ziel eine Echtzeitfähigkeit zu ermöglichen. Im Ergebnis konnten 4–5 fps mit einer neuen, GPU-basierten Implementierung des GVC-IB-Verfahrens [Culbertson et al., 1999] erreicht werden.

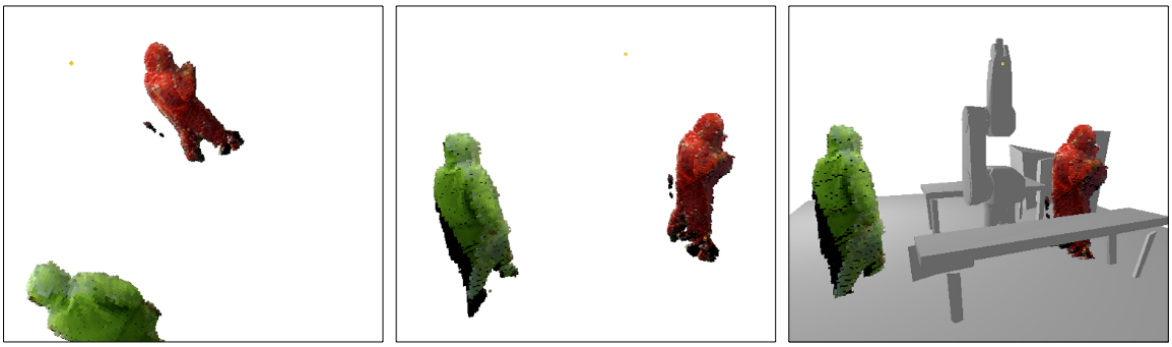


Abb. 9.23: Photohülle von zwei Personen in der Roboterarbeitszelle aus unterschiedlichen Blickwinkeln. Abbildungen entnommen aus [Zwicker, 2013, S. 88].

Vergleicht man die geometrische Approximation der Objekte durch die Photohülle mit einer Visuellen Hülle, so konnte nur eine geringfügige Verbesserung unter den gegebenen Umgebungsbedingungen mit sieben Kameras, VGA-Auflösung sowie verdeckenden statischen Objekten festgestellt werden. Grundsätzlich erfüllen die erzeugten Voxelmengen mit ihrer Kolorierung zwar das Qualitätsmaß der Photointegrität, jedoch hängt die Verbesserung der geometrischen Approximation von einer Reihe weiterer Parameter ab. Neben der gewählten Anzahl an Kameraperspektiven sowie Schatteneffekten und Reflektionseigenschaften der Oberflächen spielt auch die Texturierung der Objekte eine entscheidende Rolle. Um feine, nicht-konvexe Details aus der Oberfläche „herauszuarbeiten“, wird eine variierende Oberflächenkolorierung benötigt, damit mehr Voxel entfernt werden können. Ist ein Objekt beispielsweise gleichmäßig eingefärbt, so werden Einwölbungen nicht rekonstruiert, da bereits vorher Farbkonsistenzen entstehen, die zu einer verfrühten Terminierung des Algorithmus führen. Andererseits können stark texturierte Oberflächen auch dazu führen, dass zu viele Voxel entfernt werden, da durch die Textur unerwünschte Inkonsistenzen in Oberflächenvoxeln entstehen können. Ursachen liegen hier insbesondere in der Voxelquantisierung/-auflösung und Präzision der Kamerakalibrierung. Dieses Problem existiert auch an den Rändern der Objekte. Ein Ansatzpunkt wäre daher, erheblich mehr Rechenleistung für eine Erhöhung der Kameraanzahl und der Pixel- sowie Voxelauflösung zu investieren. Gleichzeitig impliziert eine Erhöhung der Auflösung eine erhöhte Anforderung an die Präzision der Kamerakalibrierung. Die hierbei erzielbaren qualitativen Effekte könnten Gegenstand weiterer Untersuchungen sein (unter Beibehaltung der Betrachtung statischer verdeckender Objekte). Auch könnte untersucht werden, inwieweit eine geeignete Texturierung in realen Umgebungen überhaupt gegeben ist.

9.4 Voxelsichtbarkeitsgrade

Algorithmus 9.8 Bestimmung der Voxelsichtbarkeitsgrade

```
1: procedure VoxelVisibilities(V, C, I_dep)
2:     for all v_i ∈ V do                                          ▷ for each voxel
3:         setVoxelVisibility(v_i) ← 0
4:         for all c_j ∈ C do                                      ▷ for each camera
5:             flag_visibleInAllPixels = true
6:             Φ_{v_i,c_j} ← projectVoxelToCam(v_i, c_j)
7:             for all p_l ∈ Φ_{v_i,c_j} do          ▷ for each projection pixel of the voxel
8:                 if |r_{v_i} − e_{c_j}| + d_v < getImageValue(p_l, I_dep, c_j) then ▷ if voxel is
9:                                                                                    visible
10:                else                                ▷ if voxel is not visible for the pixel
11:                    flag_visibleInAllPixels = false
12:                end if
13:            end for
14:            if (flag_visibleInAllPixels = true) then ▷ voxel visible for all projection
15:                                                                  pixels of that camera
16:                setVoxelVisibility(v_i) ← getVoxelVisibility(v_i) +1
17:            end if
18:        end for
19:    end for
20:    return V                                  ▷ with visiblity information of each voxel
21: end procedure
```

Zur Bestimmung der Voxelsichtbarkeitsgrade aller Voxel dient die Prozedur VoxelVisibilities in Algorithmus 9.8. Als Eingabe wird benötigt: die Menge aller Voxel des Voxelsraums V, das Multi-View-Kamerasystem C mit einer endlichen Menge an kalibrierten und synchronisierten Farbkameras c_j sowie die Menge der synthetischen Tiefenbilder I_{dep} von den Modellen der statischen Objekte. Gegeben sei der Zentralprojektionspunkt $e_{c_j} \in \mathbb{R}^3$ einer Kamera c_j, das Zentrum eines Voxels $r_{v_i} \in \mathbb{R}^3$ sowie eine Konstante $d_v \in \mathbb{R}$, die die Hälfte der Diagonale eines Voxels beschreibt, um konservativ zu sein. Der Betrag der Differenz der Positionsvektoren des Voxelzentrums und des Zentralprojektionspunkts wird mit der L_2-Norm berechnet und entspricht der Distanz zwischen beiden Positionen. Als Φ_{v_i,c_j} sei die Menge der Projektionspixel eines Voxels v_i in der Kamera c_j definiert.

In Algorithmus 9.8 wird die Sichtbarkeit jedes Voxels wie folgt bestimmt. Zunächst wird seine Sichtbarkeit initial auf 0 gesetzt (Zeile 3). Nun werden die Kameras nacheinander

einzeln betrachtet (Zeile 4). Das Flag *flag_visibleInAllPixels* wird für jede Kamera zu Beginn auf „true“ gesetzt, was bedeutet, dass das Voxel für alle Pixel der betrachteten Kamera als sichtbar angenommen wird (Zeile 5). Die Werte der Projektionspixel Φ_{v_i,c_j} des Voxels werden für die Kamera herangezogen (Zeile 6). Für jedes dieser Pixel wird geprüft, ob sich das Voxel aus Sicht der Kamera vor einem statischen Objekt befindet (Zeile 8). Dabei wird der Abstand des Voxels zur Kamera mit dem gespeicherten Wert des Tiefenbilds verglichen, welches die Abstände der nahesten statischen Objekte kodiert. Liegt das Voxel auch nur für ein Pixel hinter der sichtbaren Oberfläche eines statischen Objekts, so wird das Flag *flag_visibleInAllPixels* entsprechend auf „false“ gesetzt (Zeile 11). Das Voxel wird für die betrachtete Kamera als nicht sichtbar klassifiziert und der Sichtbarkeitsgrad des Voxels nicht erhöht. Ist ein Voxel hingegen für alle Projektionspixel sichtbar, was bedeutet, dass sich das Voxel vollständig vor den modellierten statischen Objekten befindet, so wird der Sichtbarkeitsgrad des Voxels um 1 erhöht. Anschließend wird die Sichtbarkeit des Voxels für alle weiteren Kameras analog ermittelt, woraus sich im Ergebnis der finale Sichtbarkeitsgrad des Voxels ergibt. Wurden die Sichtbarkeitsgrade aller Voxel bestimmmt, so wird der Voxelraum mit entsprechend annotierten Informationen als Ausgabe des Algorithmus zurückgeliefert (Zeile 20).

9.5 Weitere Diagramme

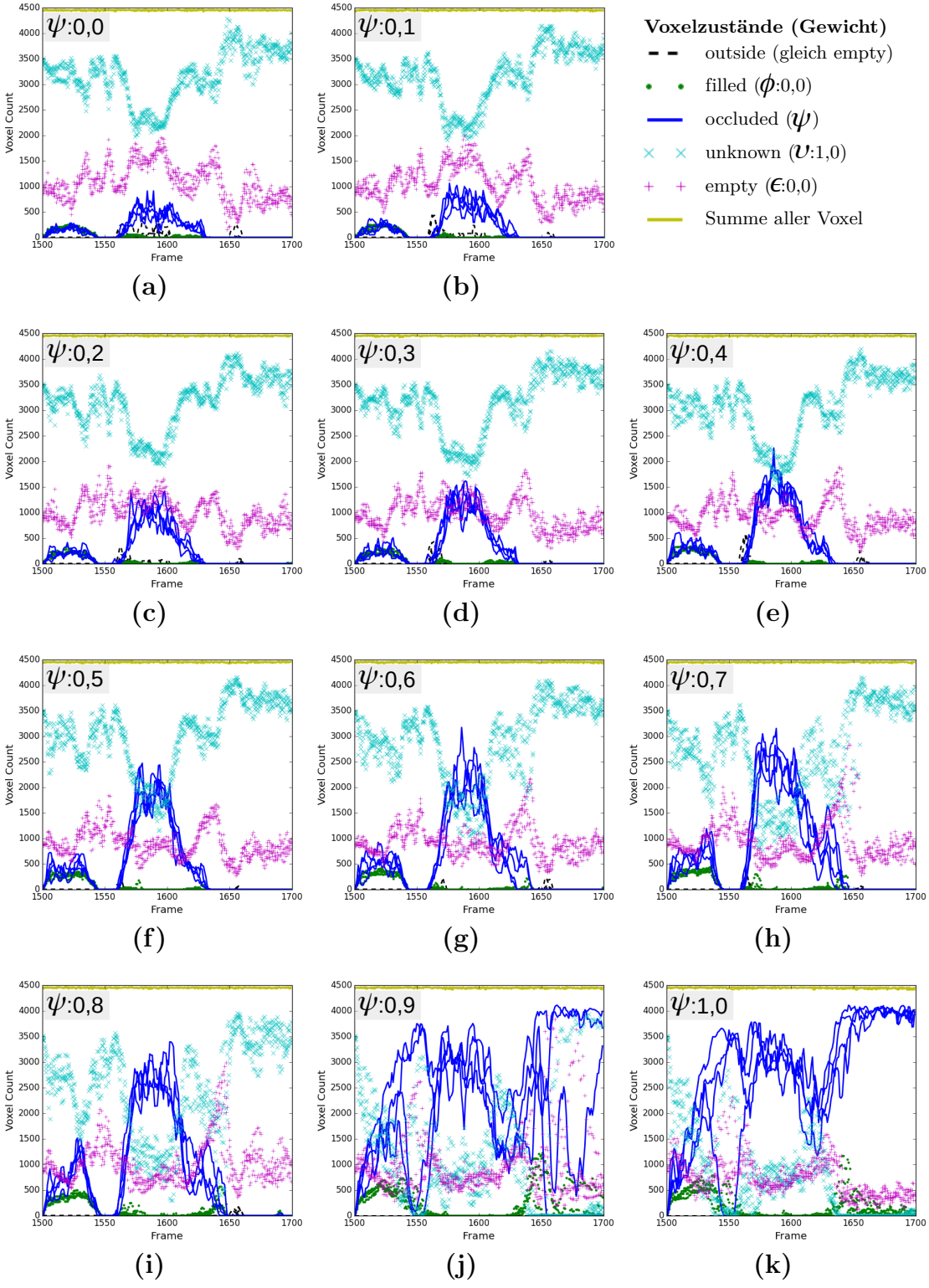


Abb. 9.24: Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel wird schrittweise erhöht von 0,0 (a) bis 1,0 (k). Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

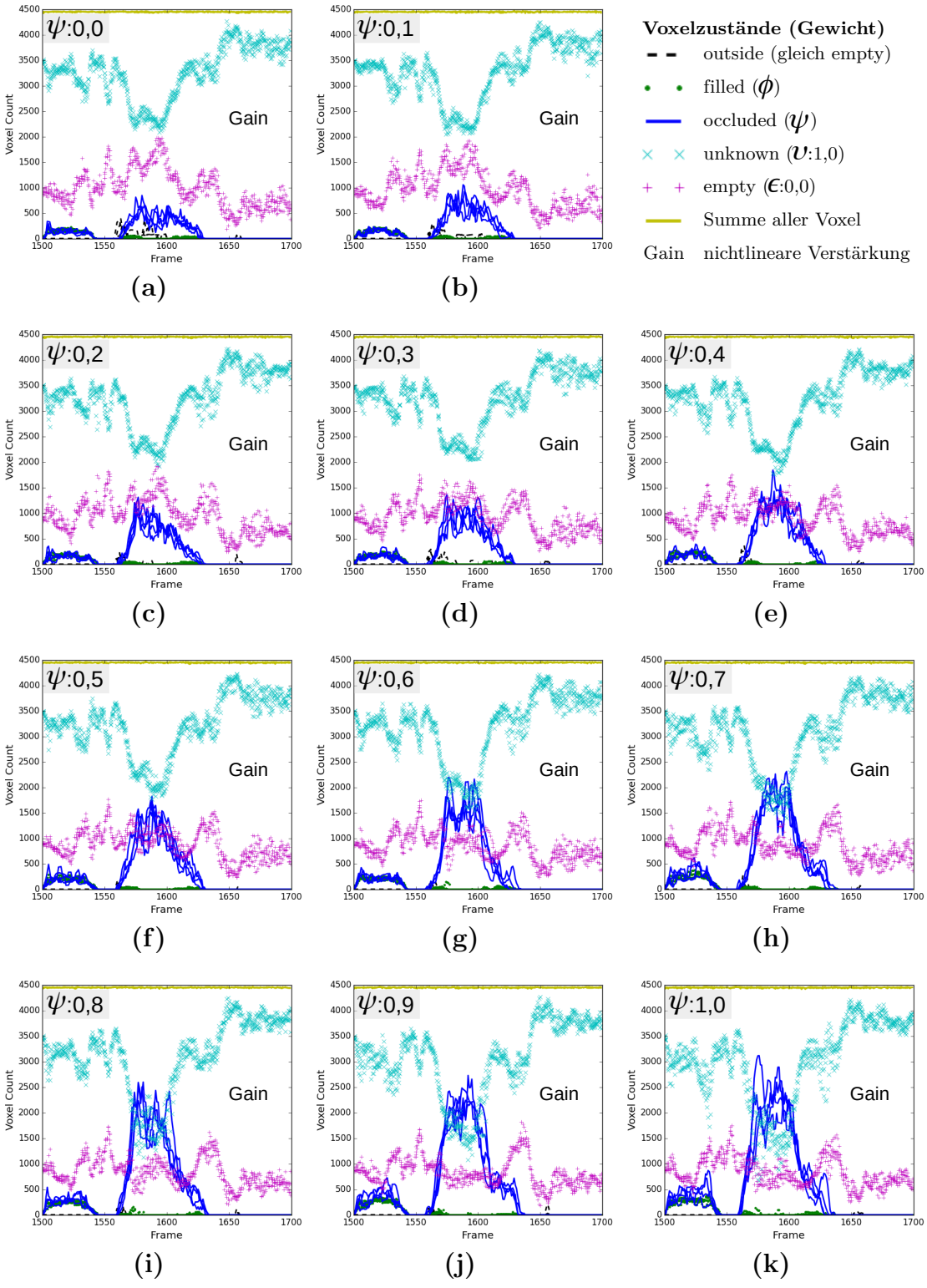


Abb. 9.25: Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel mit nichtlinearer Verstärkung (Gain) wird schrittweise erhöht von 0,0 (a) bis 1,0 (k). Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

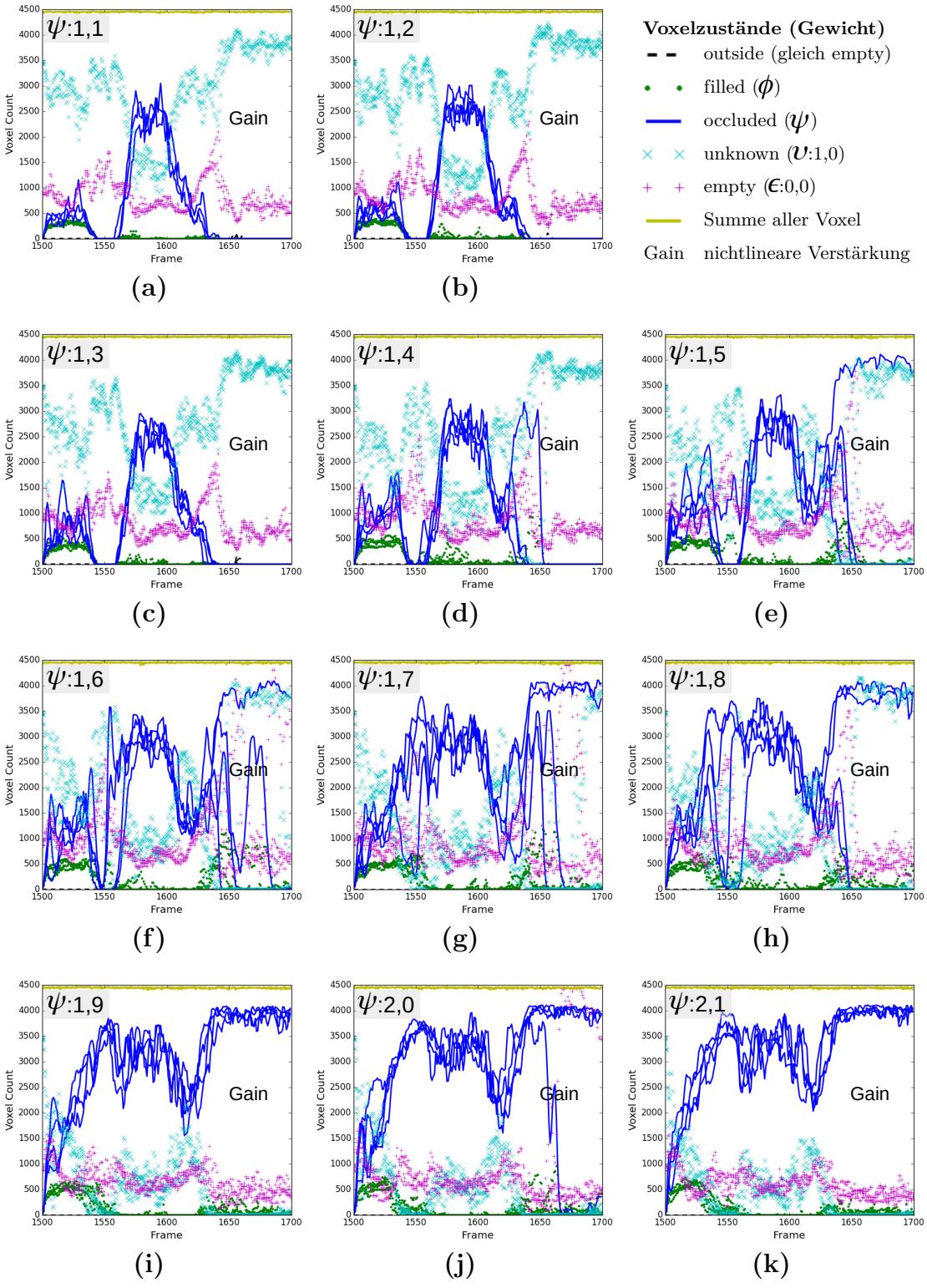


Abb. 9.26: Häufigkeit der Voxelzustände innerhalb des Schwerpunktellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel mit nichtlinearer Verstärkung (Gain) wird schrittweise erhöht von 1,1 (a) bis 2,1 (k). Das gegebene reale Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

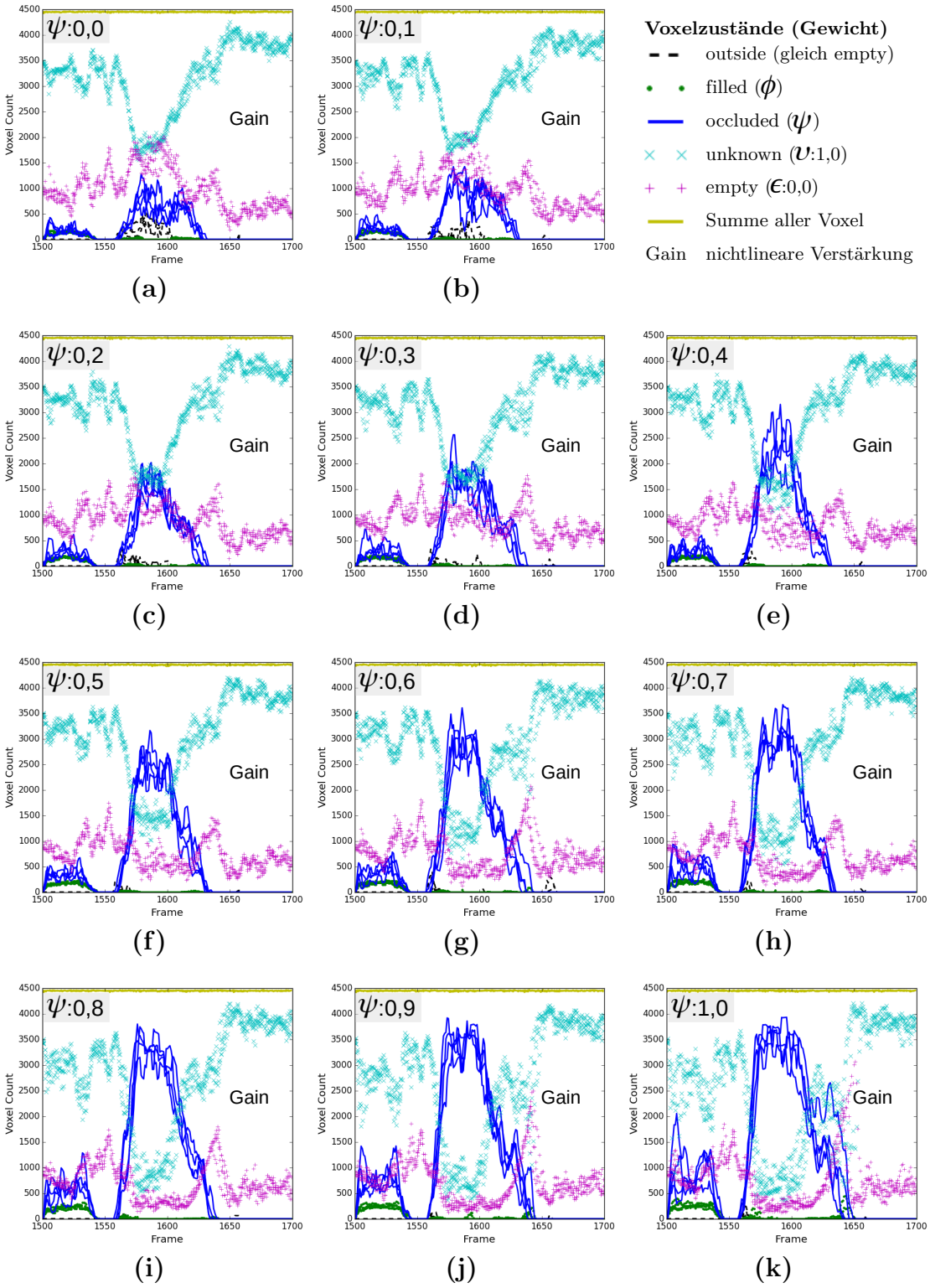


Abb. 9.27: Häufigkeit der Voxelzustände innerhalb des Schwerpunkt-Ellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel mit nichtlinearer Verstärkung (Gain) wird schrittweise erhöht von 0,0 (a) bis 1,0 (k). Das Verdeckungsvolumen unter dem Tisch wurde synthetisch vergrößert und führt zu einer partiellen Objektverdeckung der Person.

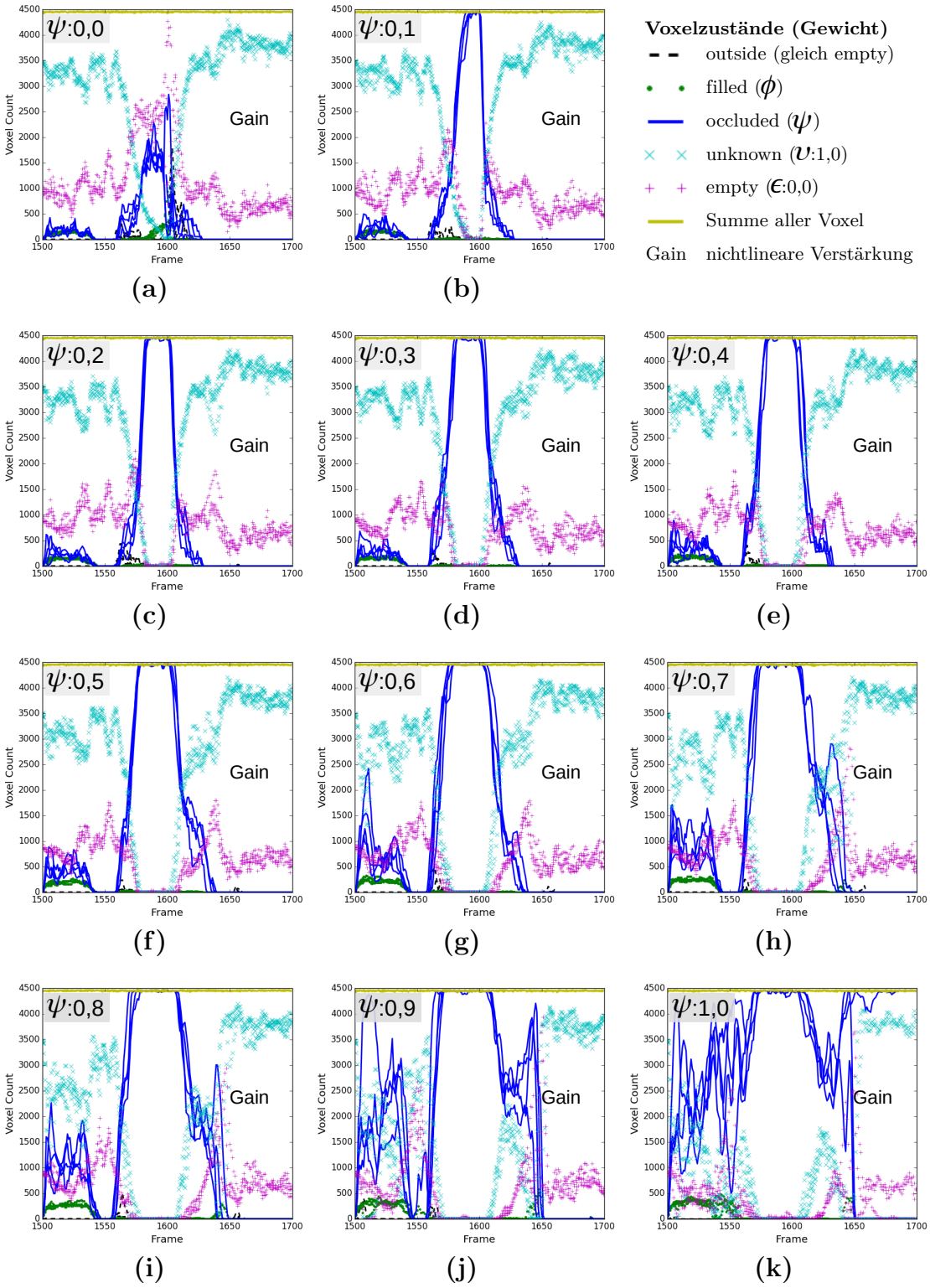


Abb. 9.28: Häufigkeit der Voxelzustände innerhalb des Schwerpunkt-Ellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel mit nichtlinearer Verstärkung (Gain) wird schrittweise erhöht von 0,0 (a) bis 1,0 (k). Das Verdeckungsvolumen unter dem Tisch wurde synthetisch vergrößert und führt zu einer vollständigen Objektverdeckung der Person.

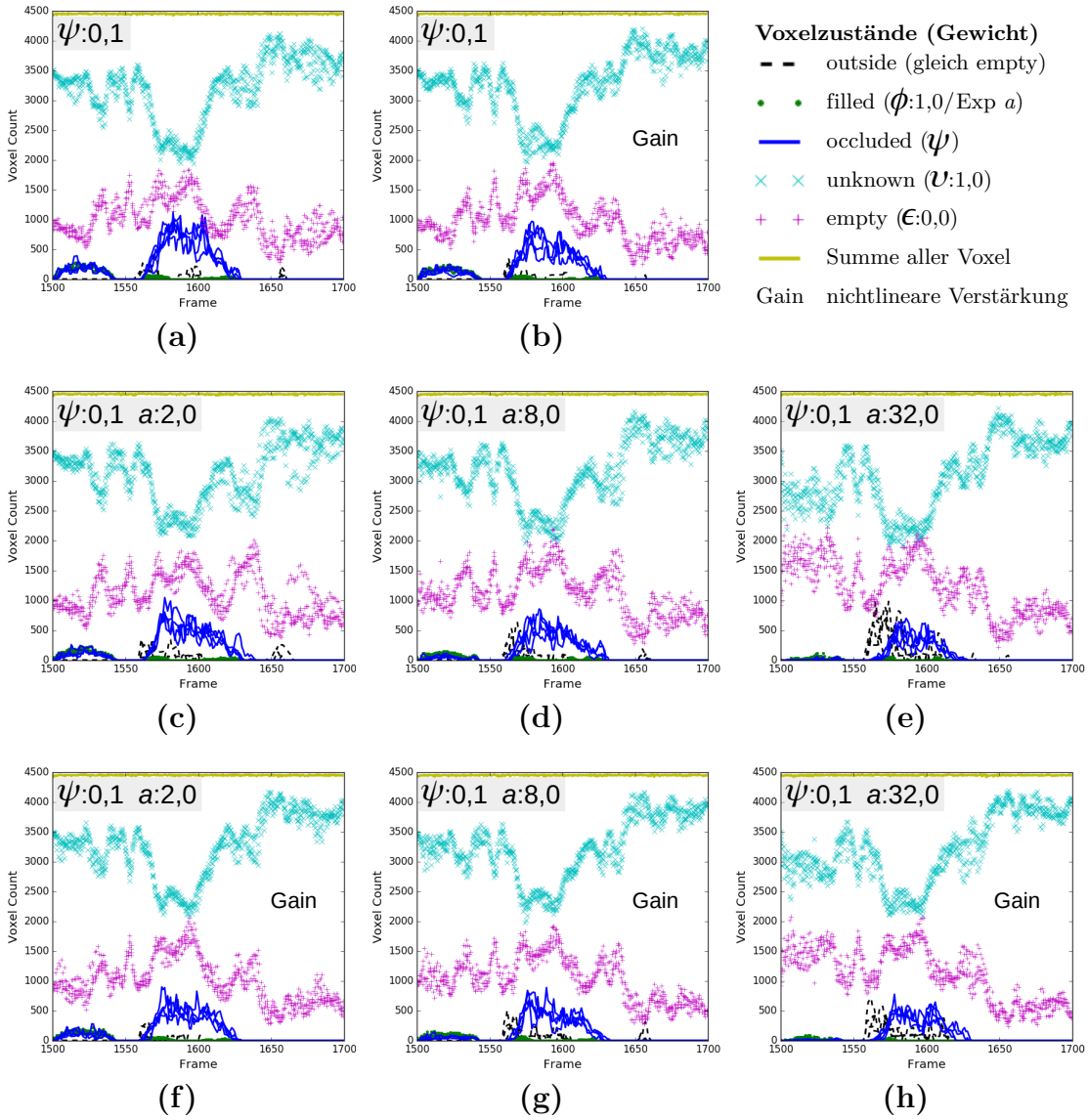


Abb. 9.29: Bestrafung von *filled*-Voxeln mit variierendem Exponenten a des Bestrafungsterms, mit und ohne nichtlinearer Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunkt-Ellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0,1$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

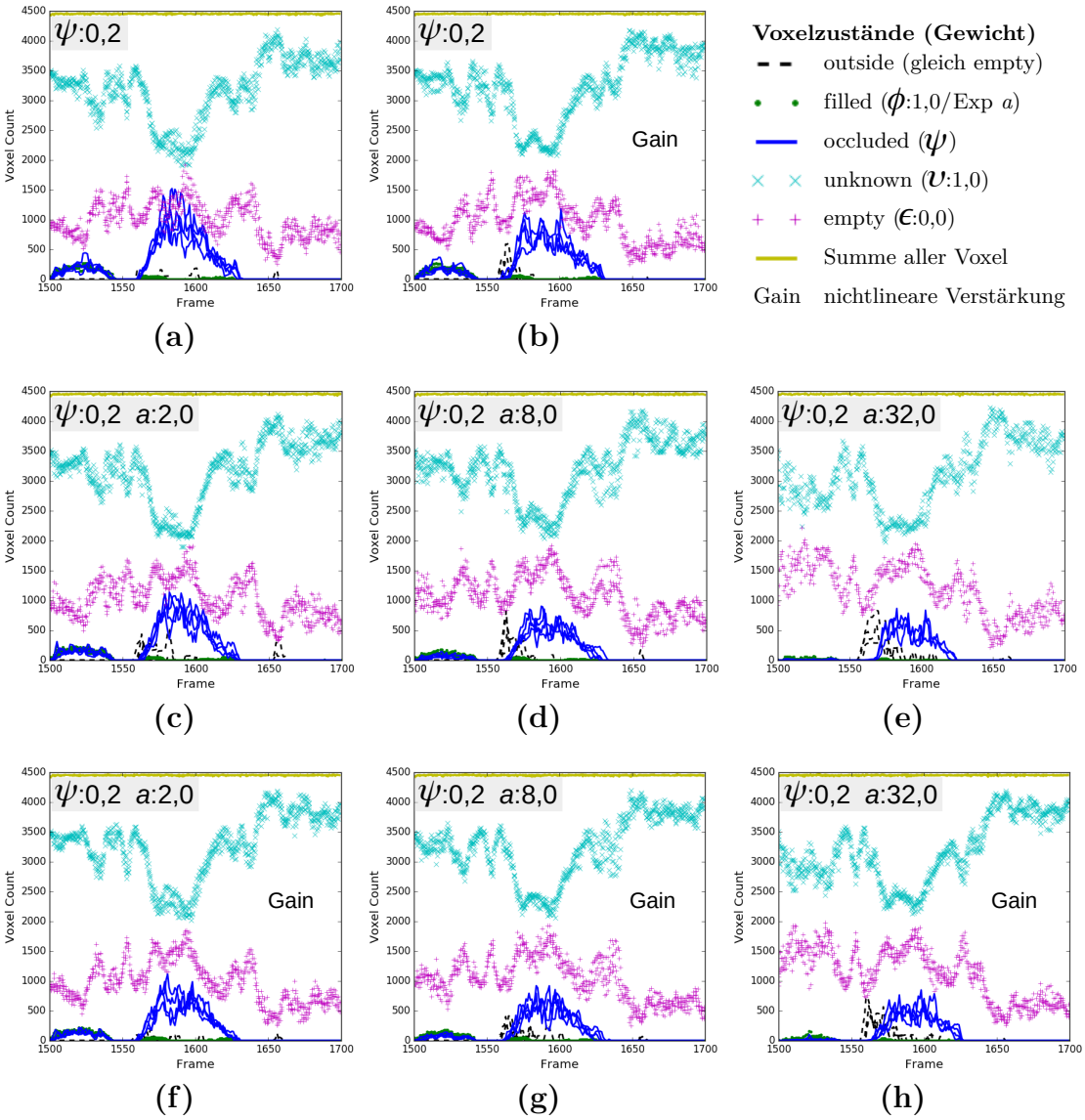


Abb. 9.30: Bestrafung von *filled*-Voxeln mit variierendem Exponenten a des Bestrafungsterms, mit und ohne nichtlinearer Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunkt-Ellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0,2$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

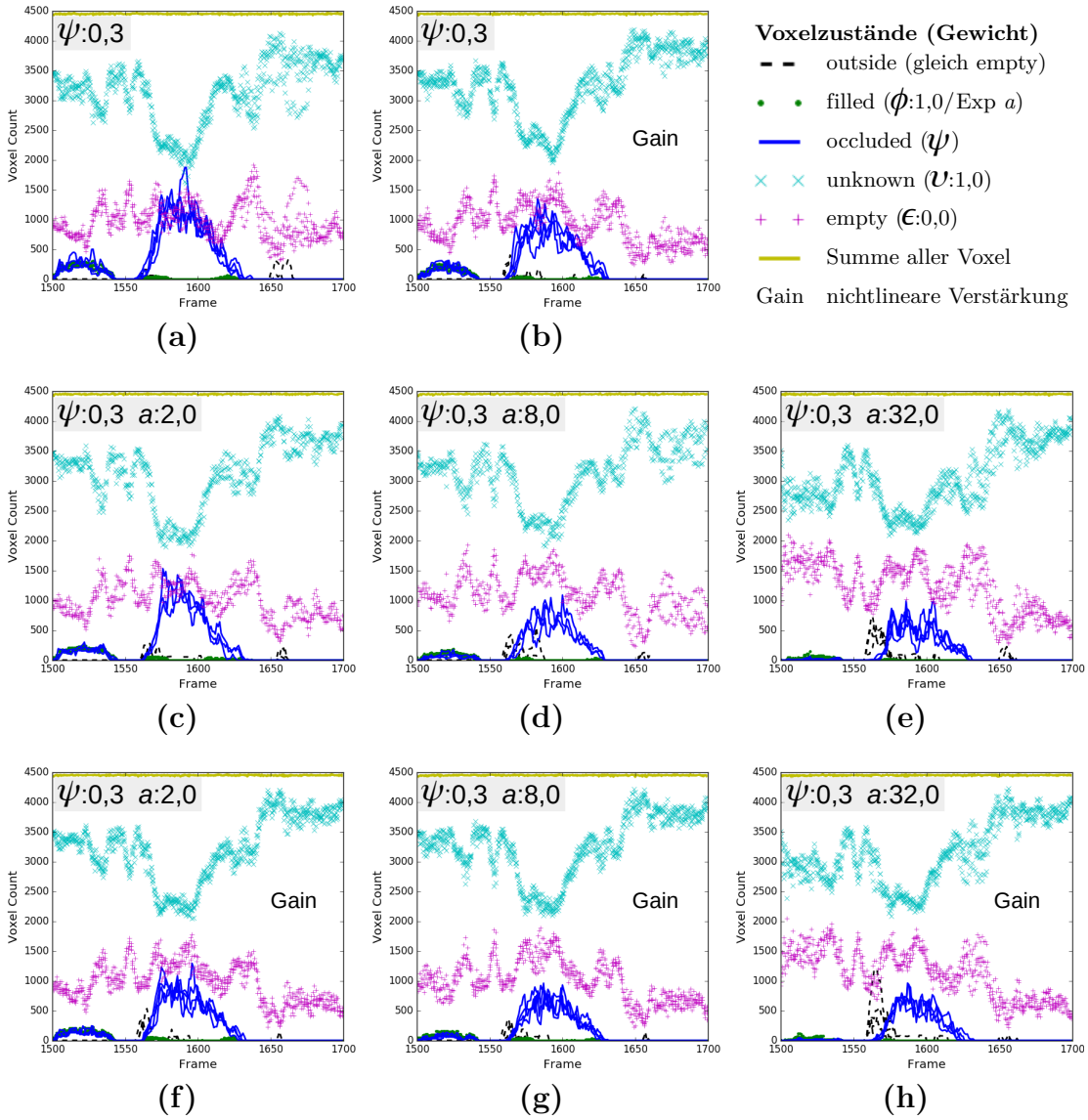


Abb. 9.31: Bestrafung von *filled*-Voxeln mit variierendem Exponenten a des Bestrafungsterms, mit und ohne nichtlinearer Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunkt-Ellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0,3$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

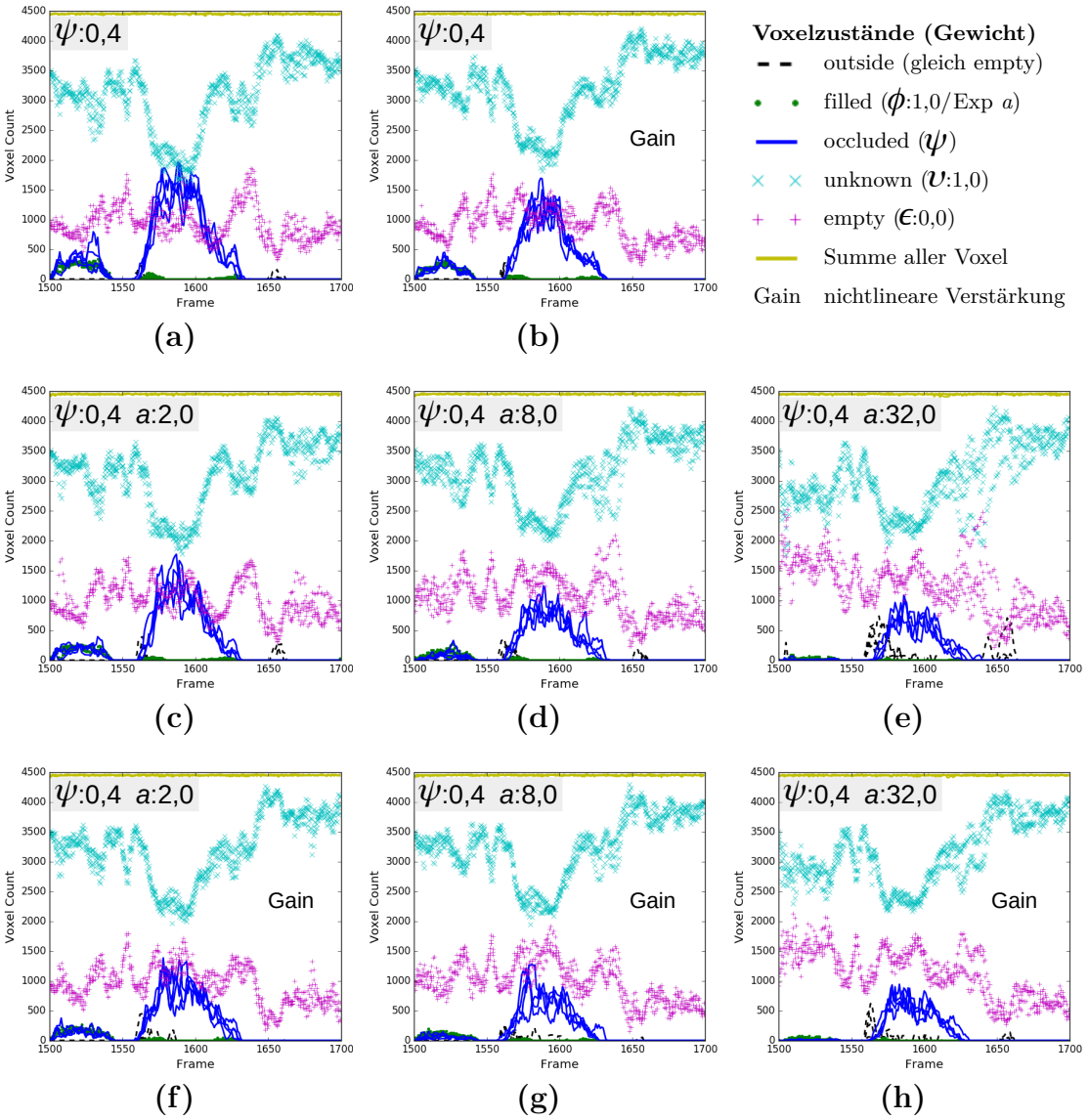


Abb. 9.32: Bestrafung von *filled*-Voxeln mit variierendem Exponenten a des Bestrafungsterms, mit und ohne nichtlinearer Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunkt-Ellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0,4$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

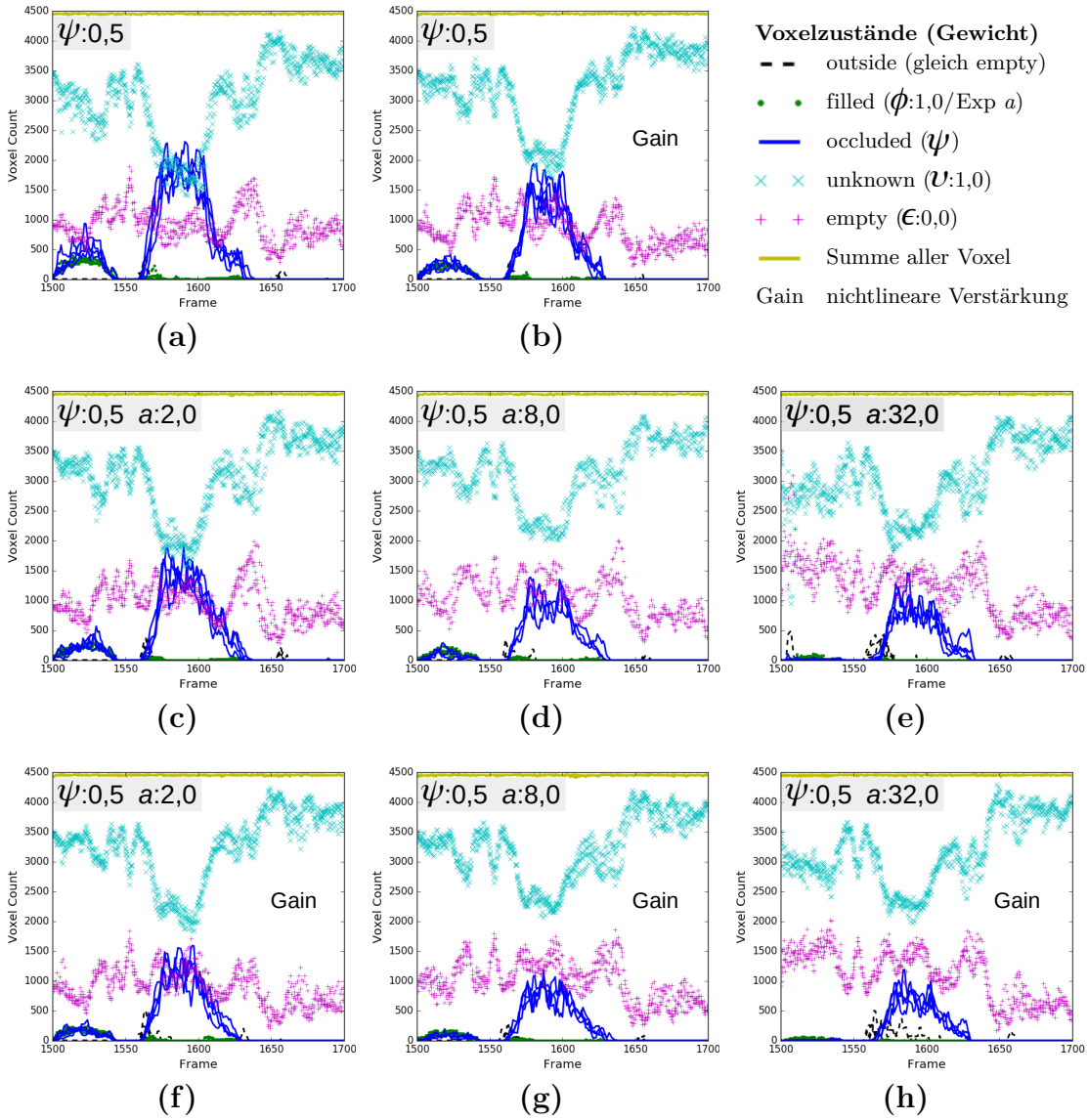


Abb. 9.33: Bestrafung von *filled*-Voxeln mit variierendem Exponenten a des Bestrafungsterms, mit und ohne nichtlinearer Verstärkung (Gain). Als Referenz werden zwei Diagramme ohne Bestrafung gezeigt (a), (b). Häufigkeit der Voxelzustände innerhalb des Schwerpunkt-Ellipsoids aufgetragen über die Frames von Teilsequenz A für alle vier Seed-Initialisierungen. Die Gewichtung der *occluded*-Voxel erfolgt in allen Fällen mit $\psi = 0,5$. Das gegebene Verdeckungsvolumen unter dem Tisch führt zu einer partiellen Objektverdeckung der Person.

Literatur

[Amanatides und Woo, 1987] Amanatides, J. und Woo, A. (1987). A Fast Voxel Traversal Algorithm for Ray Tracing. In *Proceedings of Eurographics (EG 1987-Technical Papers)*, Band 87, Seiten 3–10. Eurographics Association.

[Andriluka et al., 2010] Andriluka, M., Roth, S., und Schiele, B. (2010). Monocular 3D Pose Estimation and Tracking by Detection. *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, Seiten 623–630.

[Arulampalam et al., 2002] Arulampalam, M. S., Maskell, S., Gordon, N., und Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50:174–188.

[Ascenso et al., 2020] Ascenso, G., Yap, M. H., Allen, T., Choppin, S. S., und Payton, C. (2020). A Review of Silhouette Extraction Algorithms for Use within Visual Hull Pipelines. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 8(6):649–670.

[Asimov, 1950] Asimov, I. (1950). *I , Robot*. Bantam Books.

[Baeuerlein, 2014] Baeuerlein, F. (2014). *Hierarchisches Multi-Target-Tracking unter Berücksichtigung bekannter Hindernisse*. Masterarbeit, Lehrstuhl für Angewandte Informatik III, Universität Bayreuth.

[Bar-Shalom et al., 1988] Bar-Shalom, Y., Fortmann, T. E., Howlett, P., und Torokhti, A. (1988). *Tracking And Data Association by Yaakov BarShalom and Thomas E Fortmann*. Academic Press Boston.

[Batchelor et al., 2005] Batchelor, O., Mukundan, R., und Green, R. (2005). Ray Casting for incremental Voxel Colouring. In *Proceedings of New Zealand: International Conference on Image and Vision Computing (IVCNZ05)*, Seiten 206–211.

[Baumgart, 1974] Baumgart, B. G. (1974). *Geometric Modeling for Computer Vision.* Phd thesis, Stanford University.

[Bertsekas und Castanon, 1989] Bertsekas, D. P. und Castanon, D. A. (1989). The Auction Algorithm for the Transportation Problem. *Annals of Operations Research*, 20(1):67–96.

[Bogomjakov und Gotsman, 2008] Bogomjakov, A. und Gotsman, C. (2008). Reduced Depth and Visual Hulls of complex 3D Scenes. *Computer Graphics Forum*, 27:175–182.

[Bouwmans, 2014] Bouwmans, T. (2014). Traditional and recent Approaches in Background Modeling for Foreground Detection: An Overview. *Computer Science Review*, 11-12:31–66.

[Bouwmans et al., 2008] Bouwmans, T., Baf, F. E., und Vachon, B. (2008). Background Modeling using Mixture of Gaussians for Foreground Detection – A Survey. In *Proceedings of Recent Patents on Computer Science*, Seiten 219–237.

[Caillette, 2006] Caillette, F. (2006). *Real-time Markerless 3-D Human Body Tracking*. Phd thesis, University of Manchester.

[Canton-Ferrer et al., 2009] Canton-Ferrer, C., Casas, J. R., und Pardàs, M. (2009). Voxel based Annealed Particle Filtering for Markerless 3D Articulated Motion Capture. In *Proceedings of the 3rd 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON 2009)*.

[Canton-Ferrer et al., 2011] Canton-Ferrer, C., Casas, J. R., Pardàs, M., und Monte, E. (2011). Multi-Camera Multi-Object Voxel-based Monte Carlo 3D Tracking Strategies. *EURASIP Journal on Advances in Signal Processing*, 2011:114.

[Capek, 1920] Capek, K. (1920). *Rossum's Universal Robots*. Prague, CZ.

[Carpenter et al., 1999] Carpenter, J., Clifford, P., und Fearnhead, P. (1999). Improved Particle Filter for Non-linear Problems. *IEE Proceedings - Radar, Sonar and Navigation*, 146(1):2.

[Casas und Salvador, 2006] Casas, J. R. und Salvador, J. (2006). Image-Based Multiview Scene Analysis using 'Conexels'. In *Proceedings of HCSNet Workshop on the Use of Vision in Human-Computer Interaction (VisHCI 2006)*, Seiten 19–28.

[Chen et al., 2020] Chen, Y., Tian, Y., und He, M. (2020). Monocular Human Pose Estimation: A Survey of Deep Learning-based Methods. *Computer Vision and Image Understanding*, 192:102897.

[Cheng et al., 2020] Cheng, Y., Yang, B., Wang, B., und Tan, R. T. (2020). 3D Human Pose Estimation using Spatio-Temporal Networks with Explicit Occlusion Training. *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence (AAAI 2020)*, 34:10631–10638.

[Cheung et al., 2003] Cheung, G. K., Baker, S., und Kanade, T. (2003). Shape-from-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture. In *Proceedings of Computer Vision and Pattern Recognition*, Seiten 77–84.

[Choset et al., 2006] Choset, H. M., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., und Thrun, S. (2006). Principles of Robot Motion, Theory, Algorithms and Implementations. *Robotica*, 24:271.

[Corazza et al., 2010] Corazza, S., Mündermann, L., Gambaretto, E., Ferrigno, G., und Andriacchi, T. P. (2010). Markerless Motion Capture through Visual Hull, Articulated ICP and Subject Specific Model Generation. *International Journal of Computer Vision*, 87:156–169.

[Culbertson et al., 1999] Culbertson, W. B., Malzbender, T., und Slabaugh, G. (1999). Generalized Voxel Coloring. In *Vision Algorithms: Theory and Practice*, Band 1883, Seiten 100–115.

[De Aguiar et al., 2004] De Aguiar, E., Theobalt, C., Magnor, M., Theisel, H., und Seidel, H. P. (2004). M3: Marker-free Model Reconstruction and Motion Tracking from 3D Voxel Data. In *Proceedings of the Pacific Conference on Computer Graphics and Applications*, Seiten 101–110.

[Dean und Boddy, 1988] Dean, T. und Boddy, M. (1988). An Analysis of Time-dependent Planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, Seiten 49–54.

[Del Moral und Miclo, 2000] Del Moral, P. und Miclo, L. (2000). Branching and Interacting Particle Systems Approximations of Feynman-Kac Formulae with Applications to Non-Linear Filtering. *Séminaire de probabilités de Strasbourg*, 34:1–145.

[Deutscher et al., 2000] Deutscher, J., Blake, A., und Reid, I. (2000). Articulated Body Motion Capture by Annealed Particle Filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000, Cat. No.PR00662)*, Band 2, Seiten 126–133.

[Doucet, 1998] Doucet, A. (1998). On Sequential Simulation-based Methods for Bayesian Filtering. *Signal Processing*, Seiten 1–26.

[Doucet et al., 2001] Doucet, A., De Freitas, N., und Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer, New York, NY.

[Doucet und Johansen, 2011] Doucet, A. und Johansen, A. M. (2011). A Tutorial on Particle Filtering and Smoothing: Fifteen Years later. *The Oxford Handbook of Nonlinear Filtering*, 12:656–704.

[Ebert und Henrich, 2001] Ebert, D. und Henrich, D. (2001). Safe Human-Robot-Cooperation: Problem Analysis, System Concept and Fast Sensor Fusion. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2001)*, Seiten 239 – 244.

[Ebert und Henrich, 2002] Ebert, D. und Henrich, D. (2002). Safe Human-Robot-Cooperation: Image-based Collision Detection for Industrial Robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Band 2, Seiten 1826–1831.

[Elhabian et al., 2008] Elhabian, S. Y., El-Sayed, K. M., und Ahmed, S. H. (2008). Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art. *Recent Patents on Computer Science*, Seiten 32–54.

[Fernando und NVIDIA Corporation, 2004] Fernando, R. und NVIDIA Corporation (2004). *GPU Gems: Programming Techniques, Tips, and Tricks for Real-time Graphics*. Addison-Wesley, Boston, MA, 5. Auflage.

[Finch, 2009] Finch, T. (2009). Incremental Calculation of Weighted Mean and Variance. *University of Cambridge*, 4(11-5):41–42.

[Fischer und Henrich, 2009] Fischer, M. und Henrich, D. (2009). Surveillance of Robots using Multiple Colour or Depth Cameras with Distributed Processing. In *Proceedings of the 3rd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009)*, Seiten 1–8.

[Foley et al., 1996] Foley, J. D., van Dan, A., Feiner, S. K., und Hughes, J. F. (1996). *Computer Graphics: Principles & Practice In C*. Pearson Education, New Jersey, 2. Auflage.

[Franco und Boyer, 2009] Franco, J.-S. und Boyer, E. (2009). Efficient Polyhedral Modeling from Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:414–27.

[Gecks, 2011] Gecks, T. (2011). *Sensorbasierte, echtzeitfähige Online-Bahnplanung für die Mensch-Roboter-Koexistenz*. Dissertation, Lehrstuhl für Angewandte Informatik III, Universität Bayreuth.

[Gordon et al., 1993] Gordon, N., Salmond, D., und Smith, A. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings F Radar and Signal Processing*, 140:107.

[Guan et al., 2007] Guan, L., Franco, J.-S., und Pollefeys, M. (2007). 3D Occlusion Inference from Silhouette Cues. In *Proceedings of Computer Vision and Pattern Recognition*, Seiten 1–8.

[Guan et al., 2008] Guan, L., Franco, J.-S., und Pollefeys, M. (2008). 3D Object Reconstruction with Heterogeneous Sensor Data. In *Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission*.

[Haenel, 2015] Haenel, M. (2015). *A Matter of Perspective - Three-dimensional Placement of Multiple Cameras to Maximize their Coverage*. Dissertation, Lehrstuhl für Angewandte Informatik III, Universität Bayreuth.

[Handschin und Mayne, 1969] Handschin, J. E. und Mayne, D. Q. (1969). Monte Carlo Techniques to Estimate the Conditional Expectation in Multi-stage Non-linear Filtering. *International Journal of Control*, 9:547–559.

[Hasler et al., 2020] Hasler, N., Richter, M., Dimitrov, S., Wilcken, F., Birster, I., und Theobalt, C. (2020). The Captury, Markerless Motion Capture Technology. https://captury.com/, zuletzt aufgerufen am: 04.12.2021.

[Henrich, 2021] Henrich, D. (2021). SIMERO-Projekt. https://www.ai3.uni-bayreuth.de/de/forschung/simero/index.php, zuletzt aufgerufen am: 04.12.2021.

[Henrich et al., 2008] Henrich, D., Fischer, M., Gecks, T., und Kuhn, S. (2008). Sichere Mensch/Roboter-Koexistenz und Kooperation. In *Proceedings of the ROBOTIK 2008*.

[Henriques et al., 2011] Henriques, J. F., Caseiro, R., und Batista, J. (2011). Globally Optimal Solution to Multi-Object Tracking with Merged Measurements. In *Proceedings of the IEEE International Conference on Computer Vision*, Seiten 2470–2477.

[Hofmann, 2011] Hofmann, M. (2011). Event Detection in a Smart Home Environment using Viterbi Filtering and Graph Cuts in a 3D Voxel Occupancy Grid. In *Proceedings of the International Conference on Computer Vision Theory and Application (VISAPP 2011)*, Seiten 242–247.

[Hol et al., 2006] Hol, J. D., Schön, T. B., und Gustafsson, F. (2006). On Resampling Algorithms for Particle Filters. In *Proceedings of the Nonlinear Statistical Signal Processing Workshop (NSSPW 2006)*.

[Huang et al., 2008] Huang, C., Wu, B., und Nevatia, R. (2008). Robust Object Tracking by Hierarchical Association of Detection Responses. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5303 LNCS(PART 2):788–801.

[Ikeuchi, 2014] Ikeuchi, K. (2014). *Computer Vision: A Reference Guide*. Springer Reference. Springer, New York.

[Isard und Blake, 1998] Isard, M. und Blake, A. (1998). Condensation – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29:5–28.

[Jähne, 2005] Jähne, B. (2005). *Digital Image Processing*. Springer Verlag, Berlin.

[Jorge et al., 2004] Jorge, P. M., Jorge, P. M., Marques, J. S., und Abrantes, A. J. (2004). On-line Tracking Groups of Pedestrians with Bayesian Networks. In *Proceedings of the Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2004)*, Seiten 65–72.

[Kanazawa et al., 1995] Kanazawa, K., Koller, D., und Russell, S. (1995). Stochastic Simulation Algorithms for Dynamic Probabilistic Networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*, Seiten 346–351.

[Keck und Davis, 2008] Keck, M. und Davis, J. W. (2008). 3D Occlusion Recovery using few Cameras. In *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*.

[Kehl et al., 2005] Kehl, R., Bray, M., und Van Gool, L. (2005). Full Body Tracking from Multiple Views using Stochastic Sampling. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, Band 2, Seiten 129–136.

[Kehl und Gool, 2006] Kehl, R. und Gool, L. V. (2006). Markerless Tracking of Complex Human Motions from Multiple Views. *Computer Vision and Image Understanding*, 104:190–209.

[Khan et al., 2003] Khan, Z., Balch, T., und Dellaert, F. (2003). Efficient Particle Filter-based Tracking of Multiple Interacting Targets using an MRF-based Motion Model. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Band 1, Seiten 254–259.

[Kim et al., 2004] Kim, K., Chalidabhongse, T., Harwood, D., und Davis, L. (2004). Background Modeling and Subtraction by Codebook Construction. In *Proceedings of the International Conference on Image Processing (ICIP 2004)*, Band 5, Seiten 3061–3064.

[Kitagawa, 1996] Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5:1–25.

[Koch et al., 2006] Koch, K., McLean, J., Segev, R., Freed, M. A., Berry, M. J., Balasubramanian, V., und Sterling, P. (2006). How much the Eye tells the Brain. *Current Biology*, 16(14):1428–1434.

[Kong et al., 1994] Kong, A., Liu, J. S., und Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, 89:278–288.

[Konstantinova et al., 2003] Konstantinova, P., Udvarev, A., und Semerdjiev, T. (2003). A Study of a Target Tracking Algorithm using Global Nearest Neighbor Approach. In *Proceedings of the 4th International Conference on Computer Systems and Technologies (CompSysTech 2003)*, Seiten 290–295. Association for Computing Machinery (ACM).

[Kruger und Westermann, 2003] Kruger, J. und Westermann, R. (2003). Acceleration Techniques for GPU-based Volume Rendering. *Proceedings of the 14th IEEE Visualization (VIS 2003)*, Seiten 287–292.

[Kuhn, 2012] Kuhn, S. (2012). *Wissens- und sensorbasierte geometrische Rekonstruktion.* Dissertation, Lehrstuhl für Angewandte Informatik III, Universität Bayreuth.

[Kuhn et al., 2006] Kuhn, S., Gecks, T., und Henrich, D. (2006). Velocity Control for Safe Robot Guidance based on Fused Vision and Force/Torque Data. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, Seiten 485–492.

[Kuhn und Henrich, 2009] Kuhn, S. und Henrich, D. (2009). Multi-View Reconstruction of Unknown Objects within a Known Environment. In *Proceedings of the Lecture Notes in Computer Science, including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics (LNCS 2009)*, Band 5875, Seiten 784–795. Springer.

[Kuhn und Henrich, 2010] Kuhn, S. und Henrich, D. (2010). Multi-View Reconstruction in-between Known Environments. Technischer report, https://epub.uni-bayreuth.de/442/, zuletzt aufgerufen am: 30.01.2022.

[Kutulakos und Seitz, 2000] Kutulakos, K. N. und Seitz, S. M. (2000). A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 38(3):199–218.

[Ladikos et al., 2008] Ladikos, A., Benhimane, S., und Navab, N. (2008). Efficient Visual Hull Computation for Real-time 3D Reconstruction using CUDA. In *Computer Vision and Pattern Recognition Workshops*, Seiten 1–8.

[Laurentini, 1991] Laurentini, A. (1991). The Visual Hull: a New Tool for Contour-based Image Understanding. In *Proceedings of the 7th Scandinavian Conference on Image Analysis*, Seiten 993–1002.

[Laurentini, 1994] Laurentini, A. (1994). The Visual Hull Concept for Silhouette-based Image Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:150–162.

[Laurentini, 1995] Laurentini, A. (1995). How far 3D Shapes can be Understood from 2D Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:188–195.

[Lazebnik et al., 2007] Lazebnik, S., Furukawa, Y., und Ponce, J. (2007). Projective Visual Hulls. *International Journal of Computer Vision*, 74:137–165.

[Li und Lee, 2019] Li, C. und Lee, G. H. (2019). Generating Multiple Hypotheses for 3D Human Pose Estimation With Mixture Density Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, Seiten 9887–9895.

[MacCormick, 2012] MacCormick, J. (2012). *Stochastic Algorithms for Visual Tracking: Probabilistic Modelling and Stochastic Algorithms for Visual Localisation and Tracking*. Springer Science & Business Media.

[MacCormick und Blake, 1999] MacCormick, J. und Blake, A. (1999). Probabilistic Exclusion Principle for Tracking Multiple Objects. In *Proceedings of the IEEE International Conference on Computer Vision*, Band 1, Seiten 572–578.

[Makris und Prieur, 2014] Makris, A. und Prieur, C. (2014). Bayesian Multiple-Hypothesis Tracking of Merging and Splitting Targets. *IEEE Transactions on Geoscience and Remote Sensing*, 52(12):7684–7694.

[Marrón et al., 2009] Marrón, M., Pizarro, D., García, J. C., Marcos, A., Jalvo, R., und Mazo, M. (2009). Multi-Agent 3D Tracking in Intelligent Spaces with a Single Extended Particle Filter. In *Proceedings of the 6th IEEE International Symposium on Intelligent Signal Processing*, Seiten 305–310.

[Meissner et al., 1999] Meissner, M., Hoffmann, U., und Strasser, W. (1999). Enabling Classification and Shading for 3D Texture Mapping based Volume Rendering using OpenGL and Extensions. In *Proceedings of the IEEE Visualization 1999*, Seiten 207–214.

[Metropolis und Ulam, 1949] Metropolis, N. und Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association*, 44:335–341.

[Mikić et al., 2003] Mikić, I., Trivedi, M., Hunter, E., und Cosman, P. (2003). Human Body Model Acquisition and Tracking using Voxel Data. *International Journal of Computer Vision*, 53:199–223.

[Moeslund et al., 2006] Moeslund, T. B., Hilton, A., und Krüger, V. (2006). A Survey of Advances in Vision-based Human Motion Capture and Analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126.

[Moschini und Fusiello, 2009] Moschini, D. und Fusiello, A. (2009). Tracking Human Motion with Multiple Cameras Using an Articulated Model. In *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics Collaboration-Techniques (MIRAGE 2009)*, Seiten 1–12.

[Munder, 2013] Munder, M. (2013). *Markerless Voxel-based Motion Capturing*. Seminararbeit, Angewandte Informatik III, Universität Bayreuth.

[Munder, 2015] Munder, M. (2015). *Entwicklung und Analyse eines Initialisierungsverfahrens zur Körperposenschätzung basierend auf einem Mehrkamerasystem in teilweise verdeckten Umgebungen*. Masterarbeit, Angewandte Informatik III, Universität Bayreuth.

[Nitschke, 2006] Nitschke, C. (2006). *A Framework for Real-time 3D Reconstruction by Space Carving using Graphics Hardware*. Diplomarbeit, Bauhaus-Universität Weimar.

[Ober, 2007] Ober, A. (2007). *Analyse von Bewegungstrajektorien zur nutzerangepassten Dialoginitiierung*. Diplomarbeit, Fachgebiet Neuroinformatik und Kognitive Robotik, Technische Universität Ilmenau.

[Ober und Henrich, 2010] Ober, A. und Henrich, D. (2010). A Safe Fault Tolerant Multi-View Approach for Vision-Based Protective Devices. *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2010)*, Seiten 17–25.

[Ober-Gecks et al., 2014a] Ober-Gecks, A., Hänel, M., Henrich, D., und Werner, T. (2014a). Fast Multi-Camera Reconstruction and Surveillance with Human Tracking and Optimized Camera Configurations. In *Proceedings of the 45th International Symposium on Robotics and 8th German Conference on Robotics (ISR/Robotik 2014)*.

[Ober-Gecks et al., 2014b] Ober-Gecks, A., Zwicker, M., und Henrich, D. (2014b). Efficient GPU Photo Hull Reconstruction for Surveillance. In *Proceedings of the International Conference on Distributed Smart Cameras (ICDSC 2014)*, Seiten 21:1–21:8. ACM.

[Ober-Gecks et al., 2016] Ober-Gecks, A., Zwicker, M., und Henrich, D. (2016). Efficient Graphics Processing Unit–based Voxel Carving for Surveillance. *Journal of Electronic Imaging*, 25(4):041011.

[Oh et al., 2009] Oh, S., Russell, S., und Sastry, S. (2009). Markov Chain Monte Carlo Data Association for Multi-Target Tracking. *IEEE Transactions on Automatic Control*, 54:481–497.

[Ohsawa et al., 2013] Ohsawa, Y., Yamaguchi, K., Ichikawa, T., und Sakamoto, Y. (2013). Computer-generated Holograms using Multiview Images captured by a small Number of Sparsely Arranged Cameras. *Applied Optics*, 52(1):A167–A176.

[Owens et al., 2007] Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E., und Purcell, T. J. (2007). A Survey of General-Purpose Computation on Graphics Hardware. *Computer Graphics Forum*, 26(1):80–113.

[Padeleris et al., 2013] Padeleris, P., Zabulis, X., und Argyros, A. A. (2013). Multicamera Tracking of Multiple Humans based on Colored Visual Hulls. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2013)*, Seiten 1–8.

[Pavlenko, 2012] Pavlenko, A. (2012). Open Source Computer Vision Library (Open CV) 2.4. https://github.com/opencv/opencv/releases/tag/2.4.9, zuletzt aufgerufen am: 04.12.2021.

[Perera et al., 2006] Perera, A. G., Srinivas, C., Hoogs, A., Brooksby, G., und Hu, W. (2006). Multi-Object Tracking through Simultaneous Long Occlusions and Split-Merge Conditions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Band 1, Seiten 666–673.

[Perry, 2014] Perry, T. S. (2014). Motion Capture Technology Goes Into the Wild for Dawn of the Planet of the Apes. https://spectrum.ieee.org/motion-capture-technology-goes-into-the-wild-for-dawn-of-the-planet-of-the-apes, zuletzt aufgerufen am: 04.12.2021.

[Poppe, 2007] Poppe, R. (2007). Vision-based Human Motion Analysis: An Overview. *Computer Vision and Image Understanding*, 108:4–18.

[Prock und Dyer, 1998] Prock, A. und Dyer, C. (1998). Towards Real-time Voxel Coloring. In *Proceedings of the DARPA Image Understanding Workshop*, Seiten 315–321.

[Radke et al., 2005] Radke, R. J., Andra, S., Al-Kofahi, O., und Roysam, B. (2005). Image Change Detection Algorithms: A Systematic Survey. *IEEE Transactions on Image Processing*, 14(3):294–307.

[Reid, 1979] Reid, D. B. (1979). An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854.

[Ristic et al., 2004] Ristic, B., Arulampalam, S., und Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, MA.

[Rubin, 1987] Rubin, D. B. (1987). A Noniterative Sampling/Importance Resampling Alternative to the Data Augmentation Algorithm for Creating a Few Imputations when Fractions of Missing Information are Modest: The SIR Algorithm. *Journal of the American Statistical Association*, 82:543–546.

[Sainz et al., 2002] Sainz, M., Bagherzadeh, N., und Susin, A. (2002). Hardware Accelerated Voxel Carving. In *Proceedings of the 1st Ibero-American Symposium in Computer Graphics (SIACG 2002)*, Seiten 289–297.

[Salvador und Casas, 2008] Salvador, J. und Casas, J. R. (2008). Shape from Probability Maps with Image-Adapted Voxelization. In *Proceedings of the ECCV Workshop on Multi-Camera and Multi-Modal Sensor Fusion Algorithms and Applications (M2SFA2)*, Seiten 1–12.

[Sand, 2019] Sand, M. (2019). *Inkrementelle Rekonstruktion von planaren Volumenmodellen mit handgehaltenen Tiefenkameras*. Dissertation, Bayreuth.

[Sarafianos et al., 2016] Sarafianos, N., Boteanu, B., Ionescu, B., und Kakadiaris, I. A. (2016). 3D Human Pose Estimation: A Review of the Literature and Analysis of Covariates. *Computer Vision and Image Understanding*, 152:1–20.

[Scharstein, 2006] Scharstein, D. (2006). Middlebury Dataset, Temple and Stegosaurus. http://vision.middlebury.edu/mview/, zuletzt aufgerufen am: 04.12.2021.

[Schick und Stiefelhagen, 2009] Schick, A. und Stiefelhagen, R. (2009). Real-time GPU-based Voxel Carving with Systematic Occlusion Handling. In *Proceedings of the 31st Symposium of the German Association for Pattern Recognition (DAGM)*, Seiten 372–381.

[Segal und Akeley, 2013] Segal, M. und Akeley, K. (2013). The OpenGL Graphics System: A Specification, Version 4.4, Core Profile.

[Seitz und Dyer, 1999] Seitz, S. M. und Dyer, C. R. (1999). Photorealistic Scene Reconstruction by Voxel Coloring. *International Journal of Computer Vision*, 35(2):151–173.

[Singh et al., 2008] Singh, V. K., Wu, B., und Nevatia, R. (2008). Pedestrian Tracking by Associating Tracklets using Detection Residuals. In *Proceedings of the IEEE Workshop on Motion and Video Computing (WMVC 2008)*, Seiten 1–8.

[Slabaugh et al., 2001] Slabaugh, G. G., Culbertson, W. B., Malzbender, T., und Schafer, R. W. (2001). A Survey of Methods for Volumetric Scene Reconstruction from Photographs. In *Proceedings of the 2001 Eurographics Conference on Volume Graphics (VG 2001)*, Seiten 81–101.

[Slabaugh et al., 2004] Slabaugh, G. G., Culbertson, W. B., Malzbender, T., Stevens, M. R., und Schafer, R. W. (2004). Methods for Volumetric Reconstruction of Visual Scenes. *International Journal of Computer Vision*, 57(3):179–199.

[Sminchisescu und Triggs, 2003] Sminchisescu, C. und Triggs, B. (2003). Estimating Articulated Human Motion with Covariance Scaled Sampling. *The International Journal of Robotics Research*, 22:371–391.

[Steinbach et al., 2000] Steinbach, E., Girod, B., Eisert, P., und Betz, A. (2000). 3-D Reconstruction of Real-world Objects using Extended Voxels. In *Proceedings of the International Conference on Image Processing (ICIP 2000, Cat. No.00CH37101)*, Band 1, Seiten 138–141.

[Stoychev, 2013] Stoychev, V. (2013). *Fotorealistische Nachbildung einer Roboterzelle*. Masterprojekt, Lehrstuhl für Angewandte Informatik III, Universität Bayreuth.

[Svoboda, 2011] Svoboda, T. (2011). Multi-Camera Self-Calibration. https://cmp.felk.cvut.cz/~svoboda/SelfCal/index.html, zuletzt aufgerufen am: 04.12.2021.

[Svoboda et al., 2005] Svoboda, T., Martinec, D., und Pajdla, T. (2005). A Convenient Multicamera Self-Calibration for Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 14:407–422.

[Szeliski, 1993] Szeliski, R. (1993). Rapid Octree Construction from Image Sequences. *CVGIP: Image Understanding*, 58(1):23–32.

[Thrun et al., 2005] Thrun, S., Burgard, W., und Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents Series)*. The MIT Press, Cambridge, MA, USA.

[Tran und Trivedi, 2008] Tran, C. und Trivedi, M. (2008). Human Body Modelling and Tracking using Volumetric Representation: Selected Recent Studies and Possibilities for Extensions. In *Proceedings of the Second ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2008)*, Seiten 1–9.

[Wang et al., 2021] Wang, J., Tan, S., Zhen, X., Xu, S., Zheng, F., He, Z., und Shao, L. (2021). Deep 3D Human Pose Estimation: A Review. *Computer Vision and Image Understanding*, 210:103225.

[Werner et al., 2017] Werner, T., Bloeß, J., und Henrich, D. (2017). Neural Networks for Real-Time, Probabilistic Obstacle Detection. In *Proceedings of the 26th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD 2017)*.

[Werner et al., 2019] Werner, T., Harrer, D., und Henrich, D. (2019). Efficient, Risk-Encoding Octrees for Path Planning with a Robot Manipulator. In *Advances in Service and Industrial Robotics – Proceedings of the 28th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD 2019)*, Band 980 in *Advances in Intelligent Systems and Computing*, Seiten 455–462. Springer.

[Werner und Henrich, 2014] Werner, T. und Henrich, D. (2014). Efficient and Precise Multi-Camera Reconstruction. In *Proceedings of the 8th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2014)*, Seiten 23:1–23:6.

[Werner et al., 2018] Werner, T., Henrich, D., und Sand, M. (2018). Sparse and Precise Reconstruction of Static Obstacles for Real-Time Path Planning in Human-Robot Workspaces. In *Proceedings of the 50th International Symposium on Robotics (ISR)*.

[Wikipedia, 2020] Wikipedia (2020). Films using Motion Capture by taking Actors in Specialized Suits and adding Digital Animation to their Performances. https://en.wikipedia.org/wiki/Category:Films_using_motion_capture, zuletzt aufgerufen am: 04.12.2021.

[Wikipedia, 2021] Wikipedia (2021). Lochkamera (Funktionsweise, Geometrische Abbildungseigenschaften usw.). https://de.wikipedia.org/wiki/Lochkamera, zuletzt aufgerufen am: 04.12.2021.

[Yilmaz et al., 2006] Yilmaz, A., Javed, O., und Shah, M. (2006). Object Tracking: A Survey. *ACM Computing Surveys*, 38:13.

[Zaritskii et al., 1975] Zaritskii, V. S., Svetnik, V. B., und Simelevic, L. I. (1975). Monte-Carlo Technique in Problems of Optimal Information Processing. *Automation and Remote Control*, 36(12):2015–2022.

[Zhang et al., 2006] Zhang, J., Luo, J., Collins, R., und Liu, Y. (2006). Body Localization in Still Images using Hierarchical Models and Hybrid Search. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, Band 2, Seiten 1536–1543.

[Zhang, 2012] Zhang, Z. (2012). Microsoft Kinect Sensor and Its Effect. *IEEE Multimedia (IEEEMM)*, 19(2):4–10.

[Zheng et al., 2020] Zheng, C., Wu, W., Yang, T., Zhu, S., Chen, C., Liu, R., Shen, J., Kehtarnavaz, N., und Shah, M. (2020). Deep Learning-Based Human Pose Estimation: A Survey. *CoRR*, abs/2012.1.

[Zwicker, 2013] Zwicker, M. (2013). *Erweiterung der Photohülle zur schnellen Onlinerekonstruktion auf moderner Grafikhardware*. Masterarbeit, Lehrstuhl für Angewandte Informatik III, Universität Bayreuth.

Eigene Publikationen

[Müller et al., 2008] Müller, S., Hellbach, S., Schaffernicht, E., Ober, A., Scheidig, A., und Gross, H.-M. (2008). Whom to Talk to? Estimating User Interest from Movement Trajectories. In *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2008)*, Seiten 532–538.

[Müller et al., 2007] Müller, S., Scheidig, A., Ober, A., und Gross, H.-M. (2007). Making Mobile Robots Smarter by Probabilistic User Modeling and Tracking. In *Proceedings of the 52th International Scientific Colloquium (IWK 2007)*, Seiten 451–456.

[Ober, 2007] Ober, A. (2007). *Analyse von Bewegungstrajektorien zur nutzerangepassten Dialoginitiierung*. Diplomarbeit, Fachgebiet Neuroinformatik und Kognitive Robotik, Technische Universität Ilmenau.

[Ober und Henrich, 2010] Ober, A. und Henrich, D. (2010). A Safe Fault Tolerant Multi-View Approach for Vision-Based Protective Devices. In *Proceedings of the 7th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2010)*, Seiten 17–25.

[Ober-Gecks et al., 2014a] Ober-Gecks, A., Hänel, M., Henrich, D., und Werner, T. (2014a). Fast Multi-Camera Reconstruction and Surveillance with Human Tracking and Optimized Camera Configurations. In *Proceedings of the 45th International Symposium on Robotics and the 8th German Conference on Robotics (ISR/Robotik 2014)*.

[Ober-Gecks et al., 2014b] Ober-Gecks, A., Zwicker, M., und Henrich, D. (2014b). Efficient GPU Photo Hull Reconstruction for Surveillance. In *Proceedings of the International Conference on Distributed Smart Cameras (ICDSC 2014)*, Seiten 21:1–21:8.

[Ober-Gecks et al., 2016] Ober-Gecks, A., Zwicker, M., und Henrich, D. (2016). Efficient Graphics Processing Unit–based Voxel Carving for Surveillance. *Journal of Electronic Imaging*, 25(4):041011.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin erkläre ich, dass ich die Hilfe von gewerblichen Promotionsberatern bzw. -vermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe, noch künftig in Anspruch nehmen werde.

Zusätzlich erkläre ich hiermit, dass ich keinerlei frühere Promotionsversuche unternommen habe.

Bayreuth, den

Unterschrift