

Wissens- und sensorbasierte geometrische Rekonstruktion

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

von

Stefan Kuhn

aus Birkenfeld

1. Gutachter: Prof. Dr. Dominik Henrich,
Universität Bayreuth
2. Gutachter: Prof. Dr.-Ing. Friedrich M. Wahl,
Technische Universität Braunschweig

Tag der Einreichung: 25.07.2012

Tag des Kolloquiums: 6.12.2012

Für die spannende, interessante und lehrreiche Zeit,
die nette und hilfreiche Anleitung,
die vielen abwechslungsreichen Möglichkeiten und Momente,
die angenehme und motivierende Atmosphäre,
die intensiven fachlichen und nichtfachlichen Gespräche und
den täglichen Ausgleich
möchte ich mich ganz herzlich bei allen bedanken,
die dazu beigetragen haben:

Meinem Doktorvater Prof. Dr. Dominik Henrich,
meinem Zweitgutachter Prof. Dr. Friedrich M. Wahl,
meinen Kollegen,
meiner Familie und
meinen Eltern.

Zusammenfassung

Möchte man herausfinden, wo sich bestimmte Objekte innerhalb eines gegebenen Raumes aufhalten könnten, so bleibt einem zunächst nur die Antwort *“Irgendwo im gegebenen Raum”*. Erst mit Hilfe zusätzlicher Informationen, wie beispielsweise Sensordaten oder Eigenschaften der bestimmten Objekte oder der Umwelt im gegebenen Raum, lassen sich die möglichen Aufenthaltsorte einschränken, indem man solche Bereiche des Raumes ausschließt, in denen sich keines der bestimmten Objekte aufhalten kann.

Sind beispielsweise Menschen die bestimmten Objekte und deren mögliche Aufenthaltsorte innerhalb einer Roboter-Arbeitszelle von Interesse, dann muss man ohne weitere Informationen in der gesamten Roboter-Arbeitszelle Menschen vermuten. Unter der Voraussetzung, dass sich ein Mensch nicht in soliden Objekten der Umwelt aufhalten kann, reduzieren sich die möglichen Aufenthaltsorte innerhalb des gegebenen Raumes. Sensoren können verwendet werden, um freie Bereiche innerhalb der Roboter-Arbeitszelle zu erfassen, um damit die möglichen Aufenthaltsorte weiter einzuzugrenzen. Ein anderer Aspekt könnte das Minimalvolumen der bestimmten Objekte berücksichtigen, um so Regionen bei Unterschreitung dieses Minimalvolumens zu verwerfen, in denen zuvor Menschen vermutet werden mussten. Weitere Aspekte zur Eingrenzung möglicher Aufenthaltsorte stellen beispielsweise die Berücksichtigung von Farben, Distanzen, Geschwindigkeiten, Gewichten etc. dar.

Ziel dieser Arbeit ist die automatisierte, computerbasierte Lösung des obigen Problems, nämlich die Bestimmung und geometrische Beschreibung möglicher Aufenthaltsorte bestimmter Objekte innerhalb eines gegebenen Raumes unter Nutzung von Wissen und Sensoren. Es wird dabei gefordert, dass die geometrische Beschreibung – im Folgenden als geometrische Rekonstruktion bezeichnet – konservativ sein muss, d.h. die bestimmten Objekte innerhalb des gegebenen Raumes dürfen nicht aus der Rekonstruktion herausragen.

Das Problem wird zunächst allgemein im n -dimensionalen euklidischen Raum modelliert. Dieses Modell kann als Rahmenwerk angesehen werden, welches die konsistente Integration von Wissen und Sensoren erlaubt, um eine geometrische Rekonstruktion zu bestimmen. Unterschiedliches Wissen und unterschiedliche Sensoren werden exemplarisch integriert und diskutiert.

Basierend auf dem allgemein eingeführten Modell wird eine Implementierung für einen dreidimensionalen Voxelaum abgeleitet. Besondere Aufmerksamkeit ist bei der Verwendung diskreter Datenstrukturen erforderlich, um die Konservativität der resultierenden geometrischen Rekon-

struktion zu gewährleisten.

Ein Prototyp wurde versuchsweise im industriellen Umfeld in einem Mensch/Roboter-Koexistenz-Szenario in Zusammenarbeit mit einem deutschen Automobilhersteller eingesetzt. Das Robotersystem nutzt die berechneten geometrischen Rekonstruktionen, um die Geschwindigkeit des Roboterarms bei Annäherung an einen Menschen zu reduzieren.

Ein weiteres Experiment diente der quantitativen Untersuchung der resultierenden geometrischen Rekonstruktionen in einem vergleichbaren Aufbau. Die verbleibende Anzahl an Voxeln der geometrischen Rekonstruktionen beläuft sich im Durchschnitt über alle Zeitpunkte der im Experiment betrachteten Sequenz auf etwa 1,22% bezüglich aller Voxel innerhalb des gegebenen Raumes. Im direkten Vergleich dazu verbleiben bei einer einfachen Multi-Kamera-Rekonstruktion, welche "Occlusion Masks" zur Behandlung von sichtverdeckenden Hindernissen nutzt, durchschnittlich etwa 9,62% der Gesamtanzahl an Voxeln innerhalb des gegebenen Raumes. Die wissens- und sensorbasierte geometrische Rekonstruktion besteht also durchschnittlich aus etwa 12,7% der Voxel gegenüber dem einfachen Ansatz, welcher "Occlusion Masks" nutzt und beschreibt damit die bestimmten Objekte wesentlich exakter.

Abstract

If we want to know where given objects could be located inside a given space, the answer must be *"anywhere within the given space"*. But if we have additional information like sensory data or properties of these objects or the given space, we can delimit the possible locations by excluding parts of the space which cannot contain these objects.

For example, consider humans to be the given objects inside a robot work cell which represents the given space. If we do not have any information, we must assume the humans to be located anywhere inside this robot work cell. Provided that a human cannot reside inside a solid object of the environment, possible locations of humans are reduced in the given space. Sensors could be used to detect free parts of the robot work cell which also delimit the possible locations of the given objects. Another piece of information might be the minimum volume of the given objects so that regions which formerly had to be assumed containing given objects can be rejected due to a too low volume. Information about colors, distances, speeds, weights et cetera can be utilized, too.

This work aims the automatic, computer based solution of the described problem namely the determination and geometric description of possible locations of given objects within a

given space, based on knowledge and sensory data. The geometric description – referred to as geometric reconstruction – is required to be conservative which means that the given objects must be completely covered by it within the given space.

The problem is generally modelled for the n -dimensional Euclidean space. This model can also be seen as framework which allows the consistently integration of knowledge and sensors to determine a geometric reconstruction. Different knowledge and sensors are exemplary integrated and discussed.

Based on the introduced general model an implementation using a three-dimensional voxel space is derived. Special attention must be paid when using discrete data structures in order to guarantee that the resulting geometric reconstruction actually is conservative.

A prototype was experimentally applied to an industrial human/robot coexistence setting in collaboration with a german automotive manufacturer. The robot system uses the calculated geometric reconstructions in order to reduce the motion speed of the robot arm when approaching a human.

Another experiment in a similar setting was used to quantitatively analyze the resulting geometric reconstructions. The average resulting number of voxels of all geometric reconstructions within the analyzed sequence of the experiment is about 1,22% in relation to the number of all voxels within the given space. In comparison, the average number of voxels resulting from a simple multi view reconstruction approach which uses “occlusion masks” to cope with occluders within the scene is about 9,62% in relation to the number of all voxels within the given space. Hence, the knowledge and sensor based geometric reconstruction contains about 12,7% of voxels in relation to the simple approach which uses “occlusion masks” and thus describes the given objects much more exactly.

Inhaltsverzeichnis

1. Einleitung	13
1.1. Motivation	13
1.2. Ziele	16
1.3. Aufgabenstellung	18
1.4. Abgrenzung	18
1.5. Kapitelübersicht	19
2. Stand der Forschung	21
2.1. Historischer Überblick	22
2.2. Forschungsprojekte	23
2.3. Ansätze zur Mensch/Roboter-Koexistenz und -Kooperation	25
2.4. Ansätze im Bereich der Multi-Kamera-Rekonstruktion	33
2.4.1. Merkmale und Einordnung von Multi-Kamera-Rekonstruktionsmethoden	33
2.4.2. Bewertung	35
2.5. Schlussfolgerungen	37
3. Modellierung	39
3.1. Geometrische Rekonstruktion und Plausibilisierung	39
3.2. Plausibilisierungsfunktionen	50
3.2.1. Ausschließbarer Raumbereich	50
3.2.2. Sensorbasierter ausschließbarer Raumbereich	51
3.2.3. Erfassbarkeit	63
3.2.4. Form bezüglich Raumbereich	70
3.2.5. Volumen	72
3.2.6. Distanz	76

3.2.7. Rand	80
3.2.8. Zeit	82
3.2.9. Zusammenfassung	84
3.3. Anwendungsbeispiel: Mensch/Roboter-Koexistenz	84
3.4. Zusammenfassung und Schlussfolgerungen	90
4. Umsetzung	91
4.1. Voxel: Grundlagen und Operationen	91
4.2. Voxelbasierte geometrische Rekonstruktion und Plausibilisierung	99
4.3. Voxelbasierte Plausibilisierungsfunktionen	103
4.3.1. Ausschließbarer Raumbereich	103
4.3.2. Sensorbasierter ausschließbarer Raumbereich	106
4.3.3. Erfassbarkeit	114
4.3.4. Form bezüglich Raumbereich	125
4.3.5. Volumen	127
4.3.6. Distanz	132
4.3.7. Rand	135
4.3.8. Zeit	137
4.4. Experiment	139
4.4.1. Aufbau, Wissen und Sensoren	139
4.4.2. Auswertung	143
4.4.3. Vergleich mit der Methode der Occlusion Masks	153
4.4.4. Zusammenfassung und Schlussfolgerungen	153
4.5. Zusammenfassung und Schlussfolgerungen	156
5. Anwendung	159
5.1. Aufgabenstellung	159
5.2. Umsetzung	161
5.3. Plausibilisierungen	165
5.4. Ergebnisse	168
5.5. Schlussfolgerungen	169
6. Schlussfolgerungen	171

7. Ausblick	173
7.1. Intuitive Umweltmodellierung	173
7.2. Optimale Sensorplatzierung, -kalibrierung und -registrierung	175
7.3. Automatisierte Parameterbestimmung	176
7.4. Evaluation weiterer Sensortypen	176
7.5. Sicherheit	177
7.6. Ausführungsgeschwindigkeit und Genauigkeit	177
7.7. Tracking und Gestenerkennung	179
7.8. Bahnplanung und Mensch/Roboter-Kooperation	180
Literaturverzeichnis	181
A. Dreiecks-, Kugel- und Voxelmodelle	195
A.1. Dreiecksmodelle	197
A.1.1. Erzeugung einer vergrößerten bzw. verkleinerten Voxelapproximation	199
A.2. Kugelmodelle	202
A.2.1. Erzeugung einer vergrößerten Voxelapproximation	204
A.2.2. Bemerkung: Verkleinerte Voxelapproximation	204
A.3. Erzeugung von Kugelmodellen	206
A.3.1. Algorithmus	206
A.3.2. Experimente	208
A.3.3. Schlussfolgerungen und Ausblick	218
A.4. Expansion und Kontraktion von Voxelmodellen	218
A.5. Beispiel: Schnelle verkleinerte Voxelapproximation	222
A.6. Schlussfolgerungen	224
B. Symbole und Bezeichnungen	225

1. Einleitung

1.1. Motivation

Man stelle sich ein Wohnzimmer mit einigen Schränken, Tischen, Stühlen, einer Couchgarnitur und dem sonstigen üblichen Inventar vor. Man selbst stehe in der Tür und schaue in diesen Raum hinein, ohne die Perspektive, also die eigene Position, zu verändern. Nun wird die Frage gestellt, wo sich zum aktuellen Zeitpunkt überall in diesem Raum Menschen aufhalten. Schnell kommt man zu dem Ergebnis, dass natürlich ein Mensch am Tisch sitzt, man allerdings *nicht* ausschließen kann, dass sich ein Mensch in den Schränken oder hinter der Couch versteckt und insofern auch dort Menschen sein könnten. Wenn man nun zusätzlich wüsste, dass zu einem früheren Zeitpunkt sich *kein* Mensch im Schrank versteckt hatte und man außerdem wüsste, dass der Schrank in der Zwischenzeit *nicht* geöffnet wurde, so könnte man daraus folgern, dass auch nun kein Mensch in dem Schrank versteckt sein kann. Wird jedoch die Frage gestellt, wo sich überall Kugelschreiber befinden, so ist die Liste der möglichen Aufenthaltsorte viel länger: Es könnten sehr viele in der Tischschublade liegen, hinter, unter, auf, neben und im Schrank, hinter dem Tisch auf dem Boden und nahezu überall dort, wo man aus der eingenommenen Perspektive nicht hinsehen kann.

Analysiert man den Prozess zur Ermittlung der Raumbereiche, wo sich gesuchte Objekte befinden könnten, so stellt man fest, dass man hier nach einem Ausschlussverfahren vorgegangen ist: Überall dort wo man keinen Menschen erblicken konnte, kann sich auch keiner befinden. Raumbereiche, die nicht eingesehen werden können, werden basierend auf der eigenen Wissensgrundlage ausgeschlossen – das Innere der Tischschublade ist zu klein, als dass dort ein Mensch hineinpassen würde; das Tischbein ist zu schmal, als dass sich dahinter ein Mensch verstecken könnte, ohne sichtbar an den Seiten herauszuragen; im nicht gesehenen Teil an der Zimmerdecke wird sich kein Mensch befinden, sofern man davon ausgehen kann, dass in diesem betrachteten Wohnzimmer Menschen nicht an Decken klettern, schweben oder so hoch hüpfen können. War

1. Einleitung

zuvor kein Mensch im Schrank und wurde die Schranktür in der Zwischenzeit nicht geöffnet, so kann auch nun kein Mensch darin sein. Übrig bleiben also diejenigen Raumbereiche, in denen sich gesuchte Objekte aufhalten könnten. Ergänzend sei zu bemerken, dass die gleichzeitige Einnahme unterschiedlicher Perspektiven auf den betrachteten Raum den Ausschluß größerer Bereiche ermöglicht und damit zu einem besseren bzw. genaueren Ergebnis führt – vorausgesetzt, dass diese Perspektiven geeignet gewählt sind. Realisieren ließe sich dies im Beispiel mittels mehrerer Überwachungskameras, deren Bilder dem Betrachter gleichzeitig an einem Monitor präsentiert werden.

Die wissens- und sensorbasierte geometrische Rekonstruktion ist nichts anderes als genau dieses “Versteckspiel”, jedoch in automatisierter Form: Sie nutzt zur Verfügung gestelltes Wissen und Sensorinformation, um per Ausschlussverfahren all diejenigen Raumbereiche automatisch zu ermitteln, in denen sich gesuchte Objekte befinden können. Die Platzierung eines Sensors stellt dabei die oben erwähnte Perspektive auf den zu betrachtenden Raum dar. Die aus dieser Perspektive gelieferte Sensorinformation wird allerdings automatisiert und nicht mehr durch einen menschlichen Betrachter ausgewertet. Die vorliegende Arbeit gibt Antworten auf die resultierenden Fragestellungen nach der Art der Repräsentation des Wissens und der Sensordaten, wie diese verknüpft werden und wie all das in einer Implementierung konkret umgesetzt wird und was dabei zu beachten ist.

Die für diese Arbeit ausschlaggebende und praktische Anwendung ist die *Mensch/Roboter-Koexistenz bzw. -Kooperation*: Damit Mensch und Roboter gleichzeitig im gemeinsamen Arbeitsraum arbeiten können, muss das Robotersystem in die Lage versetzt werden, seine Umwelt zu erfassen und geeignet darauf zu reagieren, d.h. die Raumbereiche ermitteln zu können, in denen sich Menschen aufhalten könnten, um dann geeignet auf diese – beispielsweise durch eine abstandsabhängige Geschwindigkeitsanpassung oder durch Ausweichen bzw. einer Neuplanung der Bahn – zu reagieren [Henr 06] (Abbildung 1.1).

Die Mensch/Roboter-Koexistenz und -Kooperation ist ein derzeit vieldiskutiertes aber auch relativ neues Thema. Seitens der Industrie wurde erkannt, dass dadurch einerseits Zeit, Platz und Kosten gegenüber einer Vollautomatisierung eingespart und andererseits ergonomischere Arbeitsplätze für den Menschen geschaffen werden können [Brec 05, Bart 11, Frau 11, Bort 11, Mura 00]. Normen zum Einsatz von Kamerasystemen anstelle der bisher üblichen und verordneten Sicherheitszäune wurden und werden derzeit durch die Berufsgenossenschaft (BG) und das Berufsgenossenschaftliche Institut für Arbeitsschutz (BGIA) ebenso wie die Belastbarkeit

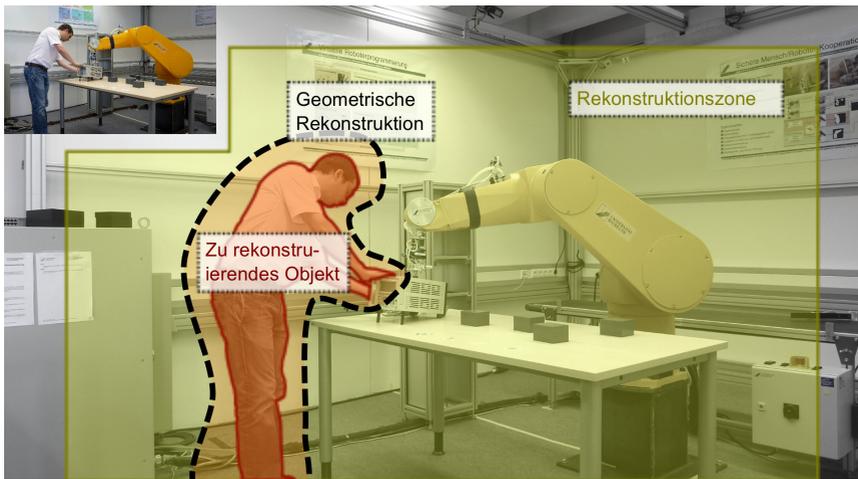


Abbildung 1.1.: Illustration der geometrischen Rekonstruktion am Anwendungsbeispiel der Mensch/Roboter-Koexistenz und -Kooperation. Das gesuchte Objekt ist der Mensch (rot). Die geometrische Rekonstruktion (orange) ist Teil der Rekonstruktionszone (gelb) und enthält den Mensch.

1. Einleitung

des menschlichen Körpers im Hinblick auf einwirkende Kräfte, Impulse und Drücke erarbeitet [Deut 09]. Ansätze, wie beispielsweise nachgiebige, schwache Roboter, einhergehend mit Beschränkungen von Bewegungsgeschwindigkeiten und Traglasten [Albu 07, Lens 10] ermöglichen schon heute eine Zusammenarbeit zwischen Mensch und Roboter, stehen jedoch orthogonal zu den Betrachtungen dieser Arbeit und werden daher nicht weiter erörtert.

Eine andere Anwendungsdomäne, in der ein solches System von großem Nutzen sein kann, sind die sogenannten *Smart Homes*. Insbesondere für ältere, alleinlebende Menschen kann es lebensrettend sein, wenn automatisch eine Hilfsituation, beispielsweise die eines Sturzes erkannt und einer betreuenden Sozialstation gemeldet wird [Foro 08]. Liegt eine Rekonstruktion des Menschen vor, so kann analysiert werden, ob er auf dem Boden liegt [Hofm 11].

1.2. Ziele

Bei der wissens- und sensorbasierten geometrischen Rekonstruktion geht es darum, dass Aufenthaltsorte bestimmter gesuchter Objekte innerhalb eines überwachten Raumbereiches mithilfe von Wissen und Sensoren ermittelt und anschließend geometrisch beschrieben werden können. Aus der Motivation lassen sich diesbezüglich sechs zentrale Ziele ableiten, wobei die ersten vier Ziele konzeptueller Natur und die beiden letzten eher technischer Natur sind:

Überwachung eines beliebigen Raumbereiches Der Raumbereich, in welchem man gesuchte Objekte betrachten möchte, sollte völlig unabhängig von irgendwelchem Wissen und von Sensoren gewählt werden können. Interessiert einen beispielsweise ein Raumbereich, der zwei halbe, durch eine Wand getrennte Roboterzellen zylinderförmig abdeckt, so sollte dies ohne Weiteres machbar sein. Ein Gegenbeispiel hierfür wäre demnach, dass der überwachte Raumbereich zwangsläufig durch die Schnittmenge mehrerer Kamera-Sichtbereiche festgelegt ist.

Integration beliebigen Wissens Kennt man Eigenschaften des zu überwachenden Raumbereiches und der innerhalb des Raumbereiches gesuchten Objekte, so sollte dieses Wissen – sofern es sich zum Ausschluss von freien Teilbereichen eignet – in einer konsistenten Weise integrierbar sein. Kennt man beispielsweise das Mindestvolumen gesuchter Objekte im überwachten Raumbereich, so sollte sich dieses Wissen in der Form integrieren lassen, als Teilbereiche mit kleinerem Volumen verworfen werden, da sich hierin ohnehin kein

gesuchtes Objekt aufhalten kann. Kennt man die Geometrien der soliden Gegenstände im überwachten Raumbereich und weiß man, dass sich dort nicht gleichzeitig ein gesuchtes Objekt aufhalten kann, so kann man dieses Wissen nutzen, um jene Teilbereiche zu verwerfen.

Integration beliebiger Sensoren Jede Art von Sensor, mit dessen Hilfe es möglich ist Teilbereiche des überwachten Raumbereiches auszuschließen, weil dieser beispielsweise dort die Abwesenheit gesuchter Objekte detektiert, sollte in konsistenter Weise integrierbar sein. Neben Farb-, Infrarot- und Tiefenkameras sind ebenso Näherungssensoren, Türschalter, Zugangslichtschranken und Kontakt-Fußmatten hierfür denkbar. Ein System zur wissens- und sensorbasierten geometrischen Rekonstruktion darf daher nicht auf einen einzigen Sensortyp festgelegt sein.

Konservatives geometrisches Modell gesuchter Objekte Das Ergebnis muss ein geometrisches Modell der Teilbereiche innerhalb des überwachten Raumbereichs sein, in welchen sich gesuchte Objekte aufhalten könnten. Es muss insofern konservativ sein, als außerhalb dieses geometrischen Modells sich kein gesuchtes Objekt befinden darf. Beispielsweise stellt die Annahme, dass sich ein Mensch innerhalb eines vertikalen Zylinders befindet, der auf Basis detektierter Beine mittels eines Laserscanners erzeugt wird [Som 05], im Allgemeinen kein konservatives Modell dar: Der Mensch könnte herausragen, wenn er die Arme ausstreckt; er könnte vom Sensor "übersehen" werden, wenn er sich über der Detektionsebene des Laserscanners befindet, weil er beispielsweise auf einen Stuhl gestiegen ist; er könnte sich außerhalb des Detektionsbereiches des Laserscanners befinden.

Sicherheit Insbesondere im Umfeld der Mensch/Roboter-Koexistenz und -Kooperation ist es notwendig, dass ein System zur wissens- und sensorbasierten geometrischen Rekonstruktion sicher und zuverlässig arbeitet. Hierbei gilt es Aspekte der Redundanz, Ausfallsicherheit und Fehlertoleranz zu berücksichtigen, sodass ein Einsatz im industriellen Umfeld ermöglicht wird.

Schnelligkeit und Genauigkeit Welche Anforderungen bezüglich Rechengeschwindigkeit bzw. -genauigkeit an eine Implementierung der wissens- und sensorbasierte geometrische Rekonstruktion gestellt werden, hängt von der konkreten Anwendung ab. Handelt es sich beispielsweise um die Mensch/Roboter-Koexistenz und -Kooperation und damit bei den

1. Einleitung

gesuchten Objekten um Menschen, so sollte es in den meisten Fällen ausreichend sein, wenn das System einzelne Finger erfassen kann und in weniger als 100ms ein Ergebnis liefert. Im Allgemeinen lässt sich salopp sagen: Je genauer und je schneller ein solches System arbeitet, desto besser ist es.

1.3. Aufgabenstellung

In dieser Arbeit gilt es die *ersten vier*, also die konzeptuellen Ziele, in folgenden drei Stufen zu erreichen:

1. Zunächst muss ein allgemeiner Formalismus entwickelt werden, der die gesteckten Ziele von Grund auf enthält und sich im Zuge mehrerer praxisnaher Beispiele als tauglich erweist: Die Integration von Wissen und Sensoren muss in konsistenter Weise durchführbar sein.
2. Mittels einer konkreten Implementierung in Form geeigneter Algorithmen und Datenstrukturen des nun zur Verfügung stehenden allgemeinen Formalismus sowie einem Großteil der beschriebenen Beispiele sollen die Umsetzbarkeit und gegebenenfalls Grenzen gezeigt werden.
3. Die Anwendbarkeit der Implementierung soll im Rahmen eines Beispiels im Bereich der Mensch/Roboter-Koexistenz gezeigt und gewonnene Erkenntnisse diskutiert werden.

1.4. Abgrenzung

Die *beiden letzten* Ziele, also "Sicherheit" sowie "Schnelligkeit und Genauigkeit" werden im Rahmen der vorliegenden Arbeit nicht berücksichtigt und erst im Ausblick kurz aufgegriffen. Sowohl eine sichere Implementierung, als auch die Untersuchung von Datenstrukturen und Algorithmen hinsichtlich einer zeitoptimalen Ausführung bzw. einer hochauflösenden Darstellung stellen zwar interessante, aber gleichzeitig auch äußerst komplexe Themengebiete dar.

Neben Anwesenheitssensoren, Farb- und Tiefenkameras werden keine weiteren Sensoren konkret betrachtet. Die Evaluation beschränkt sich zudem auf Farbkameras in einer realen Umwelt.

Untersuchungen zur komfortablen Einrichtung eines Systems, welches die wissens- und sensorbasierte geometrische Rekonstruktion durchführt, sowie die optimalen Platzierung von Sen-

soren zur Verbesserung der Güte geometrischer Rekonstruktionen sind ebenfalls nicht Teil der vorliegenden Arbeit, werden aber im Ausblick kurz angesprochen.

1.5. Kapitelübersicht

Im folgenden Kapitel wird der Stand der Forschung präsentiert. Die weitere Gliederung der vorliegenden Arbeit orientiert sich an der Aufgabenstellung und sieht damit wie folgt aus: In Kapitel 3 wird ein Formalismus entwickelt, welcher die ersten vier Ziele berücksichtigt. Im Zuge dessen werden praxisnahe Beispiele eingeführt. Eine Implementierung dieses Formalismus und der Beispiele wird in Kapitel 4 präsentiert. In Kooperation mit einem Partner aus der Industrie wurde die Implementierung im Rahmen einer Mensch/Roboter-Koexistenz angewendet. Ausführungen hierzu finden sich in Kapitel 5. In Kapitel 6 wird die Arbeit zusammengefasst und werden Schlüsse gezogen. Kapitel 7 gibt einen Ausblick auf weitere Forschungsmöglichkeiten. Die Implementierung (Kapitel 4) greift neben Voxeln auf weitere verschiedene geometrische Repräsentationen zurück. Diese werden im Anhangskapitel A diskutiert. Das Anhangskapitel B enthält eine alphabetisch sortierte Übersicht der verwendeten Symbole und Bezeichnungen.

2. Stand der Forschung

Ausgehend von den in der Aufgabenstellung betrachteten Zielen (Kapitel 1.3) werden in diesem Kapitel Ansätze bezüglich folgender Kriterien untersucht:

Überwachter Raumbereich Welchen geometrischen Raumbereich überwacht der untersuchte Ansatz? Ist dieser frei wählbar oder beispielsweise an die Platzierung der verwendeten Sensoren gebunden?

Verwendetes Wissen Auf welcher Information beruht der Ansatz? Ist diese Information fest vorgegeben oder sind Erweiterungen vorgesehen? Für den Fall, dass sich der Ansatz um zusätzliche Information erweitern lässt, auf welche Weise geschieht dies?

Verwendete Sensoren Mit welchen Sensoren arbeitet der Ansatz? Ist die Integration weiterer Sensoren vorgesehen? Geschieht die Integration weiterer Sensoren auf konsistente Weise?

Geometrisches Modell Wird mit einem (dreidimensionalen) geometrischen Modell gearbeitet? Wie sieht dieses geometrische Modell aus? Handelt es sich um ein konservatives Modell, sodass sich kein gesuchtes Objekt außerhalb dieses Modells befindet?

Diese vier Kriterien sind an die formulierten konzeptionellen Ziele (Kapitel 1.2) angelehnt und ermöglichen schließlich einen Vergleich mit der Aufgabenstellung der vorliegenden Arbeit in den Schlussfolgerungen (Kapitel 2.5). Im Idealfall könnte somit ein System exakt den formulierten Zielen entsprechen, wenn es also frei wählbare überwachte Raumbereiche, Integration beliebigen Wissens und beliebiger Sensoren sowie ein konservatives dreidimensionales geometrisches Modell als Ergebnis bereitstellt.

Da der Ursprung der Arbeit im Bereich der Mensch/Roboter-Koexistenz und -Kooperation liegt, gilt es, Ansätze aus dieser Domäne zu untersuchen (Kapitel 2.3). Eine andere wichtige

Domäne sind Multi-Kamera-Systeme, mit denen sich geometrische Modelle auf Basis aufgenommener Bilder rekonstruieren lassen (Kapitel 2.4). Zu bemerken sei, dass beide Domänen inzwischen eine Schnittmenge besitzen, in denen sich Systeme zur Mensch/Roboter-Koexistenz und -Kooperation auf Basis von Multi-Kamera-Systemen finden. Solche gemischten Systeme werden in Kapitel 2.3 detailliert beschrieben.

Bevor jedoch einzelne Ansätze untersucht werden, soll ein kurzer historischer Abriss Ursprünge und Entwicklungen beider Domänen skizzieren (Kapitel 2.1). Kapitel 2.2 liefert eine Übersicht über DFG-, EU- und BMBF-Projekte.

2.1. Historischer Überblick

Industrieroboter sind aus dem Wunsch nach einer universellen Maschine, welche nicht nur für eine einzige spezielle Aufgabe bestimmt ist, entstanden. Übliche Industrieroboter sind heutzutage dem menschlichen Arm nachempfunden und bieten bis zu sieben Freiheitsgrade. Die typischerweise schnellen und komplexen Bewegungen stellen ein hohes Sicherheitsrisiko für den Menschen dar, weswegen es bereits zu vielen Unfällen kam [Jian 87]. Infolgedessen wurden Normen entwickelt, welche zunächst eine strikte Trennung von Mensch und Roboter durch trennende Sicherheitsmaßnahmen wie beispielsweise Zäune fordern [Wikia, Otte 09]. Eine solche Trennung stellte anfangs kein Problem dar, da ohnehin die ausschließliche Vollautomatisierung seitens der Industrie angestrebt wurde. Die Sicherheitszäune wurden etwas später teilweise durch sichere Lichtgitter ersetzt. Der Vorteil dieser Lichtgitter besteht darin, dass sie im Bedarfsfall deaktiviert werden können. Im Laufe der Zeit haben sich die Anforderungen in der Industrie jedoch geändert: Die Bevölkerung ist gealtert, weshalb der Bedarf nach ergonomischen Arbeitsplätzen mit maschineller Unterstützung stets wächst. Kleinere Stückzahlen und Vielfalt in der Herstellung der Produkte lassen eine Vollautomatisierung unverhältnismäßig teuer werden. Wirtschaftlichkeit durch Einsparung von Räumlichkeiten und Herstellungszeiten sind im Rahmen der Globalisierung noch weiter in den Vordergrund gerückt. Zugleich ist aber auch die Rechenleistung heutiger Computer um Größenordnungen gestiegen. Vor diesem Hintergrund ist die Mensch/Roboter-Koexistenz und -Kooperation, welche die Stärken beider Partner nutzt, zu einem wichtigen aktuellen Thema geworden [Brec 05, Bart 11, Frau 11, Bort 11, Mura 00] und aufgrund heutiger Rechenleistung in greifbare Nähe gerückt. Infolgedessen haben sich zwei Stoßrichtungen entwickelt: Auf der einen Seite wird an der Entwicklung von Robotersystemen,

welche für den Menschen an sich ungefährlich sind, geforscht [Lens 10, Lim 00], auf der anderen Seite an Systemen, welche den Mensch in irgendeiner Weise erfassen und dann geeignet auf ihn reagieren. Letztere sind Anstoß zu der vorliegenden Arbeit gewesen und daher näher zu betrachten: Zur Erfassung des Menschen findet man in der Literatur den Einsatz verschiedenster Sensoren, beispielsweise Kontaktsensoren, Näherungssensoren, Kameras oder Laserscanner.

Kamerasysteme werden schon seit über drei Jahrzehnten genutzt, reale Objekte geometrisch zu erfassen, um sie danach virtuell mit dem Computer weiter verarbeiten zu können [Baum 74]. Statische Objekte lassen sich auf einem Drehteller mittels einer einzelnen kalibrierten Kamera erfassen. Bessere Ergebnisse können erzielt werden, indem ein Muster auf das sich drehende Objekt projiziert wird. Handelt es sich allerdings um Objekte, welche sich bewegen, so ist ein Drehteller eher ungeeignet. Man benötigt mehrere Kameras, die zum gleichen Zeitpunkt ein Bild des Objektes aufnehmen. Befindet sich das zu erfassende Objekt nicht in einem idealen, d.h. leeren und einfarbigen Raum, so müssen Verdeckungen explizit berücksichtigt werden, und es entstehen höhere Anforderungen an die Bildverarbeitung. Soll das geometrische Modell des Objektes möglichst zeitnahe vorliegen, müssen schnelle Algorithmen und geeignete Datenstrukturen gefunden werden, die dies bewerkstelligen.

2.2. Forschungsprojekte

Dieses Unterkapitel gibt einen Überblick hinsichtlich der Forschungsprojekte der DFG, der EU und des BMBF zunächst für die Thematik der Mensch/Roboter-Koexistenz und -Kooperation und danach für die Thematik der 3D Rekonstruktion.

Im Bereich der *Mensch/Roboter-Koexistenz und -Kooperation* liegt der Fokus seit mehreren Jahren insbesondere auf der Mensch/Roboter-Interaktion bzw. allgemein der Mensch/Maschine-Interaktion und -Kommunikation mit den Unterbereichen Sprach-, Gesten- und Mimikerkennerung sowie dem Verstehen und Erlernen von Verhaltensweisen und dem Ausführen zielorientierter Aktionen bzw. Reaktionen seitens des Roboters. Zu diesem Bereich gehört ebenso auch die Entwicklung geeigneter Hardware, beispielsweise Greifer, die der menschlichen Hand nachempfunden sind und so eine natürliche Interaktion gewährleisten sollen. Typischerweise geht es dabei um die Entwicklung sogenannter humanoider und mobiler Roboter und nicht um herkömmliche Industrieroboter. Dieser Schwerpunkt schlägt sich in der Anzahl der Forschungsprojekte nieder. So gibt es beispielsweise seit 2001 den von der Deutschen Forschungs-

2. Stand der Forschung

gemeinschaft (DFG) geförderten Sonderforschungsbereich (SFB) 588 [SFB5 12] (Humanoider Roboter: Lernende und kooperierende multimodale Roboter), dem selbst zahlreiche Projekte zugeordnet sind. Der SFB 673 [SFB6 06] (Ausrichtung in der Kommunikation) sowie der SFB 453 [SFB4 10] (Wirklichkeitsnahe Telepräsenz und Teleaktion) und der Sonderforschungsbereich/Transregio (TRR) 8 [TRR8 03] (Raumkognition – Schließen, Handeln, Interagieren) stellen verwandte Sonderforschungsbereiche hinsichtlich humanoider Roboter und der damit verbundenen Mensch/Maschine-Interaktion bzw. Mensch/Roboter-Interaktion dar. So beinhalten diese beispielsweise die Teilprojekte SFB 453 – M5 [SFB4 05] (Gestaltung und Evaluation der Mensch-Roboter-Schnittstelle), SFB 673 – C4 [SFB6] (Adaptive alignment in human-roboter-cooperation) und TRR 8 – N1-SocialSpace [TRR8 12] (Social Learning for Cognitive Robots). Auch auf Ebene des BMBF und vor allem der Europäischen Kommission sind weit über 50 ähnlich ausgerichtete Projekte in den vergangenen Jahren zu verzeichnen. Die Sicherheit des Menschen wird hierbei meistens nicht oder nur am Rande betrachtet, da dieser Aspekt bei humanoiden Robotern wenig Relevanz hat. Humanoide Roboter sind typischerweise ähnlich stark bzw. schwach wie ein Mensch zu gestalten, sodass sie ohnehin keine Gefahr für den Menschen darstellen. Eine Ausnahme ist das DFG-Projekt SIMERO-2 [SIME 05] (Sicherheitsstrategien für die Mensch/Roboter-Kooperation und -Koexistenz), in dessen Rahmen auch die vorliegende Arbeit entstanden ist. In diesem Projekt geht es darum, eine Mensch/Roboter-Koexistenz und -Kooperation mit herkömmlichen Industrierobotern zu gewährleisten. Hier stellt der Roboter eine Gefahr für den Menschen dar. Insofern bedarf es einer Komponente, die die möglichen Aufenthaltsorte des Menschen konservativ ermitteln und dem Robotersystem mitteilen kann.

Im Bereich der *3D Rekonstruktion* findet man Projekte, die thematisch zum Teil etwas näher mit der vorliegenden Arbeit verwandt sind:

Wie im historischen Überblick (Kapitel 2.1) bereits erwähnt, geht es bei der 3D Rekonstruktion oftmals um konkrete Objekte, von denen ein dreidimensionales virtuelles Ebenbild erzeugt werden soll und somit optimale Rahmenbedingungen dafür geschaffen werden können – beispielsweise durch die Vermeidung von Verdeckungen, durch die Platzierung der Objekte auf einem Drehteller, durch die optimale Platzierung des Sensors gegenüber dem zu rekonstruierenden Objekt etc. Ziel ist häufig, besonders genaue und/oder schöne Rekonstruktionen bezüglich der Geometrie und der Textur zu erzeugen sowie preislich günstige Systeme zu ermöglichen. Projekte mit dieser Ausrichtung sind beispielsweise [Akti 11, Reko 09, REAL 16]. Einige Projekte beschränken sich bei den Betrachtungen auf eine einzelne Kamera, um mittels "Structure

from Motion" Objekte zu rekonstruieren [REAL 96, VISI 03]. Die Kombination von Farb- und Tiefensensor vereinfacht den Prozess, da Geometrieinformationen direkt aus den Sensordaten gelesen werden können [RESO 98].

Oftmals liegen die Schwerpunkte der hier angesiedelten Projekte auch im 3D-Sehen, z.B. [2D3D 12], und in der Stereobildverarbeitung, beispielsweise [Hoch 11], in der Objekt-Erkennung, z.B. [PMDL 12], sowie in der Navigation und 3D-Kartierung in unbekanntem Umgebungen, z.B. [Towa 08]. Speziell die Mensch- und Posen-Erkennung sowie die Erkennung des Verhaltens eines Menschen in monokularen Bild- und Videoaufnahmen sind Inhalt von [HUMO 11].

Im Rahmen der BMBF-Fördermaßnahme „Altersgerechte Assistenzsysteme für ein gesundes und unabhängiges Leben“ [AAL 08] wurde das Projekt [sens 12b] gefördert, welches gemäß [Vitr 11] mit Tiefensensoren zu arbeiten scheint und mittels derer zwischen normalen und Sturzposen eines Menschen unterscheiden kann, um darauf basierend im Notfall ein Signal an eine Betreuungsstation zu übermitteln. Ein im gleichen Rahmen gefördertes Projekt [Sens 12a] ermittelt stattdessen anhand einer Bodenberührungsfäche, ob der Mensch liegt oder steht.

Im Bereich der 3D Rekonstruktion sind Projekte, deren Zielsetzung der vorliegenden Arbeit entspricht, nicht anzutreffen.

2.3. Ansätze zur Mensch/Roboter-Koexistenz und -Kooperation

Kontaktbasierte Systeme Der Kontakt zwischen Roboter und Mensch kann beispielsweise über die Motorströme, Drehmomentsensoren oder künstliche Häute festgestellt werden [Albu 07, Yama 96, Hein 03, Wink 09, Kerp 03, Heil 03]. Die Detektion eines Kontaktes dient beispielsweise zur Kollisionserkennung und/oder zur Führung des Roboterarms. Der überwachte Raumbereich ist gewissermaßen auf die Oberfläche des Roboterarms beschränkt, also auf den Teil des Roboters, an dem ein Kontakt detektiert werden kann. Ein geometrisches Modell beschränkt sich auf Rückschlüsse von Kräfteverteilungen auf dieser Oberfläche oder den gemessenen Drehmomenten in den Gelenken und ist somit in keinem Fall konservativ. Zur Integration weiterer (beliebiger) Sensoren oder anderen Wissens neben der Annahme dass die Sensoren einen Kontakt feststellen können, sind diese Ansätze im Rahmen ihrer Publikationen nicht vorgesehen. In [Albu 07] werden mehrere Sensortypen verwendet, um so die Sicherheit durch Redundanz

2. Stand der Forschung

zu gewährleisten. Ebenso werden in [Heil 03] zweierlei Sensortypen eingesetzt, mit dem Ziel, Annäherungen zum einen und Kontakte zum anderen erkennen zu können.

Näherungsbasierte Systeme Für Systeme auf Basis von Näherungssensoren gelten die gleichen Einschränkungen in Bezug auf die in dieser Arbeit formulierten Ziele wie für die kontaktbasierten Systeme. Der Unterschied liegt lediglich in der Größe des Überwachungsraumes: Ist er im Falle der kontaktbasierten Systeme direkt an die Oberfläche des Roboterarms gebunden, so erhält man mit Näherungssensoren eine Art Hülle um den Roboterarm, in der der Mensch detektiert werden kann. Insofern lassen sich Ausweichbewegungen erzeugen, ohne dass ein Kontakt hergestellt werden muss. Beispiele solcher näherungsensorikbasierter Systeme sind [Heil 03, Fedd 94, Lume 93, Gand 03].

Abstandsbaasierte Systeme In [Som 05] wird ein 2D-Laserscanner zur Detektion eines Menschen in Beinöhe eingesetzt, um anschließend einen vertikalen, mensch-hohen Zylinder um die detektierten Beine zu legen und die Geschwindigkeit eines Roboters in Abhängigkeit vom Abstand zu diesem Zylinder zu regeln. Der überwachte Raum ist auf den Detektionsbereich des Laserscanners begrenzt. Es wird angenommen, dass ein Mensch vor dem Roboter durch den Laserscanner detektiert wird. Eine Erweiterung des Systems durch andere Informationen oder Sensoren ist nicht vorgesehen. Gebeugte Personen sowie ausgestreckte Arme oder Personen, welche auf einen Stuhl steigen oder über die Detektionshöhe springen, werden nicht oder nur teilweise durch den Zylinder abgedeckt. Es handelt sich somit nicht um ein konservatives geometrisches Modell.

In [Zett 02] wird eine Wasserstrahl-Schneidanlage ohne trennende Schutz einrichtungen (OTS) präsentiert. Mittels zweier Roboter werden darin PKW-Teppiche geschnitten. Mehrere 2D-Laserscanner überwachen unterschiedliche Zonen und blockieren die Anwendung des Hochdruckstrahls bei Anwesenheit eines detektierten Menschen. Kann es am detektierten Aufenthaltsort des Menschen zu Stoß- oder Quetschverletzungen durch den Roboter kommen, wird dieser komplett gestoppt. Bei diesem speziellen Systemaufbau ist die beliebige Wahl eines überwachten Raumbereiches irrelevant. Es wird davon ausgegangen, dass mittels des Laserscanners der Mensch im Falle seiner Anwesenheit detektiert wird. Aufgrund der speziellen Aufgabe, die dieses System löst, ist eine Integration weiteren Wissens oder anderer Sensoren nicht vorgesehen. Ein konservatives geometrisches Modell wird nicht erzeugt.

Kamerabasierte Systeme In [Baer 92] wird ein Netzwerk stationär angebrachter Kameras zur Überwachung von Roboterarbeitsbereichen verwendet. Ziel des Systems ist es, den Roboter je nach Abstand zum Menschen in drei unterschiedlichen Geschwindigkeiten verfahren zu lassen. Ein nicht näher beschriebener Algorithmus ermittelt dazu basierend auf einem Differenzbildverfahren, ob sich *kein Hindernis*, ein *fernes Hindernis* oder ein *nahes Hindernis* im Arbeitsbereich befindet. Auf Verdeckungen, welche durch Gegenstände in den einzelnen Kameraperspektiven auftreten, wird nicht eingegangen. Insgesamt bleibt unklar, wie flexibel die Festlegung eines überwachten Raumbereiches ist und ob überhaupt ein geometrisches Modell erzeugt wird. Die Integration weiteren Wissens oder anderer Sensoren ist auch hier nicht vorgesehen.

Zu seinem kürzlich abgeschlossenen Promotionsprojekt am selben Lehrstuhl, an dem die vorliegende Arbeit verfasst wurde, beschreibt [Geck 11] in seiner Dissertation ein System zur sensorbasierten, echtzeitfähigen Online-Bahnplanung für die Mensch-Roboter-Koexistenz. Er verwendet ein Netzwerk kalibrierter, stationärer Farbkameras, um bildbasiert Abstände [Kuhn 06, Kuhn 07], im Wesentlichen auf Basis von Differenzbildverfahren, zwischen bekannten Objekten (Roboter) und unbekanntem Objekten (Menschen) zu ermitteln. Die so gewonnenen Abstände – auch für zukünftige Roboterpositionen – werden zur Geschwindigkeitsregelung und Online-Bahnplanung eines Industrieroboters genutzt. Die Distanzberechnungen finden dabei in einem Arbeitsraumvolumen statt, wobei jeder Teil dieses Arbeitsraumvolumen von mindestens einer Kamera überblickt werden können muss. Der überwachte Raumbereich ist daher von der Kameraplatzierung abhängig. Es wird angenommen, dass ein unbekanntes Objekt vor bekanntem Hintergrund immer als Vordergrund detektiert wird. Die Integration weiterer Sensoren ist nicht vorgesehen. Geometriemodelle bekannter Objekte werden genutzt, um Pixelmengen in den Kamerabildern zu beschreiben, in welchen Verdeckungen entlang des rückprojizierten Pixels vorkommen können. Solche Pixel müssen ähnlich einer Differenz im Differenzbild behandelt werden. Als weitere Information wird ein Parameter θ verwendet, der beschreibt, in wie vielen der Kameras maximal ein unbekanntes Objekt vollständig verdeckt sein kann. Dies ermöglicht die Identifikation von Pixelmengen mittels Epipolargeometrie, in denen kein unbekanntes Objekt enthalten sein kann. Das Modell der unbekanntem Objekte wird zunächst durch Pixelmengen beschrieben und anschließend daraus ein Abstand abgeleitet. Dieses (Pixelmengen-)Modell lässt sich als konservativ innerhalb des Arbeitsraumvolumen bewerten, stellt jedoch keine 3D-Rekonstruktion dar. Vorgänger dieses Systems ist [Eber 03], welches gegenüber [Geck 11] einige Einschränkungen aufweist: Unter anderem wird lediglich der

2. Stand der Forschung

Roboter selbst als einziger sichtverdeckender Gegenstand angenommen. Statt Abstandsberechnungen werden binäre Aussagen über Kollisionen zukünftiger Roboterkonfigurationen getroffen. Die Integration weiteren Wissens ist in beiden Systemen nicht vorgesehen.

In [Ladi 08a, Ladi 08b] wird mittels eines Multi-Kamerasystems die visuelle Hülle auf Basis von Silhouettenbildern bestimmt. Der Fokus der Arbeit liegt dabei auf der zeitlichen Optimierung dieser Berechnungen. Hierfür werden Voxel- und Octree-Datenstrukturen verwendet sowie eine parallele Verarbeitung auf einer GPU betrachtet. Verdeckungen durch Gegenstände im überwachten Raumbereich werden mittels sogenannter "Occlusion masks" [Guan 06] aufgelöst. Es handelt sich hierbei um Pixelmengen, in deren Rückprojektion der Mensch verdeckt sein könnte. Diese Pixelmengen werden gemeinsam wie die Vordergrund-Pixel der Silhouettenbilder zur Bestimmung der visuellen Hülle verwendet. Es wird davon ausgegangen, dass die nicht-verdeckten Teile eines Menschen durch die Kameras detektiert werden können und korrekt als Silhouette erscheinen. Das System wird verwendet, um Kollisionen mit medizinischen Robotern vorausschauend zu erkennen. Der überwachte Raumbereich ist durch die Kameraplatzierung festgelegt: Um eine Rekonstruktion durchführen zu können, müssen mindestens vier Kameras diesen Bereich einsehen können. In einem der vorgeschlagenen Algorithmen wird zur Ermittlung, ob ein Voxel belegt ist, die Auflösung des Silhouettenbildes derart reduziert, dass der Voxel ungefähr auf einen Pixel projiziert wird. Insofern ist das Ergebnis ein geometrisches Modell, welches aufgrund der etwas ungenauen Voxelbetrachtung innerhalb des vorbestimmten Raumbereiches als eingeschränkt konservativ zu bewerten ist. Die Integration weiterer Sensoren oder weiteren Wissens ist nicht vorgesehen.

In [Viet 08] werden zwei Farbkameras zur Absicherung eines statischen Gefahrenbereiches verwendet, indem zunächst ein sich nähernder Mensch mit der ersten Kamera und schließlich dessen einzelne Körperteile mit der zweiten Kamera detektiert werden. Gelangt ein Körperteil in den Gefahrenbereich einer überwachten Maschine, kann diese angehalten werden. Aufgrund der speziellen Zielsetzung des Systems ist die Festlegung eines überwachten Raumbereiches irrelevant, hängt aber unmittelbar mit der Platzierung der Kameras zusammen, welche natürlich direkt auf den Gefahrenbereich ausgerichtet sein müssen. Das verwendete Wissen, bzw. die getroffene Annahme ist die, dass der Mensch und seine Körperteile mittels der Kameras und der verwendeten Algorithmen detektiert werden können. Eine Integration weiteren Wissens oder weiterer Sensoren ist nicht vorgesehen. Das System arbeitet komplett im Bildbereich und lässt sich im überwachten Raumbereich als konservativ bewerten. Es liefert jedoch keinerlei

geometrisches Modell.

In [Thie 05] werden mehrere über einem SCARA-Roboter stationär angebrachte Farbkameras verwendet, um eine Mensch/Roboter-Kooperation in der Kleinteilmontage zu ermöglichen, indem das Roboterverhalten (Bahn und Geschwindigkeit) in Abhängigkeit zum detektierten Werker (Abstand) angepasst wird. Die Platzierung des Menschen gegenüber dem Roboter sowie dem gemeinsamen Arbeitsbereich wurde bezüglich ergonomischer Aspekte ermittelt und festgelegt. Auch hier ist der zu überwachende Raumbereich aufgrund der speziellen Zielsetzung und Anordnung auf den gemeinsamen (kleinen) Arbeitsbereich festgelegt und bedarf keiner (beliebigen) Änderung. Es wird die Annahme getroffen, dass das nicht näher spezifizierte Bildverarbeitungssystem den Werker detektiert. Eine Erweiterung mittels anderer Sensoren oder anderen Wissens ist nicht thematisiert. In vorausgegangenen Publikationen [Spin 02, Thie 02] wurden die Hände des Werkers über ihre Hautfarbe im Kamerabild erkannt. Ob ein geometrisches Modell berechnet wird, bleibt unklar.

In [Hoov 99, Lee 03] geht es um die Überwachung und Steuerung mobiler Roboter durch ein Netzwerk fest installierter Kameras. Die Auslagerung der Wahrnehmungs- und Steuerungskomponente aus den mobilen Robotern ermöglicht es unter anderem, diese einfacher zu gestalten. Eine kollisionsfreie zweidimensionale Bahnplanung (auf dem Boden) dieser mobilen Roboter findet auf Basis der ausgewerteten Kamerabilder statt, wodurch der Aktionsbereich auf den sichtbaren Bereich der Kameras beschränkt ist. Es wird davon ausgegangen, dass Objekte (z.B. Menschen) und Roboter mittels der Kameras detektiert werden können. Aufgrund der Bahnplanung auf einer zweidimensionalen Ebene ist auch nur ein zweidimensionales geometrisches Modell erforderlich. Die Integration weiterer (beliebiger) Sensoren oder anderen Wissens ist nicht vorgesehen.

Ein zertifiziertes, sicheres kamerabasiertes Produkt zur Überwachung von Zonen steht mit dem SafetyEye der Firma Pilz [Pilz] zur Verfügung (Stand 23.09.2011: Kategorie 3 nach EN 954-1, SIL2 nach IEC 61508, PL d nach EN ISO 13849-1 nach DIN EN 61496). Es handelt sich von der Sensorik her hierbei im Prinzip um eine sichere Stereokamera bestehend aus 3 Einzelkameras, um so die geforderte Sicherheit zu gewährleisten. Auch alle weiteren Komponenten dieses Systems sind entsprechend der Normen sicher ausgelegt. Dieses System erlaubt eine grafische Konfiguration der zu überwachenden Zonen per Software, welche auch überprüfen kann, ob die Licht- und Umweltverhältnisse derart beschaffen sind, dass das System sicher arbeitet. Solche konfigurierten Zonen können dynamisch, an einen Prozess angepasst, ein- und ausgeschaltet

2. Stand der Forschung

werden. Zu bemerken ist jedoch auch, dass der Roboter selbst *nicht* innerhalb des Volumens einer Überwachungszone liegen darf, da dieser durch seine Existenz direkt eine Schutzverletzung auslösen würde. Überwachungszone dürfen daher nur *außerhalb* der sog. Hüllkurve des gefährlichen Bereiches liegen [Pilz 11]. Aufgrund der virtuellen Überwachungszone ist das System rekonfigurierbar, gleicht aber ansonsten der herkömmlichen Sicherheitszaun-Lösung: Entweder der (virtuelle) Zaun wird unterbrochen oder nicht. Der überwachbare Raumbereich ist auf den Sichtbereich des SafetyEye beschränkt. Es wird zugesichert, dass der Mensch unter geeigneten Licht- und Umweltverhältnissen durch das System detektierbar ist, sofern das SafetyEye nicht weiter als 7,5 m entfernt platziert ist. Aufgrund der Einschränkung hinsichtlich der maximalen Entfernung des SafetyEyes ergibt sich eine maximal überwachbare Bodenfläche von $9,8 \text{ m} \times 7,4 \text{ m}$. Ein geometrisches Modell des detektierten Objektes liegt nur in Form einer Zonenverletzung vor. Dies ist insofern konservativ, als sich garantiert kein Objekt innerhalb der Zone aufhält, wenn keines darin detektiert wird. Die Integration weiterer Sensoren und weiteren Wissens ist nicht vorgesehen.

Neben Farbkameras werden inzwischen auch Tiefenkameras zur Überwachung im Kontext der Mensch/Roboter-Koexistenz und Kooperation eingesetzt.

[Schi 09a] verwendet eine einzelne Time-of-Flight-Tiefenkamera (ToF) zur Ermittlung von belegten Voxeln, um dann mittels eines Potentialfeldplaners Kollisionen mit einem Roboterarm zu vermeiden. Der überwachbare Raumbereich ist auf den Sichtbereich der Kamera festgelegt und somit nicht frei wählbar. Es wird nicht berücksichtigt, dass sich Objekte auch außerhalb des Sichtbereiches aufhalten könnten. Inwieweit das geometrische Modell innerhalb des Kamera-Sichtbereiches konservativ ist, bleibt unklar, zumal es zu Verdeckungen im Tiefenkamerabild durch den Roboter selbst kommen kann. Im Ausblick wird die Verwendung mehrerer Tiefenkameras vorgeschlagen. Ansonsten ist die Integration weiteren Wissens oder weiterer Sensortypen nicht vorgesehen.

In dem zu dieser Arbeit parallel laufenden Promotionsprojekt beschreibt [Fisc 09a] ein System, bestehend aus mehreren Tiefenkameras zur Kollisionsdetektion zwischen Industrierobotern und unbekanntem Objekten (Menschen). Der überwachte Raumbereich beschränkt sich auf die Schnittmenge aller Kamerasichten und ist daher nicht frei wählbar. Neben der Tiefenbildinformation und der Annahme, dass unbekannte Objekte von den Tiefenkameras detektiert werden, wird ein geometrisches Modell des Industrieroboters verwendet. Ein Zwischenergebnis des Systems sind konvexe Hüllen, in welchen unbekannte Objekte angenommen werden. Dieses dreidi-

mensionale geometrische Modell ist unter der Annahme, dass sich der Mensch nicht in einem für sämtliche Kameras verdeckten Bereich “verstecken” kann, eingeschränkt konservativ zu bewerten, da aufgrund einer Tiefeninterpolation zwischen den Pixelnmittelpunkten das gesuchte Objekt aus der Hülle herausragen kann. Anschließend wird der kürzeste Abstand zwischen diesen konvexen Hüllen und dem Roboter ermittelt. Mit [Fisc 09b] wird das System derart erweitert, dass zugleich auch Farbkameras eingesetzt werden können und die Rekonstruktion in der Vereinigung der Kamerasichten berechnet wird. Die Integration weiteren Wissens oder weiterer Sensoren ist jedoch nicht vorgesehen.

In [Wink 07] wird eine einzelne Time-of-Flight-Tiefenkamera verwendet, um einen Menschen in der Nähe eines Roboters zu detektieren und abstandsabhängig die drei Signale “No Risk”, “Warn” und “Stop” zu generieren. Das Tiefenbild der Kamera wird dazu als triangulierte Oberfläche in den Raum rückprojiziert und mit unterschiedlich expandierten Robotervolumina auf Überschneidung getestet. Der überwachte Raumbereich ist somit auf das sichtbare Volumen der Kamera begrenzt. Das resultierende geometrische Modell ist nicht konservativ, da Abschattungen durch den Roboter oder andere Gegenstände nicht berücksichtigt werden. Auch hier ist weder die Integration weiteren Wissens noch weiterer Sensoren vorgesehen.

In [Worn 09] wird ebenfalls eine einzelne Time-of-Flight-Tiefenkamera genutzt, um aus der Vogelperspektive den Mensch im gemeinsamen Arbeitsbereich zu erfassen. Der überwachte Raumbereich ist auf den Sichtbereich der verwendeten Kamera eingeschränkt. Es wird angenommen, dass sich der Mensch mittels dieser Kamera detektieren lässt. Darauf basierend wird ein eingeschränktes menschliches Modell errechnet und neben reinen Abständen auch die Lage des Werkers gegenüber dem Roboter bestimmt. Da im Fall einer Verdeckung das menschliche Modell nicht richtig angepasst werden könnte, lässt sich das resultierende geometrische Modell nicht als zwingend konservativ bezeichnen. Die Integration weiteren Wissens oder weiterer Sensoren ist nicht vorgesehen.

Systeme mit Sensordatenfusion In diesem Abschnitt werden Ansätze beschrieben, die die Fusion von Daten von mehr als zwei Sensortypen betrachten.

[Karl 00] schlägt vor, mithilfe unterschiedlicher Sensoren (siehe unten) die Position einer Person im gemeinsamen Arbeitsraum zu bestimmen und in Verbindung mit der aktuellen Robotergeschwindigkeit und Bewegungsrichtung ein Risikolevel mittels Fuzzy-Logik zu errechnen. Bei Anstieg des Risikolevels soll zunächst ein Signal, dann eine Geschwindigkeitsreduzie-

2. Stand der Forschung

nung und schließlich ein Stillstand des Roboters erfolgen. Die Detektion soll in jedem Bereich mittels mindestens zweier Sensortypen geschehen, um so Redundanz zu gewährleisten und Unsicherheiten auszuschließen. Ein zum Teil umgesetztes System umfasst eine einzelne Lichtschranke zur Ermittlung, ob eine Person den gemeinsamen Arbeitsraum betreten hat, einen kapazitiven Sensor mit zwei Bereichen und 16 Ultraschallsensoren sowie zwei orthogonal angeordnet, liegende Lichtgitter. Die zwei Bereiche des kapazitiven Sensors beschreiben zwei Kreisscheiben mit unterschiedlichen Radien und dem Roboter als Mittelpunkt. So ist feststellbar, ob sich die Person nahe oder fern des Roboters befindet. Die Ultraschallsensoren sind um die Roboterbasis befestigt, sodass sie in 16 Sektoren um den Roboter eine Person detektieren können. Die beiden liegenden Lichtgitter ermöglichen eine Lokalisation einer Person, die den Arbeitsraum durch die Lichtschranke betreten hat. Bei diesem speziellen System ist der zu überwachende Raumbereich festgelegt, und es besteht auch nicht die Anforderung, diesen zu ändern. Es lassen sich unterschiedliche Sensoren integrieren, jedoch wird kein konservatives geometrisches Modell erzeugt. Neben der Annahme, dass die Sensoren in der Lage sind die Person zu erfassen, wird kein weiteres Wissen integriert.

In [Zura 01] wird eine Simulation beschrieben, in welcher Daten unterschiedlicher Sensoren mittels eines neuronalen Netzwerks fusioniert werden, um den Aufenthaltsort eines Menschen in der Nähe eines Roboters in einem zweidimensionalen Occupancy-Grid parallel zum Boden zu ermitteln. Dieser Aufenthaltsort wird zur Geschwindigkeitsregulierung des Roboters verwendet. Jeder Sensor trägt in eine lokale Karte auf dem Grid die Wahrscheinlichkeit ein, mit welcher ein Objekt sich in seinem Detektionsbereich aufhält. Die lokalen Karten werden mittels des neuronalen Netzwerks zu einer gesamten, kombinierten Karte fusioniert. Die kombinierte Karte enthält Regionen mit unterschiedlichen Aufenthaltswahrscheinlichkeiten des Menschen, welche dann in einem Fuzzy-Regler zur Geschwindigkeitsregulierung des Roboterarms genutzt wird. Beispielhaft werden vier simulierte Sensoren in einem Grid der Größe 11×15 Felder eingesetzt: Zwei Ultraschall-, ein Infrarot- und ein kapazitiver Sensor. Dieser Ansatz ist im Prinzip um beliebige Sensoren erweiterbar, welche für ihren (lokalen) Detektionsbereich eine Aufenthaltswahrscheinlichkeit eines Objektes ermitteln können. Das zweidimensionale Occupancy-Grid ist durch die Betrachtung von lokalen Karten nicht direkt an die Sensoren gebunden, sodass man von einem beliebig überwachbaren Raumbereich sprechen kann. Aufgrund von Wahrscheinlichkeiten auf dem zweidimensionalen Occupancy-Grid lässt sich nicht von einem konservativen geometrischen Modell sprechen. Es wird angenommen, dass die verwendeten Sensoren in der

Lage sind, die zu beobachtenden Objekte zu erfassen. Weiteres Wissen über die zu beobachtenden Objekte, wie beispielsweise Mindestvolumina, Maximaldistanzen, etc. wird nicht berücksichtigt. Eine Integration solchen Wissens ist auch nicht vorgesehen.

2.4. Ansätze im Bereich der Multi-Kamera-Rekonstruktion

Im Bereich der Multi-Kamera-Rekonstruktion gibt es nahezu unzählige Publikationen, die im Rahmen der vorliegenden Arbeit gar nicht im Einzelnen diskutiert werden können. Es ist jedoch möglich, verschiedene Merkmale anzugeben, anhand derer sich die Publikationen klassifizieren lassen. Kapitel 2.4.1 dient daher zunächst einmal als Überblick, um vor allem die Schwerpunkte der Forschung auf diesem Bereich aufzuzeigen. Über diese Schwerpunkte lassen sich Überlegungen anstellen, welche eine Einordnung in den für diese Arbeit relevanten Stand der Forschung (Kapitel 2.4.2) erlauben.

2.4.1. Merkmale und Einordnung von Multi-Kamera-Rekonstruktionsmethoden

Technologie Mit Technologie ist die zur Verfügung stehende Pixelinformation gemeint: So lässt sich im Wesentlichen zwischen *Farbkameras* [Szel 93, Mont 04, Furu 06, Cail 04, Tung 09, Laze 07, Guan 06, Schi 09b, Ladi 08a, Horn 06, Kutu 00, Seit 99, Seit 06, Dyer 01, Slab 01, Casa 06] und *Tiefenkameras* [Fisc 09a] unterscheiden, wobei Farbkameras eine Farbinformation und Tiefenkameras eine Distanz zum nächsten Gegenstand pro Pixel enthalten. Grauwert- oder Infrarotkameras lassen sich hierbei als spezielle Ausprägung den Farbkameras unterordnen. Mit [Kim 09, Guan 08, Fisc 09b] gibt es Publikationen, welche Farbkameras und Tiefenkameras gleichzeitig verwenden. [Kim 09] nutzt die direkte Kombination von Tiefen- und Farbsensor, um so zu einem exakteren Ergebnis zu kommen: Das Rauschen der gering-auflösenden Tiefenkamera soll durch Stereoalgorithmen unterdrückt, und die Lösung des Korrespondenzproblems durch die Tiefenkameras verbessert werden. Eine erst kürzlich erschienene Publikation [Hofm 11] behandelt die Fusion mehrerer Kameratypen (Farbkameras, Wärmekameras, Infrarotkameras und Tiefenkameras). Die Farbkameras, Wärmekameras und Infrarotkameras werden verwendet, um jeweils Silhouettenbilder auf Basis eines Background-Subtractions zu erzeugen. Die Tiefenka-

2. Stand der Forschung

mera wird separat behandelt, indem die Tiefenwerte genutzt werden. Insofern gleicht auch dort die Aufteilung der oben angegebenen.

Rekonstruktionsprinzip Das Rekonstruktionsprinzip bezieht sich ausschließlich auf Farbkameras und beschreibt die grundlegende Vorgehensweise zur Erzeugung des dreidimensionalen virtuellen Ebenbildes: So kann basierend auf Silhouetten die *visuelle Hülle* rekonstruiert [Szel 93, Cail 04, Laze 07, Matu 01, Fran 09, Guan 06, Schi 09b, Ladi 08a, Seit 99, Casa 06] und durch Anwendung von *Multi-Kamera-Stereoalgorithmen* auf Farbbildern ein typischerweise exakteres (foto-konsistentes) Modell berechnet werden [Mont 04, Tung 09, Kutu 00, Seit 99, Seit 06], was meist aber mit einem größeren Zeitaufwand einhergeht. Bedarf ein Silhouettenbild der Trennung von Vorder- und Hintergrundpixeln, so sind Stereoalgorithmen darauf nicht angewiesen. Manchmal wird aber auch zur Verbesserung der Berechnungszeit als erste Annäherung an die Objektgeometrie eine visuelle Hülle bestimmt, um danach Multi-Kamera-Stereoalgorithmen zu verwenden [Furu 06, Horn 06]. In [Hofm 11, Fisc 09b] wird eine Mischung der Silhouettenbasierten Herangehensweise mit Tiefenkamerabildern verwendet.

Zeit/Qualität-Verhältnis Zum einen gibt es Anwendungen, bei denen möglichst *schnell* Ergebnisse vorliegen müssen [Szel 93, Furu 06, Cail 04, Laze 07, Matu 01, Fran 09, Ladi 08a], bei anderen aber möglichst *exakte* Ergebnisse benötigt werden [Tung 09, Horn 06, Kutu 00, Seit 99]. Der Fokus beeinflusst die Wahl des Rekonstruktionsprinzips bei Farbkameras und ansonsten die Wahl der Algorithmen und Datenstrukturen.

Algorithmen In einem Fall bedarf es besonders *robuster* Algorithmen, im anderen Fall besonders *schneller*, beispielsweise auch durch Parallelisierung auf der GPU [Schi 09b, Ladi 08a]. Die Wahl der Algorithmen und Datenstrukturen beeinflusst sich dabei gegenseitig. Aber auch unterschiedliche Algorithmen bei gleichen Datenstrukturen sind möglich: Werden beispielsweise im Rahmen eines Stereo-Ansatzes im einen Fall direkt Korrespondenzen gelöst, so löst im anderen Fall ein Algorithmus das Problem mit der Minimierung von Energiefunktionen.

Datenstrukturen Allein durch die Vielfalt der Datenstrukturen lassen sich Publikationen in etliche Klassen aufteilen: So werden beispielsweise als häufigste Vertreter *Voxel* [Schi 09b, Ladi 08a, Horn 06, Kutu 00, Seit 99], *Octrees* [Szel 93, Mont 04, Cail 04, Ladi 08a], *Conexel*

[Casa 06], *Polyeder* [Furu 06, Laze 07, Fran 09, Guan 06], im Dreidimensionalen liegende *Polygone* [Matu 01] oder *Occupancy-Grids* [Guan 08, Hofm 11] verwendet. Ein weiterer Aspekt ist, ob das Resultat eine texturierte Oberfläche bzw. Farbinformation in Voxeln enthalten soll.

Verdeckungen Je nach Anwendung kann sich das zu rekonstruierende Objekt in einer ansonsten völlig leeren Umwelt befinden, oder aber auch umgeben sein von sicht-verdeckenden Gegenständen. Im ersten Fall ist es unnötig, ein Verdeckungskonzept im Algorithmus zu integrieren, während dies im zweiten Fall insbesondere bei dem Rekonstruktionsprinzip der visuellen Hülle zwingend notwendig ist. Insofern kann man Ansätze auch dadurch unterscheiden, ob sie *Verdeckungen berücksichtigen* oder ob sie *keine Verdeckungen berücksichtigen* [Szel 93, Furu 06, Cail 04, Tung 09, Laze 07, Matu 01, Fran 09, Horn 06, Casa 06, Hofm 11]. Dieser Aspekt ist relativ neu und wurde zunächst mittels Occlusion-Masks [Guan 06, Ladi 08a] gelöst und später erstmals in einer eigenen Vorarbeit im Detail [Henr 08, Kuhn 09a, Kuhn 09b] analysiert. Ebenfalls hat sich [Schi 09b] damit auseinandergesetzt, jedoch mit der Einschränkung auf statische Sichthindernisse.

2.4.2. Bewertung

Die anhand der vorgestellten Merkmale einsortierten Ansätze im Bereich der Multi-Kamera-Rekonstruktion lassen sich gemäß der Kriterien, die zu Beginn des Kapitels aufgestellt wurden nun bewerten:

Geometrisches Modell Verfahren, die nach dem Rekonstruktionsprinzip der visuellen Hülle arbeiten und Verdeckungen nicht explizit berücksichtigen, resultieren zwangsläufig in einem nicht-konservativen geometrischen Modell, sobald Gegenstände im betrachteten Raum die gesuchten Objekte in mindestens einem Kamerabild teilweise verdecken. Diejenigen Verfahren, die Verdeckungen berücksichtigen, können aufgrund Ihrer Datenstruktur dennoch nicht-konservativ sein: Beispielsweise ist es nicht ausreichend, lediglich den Voxelmittelpunkt zu betrachten, um das gesamte quaderförmige Voxelvolumen als "frei" identifizieren zu können. Unter der Annahme, dass die Kameras korrekt kalibriert sind und die Silhouette des gesuchten Objektes im Kamerabild fehlerfrei ermittelt werden kann, müssen also sowohl Verdeckungen berücksichtigt, als auch eine geeignete Datenstruktur gewählt werden, um ein konservatives geometrisches Modell zu gewährleisten.

2. Stand der Forschung

Verfahren, die nach dem Rekonstruktionsprinzip der Fotokonsistenz arbeiten, berücksichtigen Verdeckungen implizit, da sie die gesamte Szene im Sichtbereich der Kameras, also auch Gegenstände, welche gesuchte Objekte verdecken könnten, rekonstruieren. Wieder unter der Annahme, dass die Kameras korrekt kalibriert sind und die Ermittlung der Fotokonsistenz fehlerfrei ist, bedarf es lediglich der geeigneten Datenstruktur, um ein konservatives geometrisches Modell zu erzeugen.

Ein geometrisches Modell wird mittels aller Ansätze erzeugt, jedoch wird der Aspekt der Konservativität nicht diskutiert und auch nicht berücksichtigt, was vermutlich damit zusammenhängt, dass der Schwerpunkt bislang typischerweise auf der schönen und nicht auf der sicheren/konservativen Rekonstruktion lag.

Überwachter Raumbereich Algorithmen, welche die visuelle Hülle als Rekonstruktionsprinzip verwenden, arbeiteten ursprünglich ausschließlich auf dem Schnittvolumen aller Kamerasichtbereiche. Erst in neueren Publikationen findet man Verallgemeinerungen, die auch auf der Vereinigung aller Kamerasichtbereiche arbeiten. Ähnliches gilt für Multi-Kamera-Stereo-Ansätze: Nur wo Kamerainformationen verfügbar sind, lassen sich Objekte mittels Korrespondenzen rekonstruieren. Auch die genannten Tiefenkamera-Ansätze rekonstruieren nur dort, wo mindestens eine Kamera hinsehen kann. Damit gilt für die bisherigen publizierten Ansätze, abgesehen von den eigenen, dass der überwachte Raumbereich außerhalb der Vereinigung aller Kamerasichtbereiche nicht frei wählbar ist.

Verwendetes Wissen Als Wissen wird lediglich die Annahme über die Detektierbarkeit des zu rekonstruierenden Objektes mittels der Farb- oder Tiefenkameras verwendet. Die Integration weiteren Wissens ist (abgesehen von eigenen Beiträgen [Kuhn 09a, Kuhn 09b]) nicht vorgesehen.

Verwendete Sensoren Neben reinen Farb- oder Tiefenkamera-Ansätzen findet man auch Ansätze, welche beide Technologien fusionieren. Mit [Hofm 11] existiert ein Ansatz, der zudem noch Infrarot und Wärmekameras – aber diese auf gleiche Weise wie Farbkameras zur Silhouetten-Bildung – nutzt.

2.5. Schlussfolgerungen

Bei den Ansätzen zur Mensch/Roboter-Koexistenz und -Kooperation finden sich viele, sehr unterschiedliche und spezielle Lösungen (Abbildung 2.1), welche jedoch keines der in der Aufgabenstellung genannten Ziele anstrebt, mit der Ausnahme des simulierten Systems von [Zura 01], in dem die Integration beliebiger Sensoren im Prinzip möglich ist. Der überwachbare Raumbereich ist üblicherweise aufgrund eines sehr speziellen Systemaufbaus oder durch die Anordnung der Sensoren festgelegt und kann daher nicht beliebig gewählt werden. Die Integration von Wissen beschränkt sich typischerweise auf die Annahme, dass die verwendeten Sensoren in der Lage sind den Mensch zu erfassen. Ein geometrisches Modell im Dreidimensionalen wird selten erzeugt, wobei der Aspekt der Konservativität eine untergeordnete Rolle spielt.

Ansätze im Bereich der Multi-Kamera-Rekonstruktion haben ihren Ursprung in der Erfassung und geometrischen Modellierung von Objekten und *nicht* etwa in der Erfassung und geometrischen Modellierung von *möglichen Aufenthaltsorten* dieser Objekte. Nutzt man in dem ersten Fall vorhandene Sensorinformation über das Objekt selbst, so ist es im zweiten Fall zielführender, die Sensorinformation über die *Abwesenheit* des Objektes zu verwenden. Dass also typische Verfahren zur Rekonstruktion nur in der Schnittmenge oder der Vereinigung aller Kamerasichten korrekt funktionieren, ist historisch begründet. Auch die Integration von Wissen spielt in der ursprünglichen Betrachtungsweise keine Rolle. Die *explizite* Verdeckungsmodellierung neuerer Ansätze wandelt sich übrigens durch die Argumentation über die Abwesenheit von Objekten (= Freiräume) in eine *implizite* Modellierung. Ein konservatives geometrisches Modell spielt gegenüber einem schönen, möglichst realitätsnahen Modell eine untergeordnete Rolle.

Der Stand der Forschung zeigt also, dass trotz des Bedarfs an einem gemäß den Zielen ausgerichteten Systems bislang nur sehr wenige und sehr eingeschränkte Lösungen existieren.

2. Stand der Forschung

		Überwachter Raumbereich	Verwendetes Wissen	Verwendete Sensoren	Geometrisches Modell
<i>Ansätze zur Mensch/Roboter-Koexistenz und -Kooperation</i>					
Kontaktbasierte Systeme	[Yama 96, Hein 03, Wink 09, Kerp 03]	○	○	○	○
	[Albu 07, Heil 03]	○	○	◐	○
Näherungsbasierte Systeme	[Fedd 94, Lume 93, Gand 03]	○	○	○	○
	[Heil 03]	○	○	◐	○
Abstands-basierte Systeme	[Som 05]	○	○	○	◐
	[Zett 02]	○	○	○	○
Kamera-basierte Systeme	[Baer 92]	○	○	○	○
	[Geck 11, Eber 03]	○	○	○	◐
	[Ladi 08a, Ladi 08b]	○	○	○	◐
	[Viet 08]	○	○	○	○
	[Thie 05, Spin 02, Thie 02]	○	○	○	○
	[Hoov 99, Lee 03]	○	○	○	○
	[Pilz, Pilz 11]	○	○	○	◐
	[Schi 09a]	○	○	○	◐
	[Fisc 09a, Fisc 09b]	○	○	◐	◐
	[Wink 07]	○	○	○	◐
	[Worn 09]	○	○	○	◐
Systeme mit Sensor-datenfusion	[Karl 00]	○	○	◐	○
	[Zura 01]	◐	○	◐	○
<i>Ansätze im Bereich der Multi-Kamera-Rekonstruktion</i>					
Eigene	[Henr 08]	●	○	○	◐
	[Kuhn 09a, Kuhn 09b]	●	●	◐	◐
Andere	[Cail 04, Casa 06, Dyer 01, Fisc 09a, Fran09, Furu 06 Guan 06, Horn 06 Kutu 00, Ladi 08a Laze 07, Matu01, Mont 04, Schi 09b Seit 99, Seit 06 Slab 01, Szel 93 Tung 09]	○	○	○	◐
	[Kim 09, Guan 08, Fisc 09b, Hofm 11]	○	○	◐	◐
Hier		●	●	●	●
Legende: ● Erfüllt ◐ Eingeschränkt erfüllt ○ Nicht erfüllt					

Abbildung 2.1.: Übersicht der diskutierten Ansätze.

3. Modellierung

Dieses Kapitel beschreibt die wissens- und sensorbasierte geometrische Rekonstruktion aus einer mathematischen Perspektive für n -dimensionale kontinuierliche euklidische Räume. Es ist in drei Teile zuzüglich einer Zusammenfassung und Schlussfolgerungen (Kapitel 3.4) untergliedert: Im ersten Unterkapitel 3.1 werden grundlegende Konzepte eingeführt. Es wird beschrieben was eine geometrische Rekonstruktion ist und wie man mittels Einsatz von Wissen und Sensoren schrittweise ein möglichst genaues geometrisches Modell der gesuchten Objekte erzeugt. Anschließend wird in Unterkapitel 3.2 an konkreten Wissens- und Sensor-Beispielen gezeigt, wie die eingeführten Konzepte genutzt werden können: Es wird unter anderem gezeigt, wie man anhand von Volumeninhalten ausschließen kann, dass sich ein Mensch in bestimmten Raumregionen aufhält und wie man zeitliche Aspekte, Distanzinformationen und Farbkameras nutzen kann um weitere Raumregionen auszuschließen. Im dritten Unterkapitel 3.3 werden diese Beispiele aufgegriffen und anhand eines Szenarios der Mensch/Roboter-Koexistenz zu einer praxisnahen Anwendung zusammenggefügt. Die eingeführten Symbole und Bezeichnungen sind in Anhang B zur Übersicht alphabetisch aufgelistet und können dort nachgeschlagen werden.

3.1. Geometrische Rekonstruktion und Plausibilisierung

Informell gesprochen, stellt eine geometrische Rekonstruktion ein konservatives Modell gesuchter Objekte innerhalb einer Rekonstruktionszone dar. Konservativ bedeutet hier, dass der Anteil an gesuchten Objekten, der sich innerhalb der Rekonstruktionszone befindet, vollständig in dem Modell enthalten ist.

Definition 1. Seien $Z \subset \mathbb{R}^n$ die Menge, welche die Rekonstruktionszone repräsentiert und $U \subset \mathbb{R}^n$ die Menge, welche die gesuchten Objekte repräsentiert, so heißt die Menge $R \subseteq Z$ *geometrische Rekonstruktion* von U in Z , wenn $R \supseteq (U \cap Z)$ gilt (Abbildung 3.1, a).

3. Modellierung

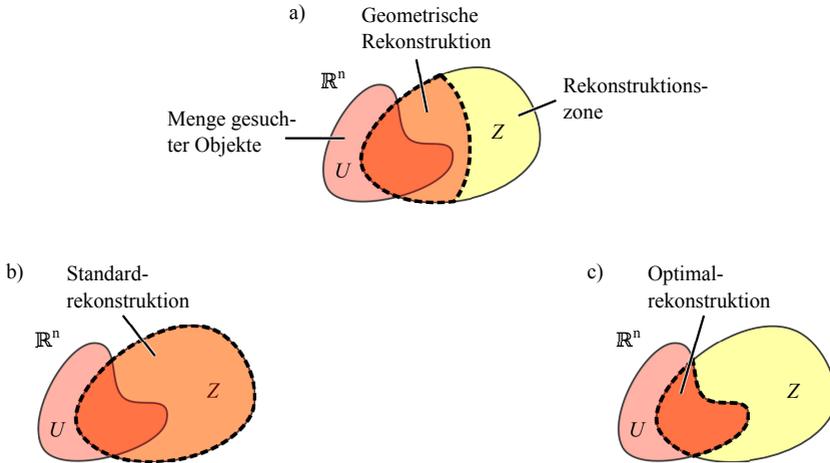


Abbildung 3.1.: Illustration einer geometrischen Rekonstruktion gesuchter Objekte $U \subset \mathbb{R}^n$ in einer Rekonstruktionszone $Z \subset \mathbb{R}^n$ im n -dimensionalen euklidischen Raum (a), der Standardrekonstruktion, welche Z entspricht (b) und der Optimalrekonstruktion, welche $U \cap Z$ entspricht (c).

Hat man nun zwei geometrische Rekonstruktionen gesuchter Objekte in einer Rekonstruktionszone, so stellt sich die Frage, welche dieser geometrischen Rekonstruktionen besser bzw. schlechter ist, oder ob beide gleich gut sind. Dies ist von Interesse, da eine bessere geometrische Rekonstruktion einer schlechteren vorgezogen wird.

Definition 2. Seien $R_1, R_2 \subseteq Z \subset \mathbb{R}^n$ zwei geometrische Rekonstruktionen von $U \subset \mathbb{R}^n$ in Z . Dann ist R_1 von *höherer Güte*, falls $R_1 \subsetneq R_2$, von *geringerer Güte*, falls $R_1 \supsetneq R_2$ und von *gleicher Güte*, falls $R_1 = R_2$ gilt.

Der Fall, dass von zwei geometrischen Rekonstruktionen weder die eine komplett in der anderen liegt noch umgekehrt wird für weitere Betrachtungen im Rahmen dieser Arbeit nicht benötigt und daher vernachlässigt. (Alternativ könnte man die Güte aber auch als Maß z.B. basierend auf dem Mengeninhalte (Lebesgue-Maß) einer geometrischen Rekonstruktion definieren, was zur Folge hätte, dass eine bessere geometrische Rekonstruktion nicht notwendigerweise in der schlechteren enthalten sein müsste. Dies würde aber auch eine Anpassung einiger folgender Definitionen nach sich ziehen.)

In Abbildung 3.1 ist die geometrische Rekonstruktion in (c) von höherer Güte als die geometrische Rekonstruktion in (a) und diese wiederum von höherer Güte als die geometrische Rekonstruktion in (b). Nun ist es möglich eine beste und eine schlechteste geometrische Rekonstruktion gesuchter Objekte in einer Rekonstruktionszone zu beschreiben. Die beste und somit auch angestrebte geometrische Rekonstruktion entspricht exakt den gesuchten Objekten innerhalb der Rekonstruktionszone, die schlechteste geometrische Rekonstruktion entspricht der Rekonstruktionszone selbst. Dazu sei wie folgt die Standard- und Optimalrekonstruktion definiert:

Definition 3. Sei $R \subseteq Z \subset \mathbb{R}^n$ eine geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z . Dann heißt R *Optimalrekonstruktion*, falls $R = U \cap Z$ (Abbildung 3.1, c) und *Standardrekonstruktion*, falls $R = Z$ gilt (Abbildung 3.1, b).

Es liegt auf der Hand, dass typischerweise nicht erkennbar ist, ob eine geometrische Rekonstruktion gleichzeitig eine Optimalrekonstruktion darstellt, da man – wie der Name schon sagt – die genaue Geometrie und den Ort der gesuchten Objekte nicht kennt.

Ohne Weiteres lässt sich immer die Standardrekonstruktion angeben, da diese in jedem Fall die gesuchten Objekte innerhalb der Rekonstruktionszone enthält. Liegt nun zusätzlich Wissen und Sensorinformation über Raumbereiche und gesuchte Objekte innerhalb der Rekonstruktionszone vor, so kann man durch geschickte Herangehensweise – ausgehend von der Standardrekonstruktion – schrittweise eine immer bessere geometrische Rekonstruktion hinsichtlich ihrer Güte erzielen. Kennt man zum Beispiel innerhalb einer Rekonstruktionszone $Z \subset \mathbb{R}^n$ die Menge $G \subseteq Z$, welche die Geometrie mehrerer Gegenstände innerhalb der Rekonstruktionszone beschreibt, und ist zudem bekannt, dass sich gesuchte Objekt $U \subset \mathbb{R}^n$ *nicht* innerhalb dieser Menge G befinden können – gilt also $G \cap U = \emptyset$ – so kann von einer geometrischen Rekonstruktion $R_1 \subseteq Z$ von U in Z die Menge G abgezogen werden, was in einer geometrischen Rekonstruktion $R_2 = R_1 \setminus G$ von U in Z mit gleicher oder höherer Güte bezüglich R_1 resultiert. Die Güte bleibt dann gleich, wenn $R_1 \cap G = \emptyset$ und damit $R_2 = R_1$ gilt.

Mit dieser Art der Integration von Wissen und Sensorinformation lassen sich also geometrische Rekonstruktionen mit gleicher oder höherer Güte nach dem Ausschlussprinzip erzeugen, was hier allgemein als *Plausibilisierung* bezeichnet wird.

Definition 4. Seien $Z \subset \mathbb{R}^n$ eine Rekonstruktionszone, $U \subset \mathbb{R}^n$ eine Menge, welche die gesuchten Objekte repräsentiert und I Platzhalter für spezifisches Wissen oder spezifische Sensorinformation (hier nicht weiter ausgeführt).

3. Modellierung

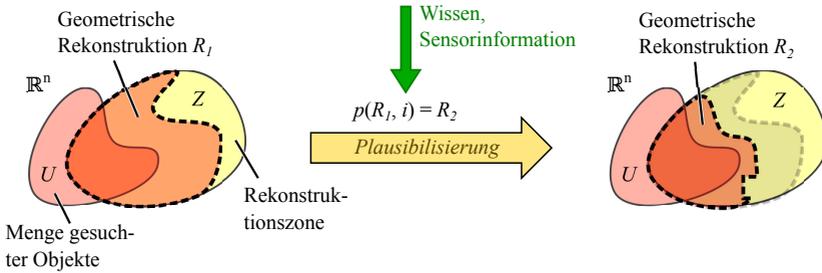


Abbildung 3.2.: Illustration der Plausibilisierung einer geometrischen Rekonstruktion R_1 mittels der Plausibilisierungsfunktion p , welche Wissen und/oder Sensorinformation i nutzt, zu einer geometrischen Rekonstruktion R_2 mit gleicher oder höherer Güte zu gelangen.

- Eine Funktion $p : 2^Z \times I \rightarrow 2^Z$ heißt *Plausibilisierungsfunktion* (Abbildung 3.2), genau dann wenn $\forall R \subseteq Z$ mit R geometrische Rekonstruktion von U in Z und $\exists i \in I$ gilt¹:
 1. $p(R, i)$ ist wieder geometrische Rekonstruktion von U in Z
 2. $p(R, i) \subseteq R$
- Das Tupel (p, i) , bestehend aus Plausibilisierungsfunktion p und spezifischer Information $i \in I$, heißt *Plausibilisierungstupel*.
- Die Menge $M \subseteq Z$ heißt *plausibel bezüglich des Plausibilisierungstupels* (p, i) , falls $p(M, i) = M$, $i \in I$ gilt. Dies gilt insbesondere auch für geometrische Rekonstruktionen.
- Sei $P = \{(p_1, i_1), \dots, (p_k, i_k)\}$ eine Menge von $k \in \mathbb{N}$ Plausibilisierungstupeln, dann heißt die Menge $M \subseteq Z$ *plausibel bezüglich der Menge P* , falls $\forall (p, i) \in P : p(M, i) = M$ gilt.

Plausibilisierungstupel ermöglichen damit die Nutzung von Wissen und Sensorinformation, um aus einer gegebenen geometrischen Rekonstruktion, eine geometrische Rekonstruktion mit gleicher oder höherer Güte zu erzeugen. Plausibilisierungsfunktionen betrachten dabei zumeist – wie sich später in den Beispielen zeigen wird (Kapitel 3.2) – entweder die einzelnen Elemente der gegebenen geometrischen Rekonstruktion oder alle größten zusammenhängenden Teilmengen der gegebenen geometrischen Rekonstruktion, welche hier als Regionen bezeichnet werden (Abbildung 3.3):

¹ 2^Z bezeichnet die Potenzmenge von Z .

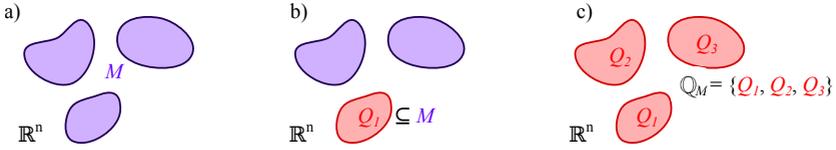


Abbildung 3.3.: Illustration des Begriffs der Region: In a) ist die Menge $M \subset \mathbb{R}^n$ dargestellt, in b) die Region $Q_1 \subseteq M$ und in c) die Menge aller Regionen $\mathbb{Q}_M = \{Q_1, Q_2, Q_3\}$.

Definition 5. Sei $M \subset \mathbb{R}^n$ eine Menge.

- Eine *Region der Menge M* (Abbildung 3.3, b) ist eine zusammenhängende Menge $Q_1 \subseteq M$ mit der Eigenschaft Q_1 maximal, das heißt $\forall Q_2 \subseteq M$, mit Q_2 zusammenhängend, gilt
 1. $Q_2 \subseteq Q_1$ oder
 2. $Q_2 \cap Q_1 = \emptyset$
- Eine Menge $\mathbb{Q}_M = \{Q_1 \subseteq M \mid Q_1 \text{ Region von } M\}$ heißt *Menge aller Regionen von M* (Abbildung 3.3, c).

Ein weiteres Merkmal, welches die meisten Plausibilisierungsfunktionen erfüllen, ist, dass egal ob sie nun auf Regionen oder den einzelnen Elementen basieren, sie dies stets unabhängig von allen anderen Regionen bzw. Elementen der gegebenen geometrischen Rekonstruktion tun (Abbildung 3.4). Dahingehend werden nun Plausibilisierungsfunktionen wie folgt charakterisiert:

Definition 6. Seien $Z \subset \mathbb{R}^n$ eine Rekonstruktionszone und (p, i) ein Plausibilisierungstupel.

- p heißt *elementbasiert*, wenn folgende zwei Bedingungen gelten:
 1. $p\left(\bigcup_{m \in M} \{m\}, i\right) = \bigcup_{m \in M} p(\{m\}, i), \forall M \subseteq Z$ (Additivität)
 2. $p(\{z\}, i) \in \{\{z\}, \emptyset\}, \forall z \in Z$ (Vernichtung)
- p heißt *regionenbasiert*, wenn folgende zwei Bedingungen gelten (Abbildung 3.4):
 1. $p\left(\bigcup_{\substack{Q \subseteq M \\ \text{Region}}} Q, i\right) = \bigcup_{\substack{Q \subseteq M \\ \text{Region}}} p(Q, i), \forall M \subseteq Z$ (Additivität)
 2. $p(Q, i) \in \{Q, \emptyset\}, \forall Q \subseteq Z \text{ Region}$ (Vernichtung)

3. Modellierung

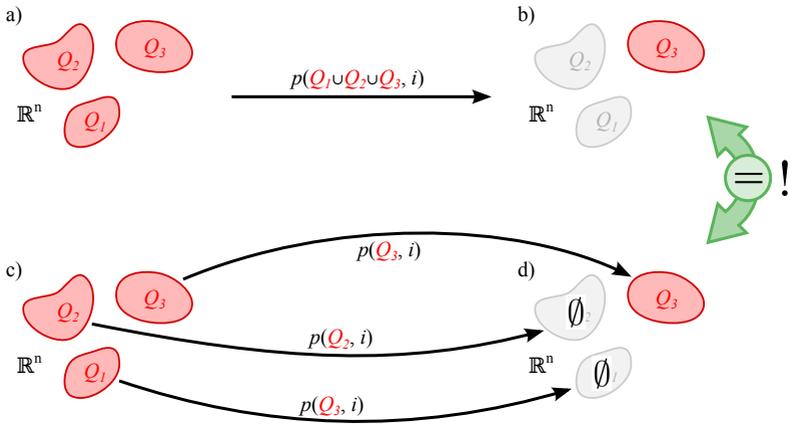


Abbildung 3.4.: Illustration der Definition *regionenbasiert*, die im Aufbau auch der Definition *elementbasiert* gleicht. Eine Plausibilisierungsfunktion wird *regionenbasiert* bezeichnet, wenn deren Anwendung auf die Ausgangsmenge $Q_1 \cup Q_2 \cup Q_3$ (a) zum gleichen Ergebnis (b) führt, wie die Vereinigung (d) deren Anwendung auf die einzelnen Regionen Q_1 , Q_2 und Q_3 (c) – “Additivität”. Zudem darf die Anwendung auf eine einzelne Region entweder die Region selbst oder die leere Menge als Ergebnis haben (d) – “Vernichtung”.

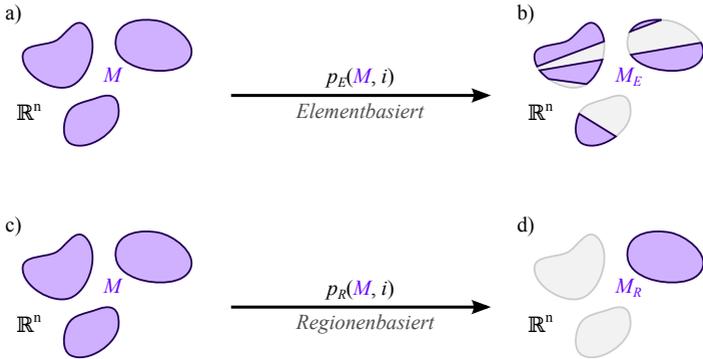


Abbildung 3.5.: Illustration einer beispielhaften elementbasierten (a, b) und einer regionenbasierten Plausibilisierungsfunktion (c, d): Während die elementbasierten Plausibilisierungsfunktion einzelne Elemente aus der Menge M eliminiert (b), werden bei der regionenbasierten Plausibilisierungsfunktion nur einzelne Regionen von M eliminiert (d).

Der Unterschied zwischen elementbasierter und regionenbasierter Plausibilisierungsfunktion ist beispielhaft in Abbildung 3.5 dargestellt.

Erfüllt eine Funktion eines dieser beiden Merkmale – also elementbasiert oder regionenbasiert –, so ist leicht ersichtlich, dass die erzeugte geometrische Rekonstruktion eine Teilmenge der eingegebenen geometrischen Rekonstruktion sein muss, was gemäß der Definition einer Plausibilisierungsfunktion (Definition 4) gefordert ist.

Des Weiteren haben Plausibilisierungstupel meistens in einem *einzigem* “Schritt” eine geometrische Rekonstruktion als Ausgabe, welche bereits plausibel bezüglich dieses Plausibilisierungstupels ist. Dies ist nicht immer der Fall, weil es denkbar ist, dass die Plausibilisierungsfunktion erst bei wiederholter Anwendung zu einem Ergebnis führt, welches plausibel bezüglich des Plausibilisierungstupels ist. Folgende Definition beschreibt eine Eigenschaft einer Plausibilisierungsfunktion, die diese dahingehend auszeichnet:

Definition 7. Sei $Z \subset \mathbb{R}^n$ eine Rekonstruktionszone und (p, i) ein Plausibilisierungstupel. Die Plausibilisierungsfunktion p heißt *vollständig*, wenn $p(M, i) = p(p(M, i), i) \forall M \subseteq Z$ gilt.

Ist eine Plausibilisierungsfunktion vollständig, so ist ihr Ergebnis immer plausibel bezüglich des zugehörigen Plausibilisierungstupels. Dieser Erkenntnis liegt folgende Überlegung zugrunde: Aus der Definition der Vollständigkeit $p(p(M, i), i) = p(M, i)$ und der Annahme, dass es ein $N \subseteq Z$ gibt, mit $p(M, i) = N$, folgt:

$$\begin{aligned} p(p(M, i), i) = p(M, i) &\Leftrightarrow p(N, i) = p(M, i) \\ &\Leftrightarrow p(N, i) = N \end{aligned}$$

Eine typische Eigenschaft regionenbasierter Plausibilisierungsfunktionen ist die, dass alle Teilregionen einer als nicht-plausibel ermittelten Region ebenfalls nicht plausibel sind:

Definition 8. Sei $Z \subset \mathbb{R}^n$ eine Rekonstruktionszone und (p, i) ein Plausibilisierungstupel. Die Plausibilisierungsfunktion p heißt

- *Teilregionen-verwerfend für die Menge $M \subseteq Z$* , wenn $\forall Q \subseteq M$ mit Q Region, die folgende Inklusion gilt: $p(Q, i) = \emptyset \Rightarrow \forall R \subseteq Q : p(R, i) = \emptyset$
- *Teilregionen-verwerfend*, wenn p Teilregionen-verwerfend $\forall M \subseteq Z$.

3. Modellierung

Eine Plausibilisierungsfunktion kapselt typischerweise nur einen Aspekt des gesamten verfügbaren Wissens und der Sensorinformation über gesuchte Objekte. Daher setzt man mehrere solcher Plausibilisierungsfunktionen iterativ ein, um so schrittweise die Güte ausgehend von einer Standardrekonstruktion zu erhöhen. Dabei spielt die Anwendungsreihenfolge der einzelnen Plausibilisierungsfunktionen eine Rolle, da sie zu unterschiedlichen Ergebnissen führen kann. Wünschenswert ist eine solche (wiederholungsfreie) Anwendungsreihenfolge, deren Ergebnis sich nicht mehr verbessern lässt:

Definition 9. Seien $Z \subset \mathbb{R}^n$ eine Rekonstruktionszone und $P = \{(p_1, i_1), \dots, (p_k, i_k)\}$ eine Menge von $k \in \mathbb{N}$ Plausibilisierungstupeln. Eine Permutation

$$\left(\begin{array}{cccc} (p_1, i_1) & (p_2, i_2) & \dots & (p_k, i_k) \\ (p_{\sigma(1)}, i_{\sigma(1)}) & \sigma(p_{\sigma(2)}, i_{\sigma(2)}) & \dots & \sigma(p_{\sigma(k)}, i_{\sigma(k)}) \end{array} \right) \in S(P)$$

heißt *Plausibilisierungspermutation*, wenn $p_{\sigma(k)}(\dots(p_{\sigma(2)}(p_{\sigma(1)}(R, i_{\sigma(1)}), i_{\sigma(2)}), \dots), i_{\sigma(k)})$, $\forall R \subseteq Z$ geometrische Rekonstruktion plausibel bezüglich der Menge P ist. $S(P)$ ist die symmetrische Gruppe der Menge P und enthält damit sämtliche Permutationen.

Alle im späteren Verlauf vorgestellten Plausibilisierungsfunktionen lassen sich in zwei unterschiedliche Klassen einteilen. Im Folgenden werden nun diese zwei Klassen beschrieben, danach wird gezeigt, dass Plausibilisierungsfunktionen innerhalb einer jeden Klasse beliebig permutierbar sind und schließlich, dass die resultierende geometrische Rekonstruktion plausibel bezüglich aller Plausibilisierungstupel ist, wenn eine der beiden Klassen vor der anderen angewendet wird.

Definition 10. Plausibilisierungsfunktionen, bzw. Plausibilisierungstupel, welche gleichzeitig elementbasiert und vollständig sind, werden kurz als *vom Typ E* bezeichnet. Plausibilisierungsfunktionen bzw. Plausibilisierungstupel, welche gleichzeitig regionenbasiert, Teilregionen-verwerfend und vollständig sind, werden kurz als *vom Typ R* bezeichnet.

Alle später vorgestellten Plausibilisierungsfunktionen sind entweder vom Typ E oder vom Typ R.

Satz 1. Seien $P = \{(p_1, i_1), \dots, (p_k, i_k)\}$ eine Menge von $k \in \mathbb{N}$ Plausibilisierungstupeln, dann gelten für die beiden Mengen $P_E = \{(p_x, i_x) \in P \mid p_x \text{ vom Typ E}\}$ und $P_R = \{(p_x, i_x) \in P \mid p_x \text{ vom Typ R}\}$ die folgenden Inklusionen:

1. $\sigma \in S(P_E) \Rightarrow \sigma$ Plausibilisierungspermutation,
2. $\sigma \in S(P_R) \Rightarrow \sigma$ Plausibilisierungspermutation und
- 3.

$$\begin{pmatrix} (p_1, i_1) & \cdots & (p_{|P_E|}, i_{|P_E|}) \\ (p_{\sigma_E(1)}, i_{\sigma_E(1)}) & \cdots & (p_{\sigma_E(|P_E|)}, i_{\sigma_E(|P_E|)}) \end{pmatrix} \in S(P_E),$$

$$\begin{pmatrix} (q_1, j_1) & \cdots & (q_{|P_R|}, j_{|P_R|}) \\ (q_{\sigma_R(1)}, j_{\sigma_R(1)}) & \cdots & (q_{\sigma_R(|P_R|)}, j_{\sigma_R(|P_R|)}) \end{pmatrix} \in S(P_R)$$

↓

$$\begin{pmatrix} (p_1, i_1) & \cdots & (p_{|P_E|}, i_{|P_E|}) & (q_1, j_1) & \cdots & (q_{|P_R|}, j_{|P_R|}) \\ (p_{\sigma_E(1)}, i_{\sigma_E(1)}) & \cdots & (p_{\sigma_E(|P_E|)}, i_{\sigma_E(|P_E|)}) & (q_{\sigma_R(1)}, j_{\sigma_R(1)}) & \cdots & (q_{\sigma_R(|P_R|)}, j_{\sigma_R(|P_R|)}) \end{pmatrix}$$

$\in S(P_E \cup P_R)$ ebenfalls Plausibilisierungspermutation.

Jede Permutation einer Menge von Plausibilisierungstupeln ausschließlich vom Typ E ist eine Plausibilisierungspermutation, und jede Permutation einer Menge von Plausibilisierungstupeln ausschließlich vom Typ R ist eine Plausibilisierungspermutation. Jede Permutation einer Menge Plausibilisierungstupel vom Typ E und vom Typ R ist dann eine Plausibilisierungspermutation, wenn die Permutation zunächst die Plausibilisierungstupel vom Typ E enthält und danach erst die vom Typ R.

Beweis. 1. Beispielhaft für zwei Plausibilisierungstupel (p_1, i_1) und (p_2, i_2) vom Typ E wird gezeigt, dass diese beiden Tupel permutierbar sind, d.h. $p_1(p_2(R, i_2), i_1) = p_2(p_1(R, i_1), i_2)$ für alle geometrischen Rekonstruktionen $R \subseteq Z$ der Rekonstruktionszone $Z \subset \mathbb{R}^n$. Außerdem wird gezeigt, dass das Ergebnis plausibel bezüglich beider Plausibilisierungstupel ist, d.h. $p_1(p_2(R, i_2), i_1)$ bzw. $p_2(p_1(R, i_1), i_2)$ plausibel bezüglich (p_1, i_1) und (p_2, i_2) . Sei daher $R \subseteq Z$ eine beliebige geometrische Rekonstruktion, aber fest.

- Permutierbarkeit:

Siehe Tabelle 3.1.

- $p_1(p_2(R, i_2), i_1)$ plausibel bezüglich p_1 , d.h. zu zeigen ist

$$p_1(p_1(p_2(R, i_2), i_1), i_1) = p_1(p_2(R, i_2), i_1)$$

3. Modellierung

$p_1(\{r_{i_1}\}, i_1)$	$p_2(\{r_{i_2}\}, i_2)$	$p_1(p_2(\{r_{i_1}\}, i_2), i_1)$	$p_2(p_1(\{r_{i_2}\}, i_1), i_2)$
\emptyset	\emptyset	\emptyset	\emptyset
\emptyset	$\{r_{i_1}\}$	\emptyset	\emptyset
$\{r_{i_1}\}$	\emptyset	\emptyset	\emptyset
$\{r_{i_1}\}$	$\{r_{i_1}\}$	$\{r_{i_1}\}$	$\{r_{i_1}\}$

Tabelle 3.1.: Wegen der Vernichtungseigenschaft in der Definition zu elementbasierter Plausibilisierungsfunktionen: Liefert eine der beiden einzelnen Plausibilisierungsfunktionen die leere Menge, so tun sie das auch in der Verkettung $p_1 \circ p_2$ bzw. $p_2 \circ p_1$, was der kommutativen Und-Verknüpfung entspricht.

Gemäß der Definition zur Vollständigkeit gilt $p(p(M, i), i) = p(M, i) \forall M \subseteq Z$. Insbesondere gilt dies für $(p, i) = (p_1, i_1)$ und $M = p_2(R, i_2)$, was zu zeigen war.

- $p_1(p_2(R, i_2), i_1)$ plausibel bezüglich p_2 , d.h. zu zeigen ist

$$p_2(p_1(p_2(R, i_2), i_1), i_2) = p_1(p_2(R, i_2), i_1)$$

$$\begin{aligned} p_2(p_1(p_2(R, i_2), i_1), i_2) &= p_2(p_2(p_1(R, i_1), i_2), i_2) \quad (\text{wegen Permutierbarkeit}) \\ &= p_2(p_1(R, i_1), i_2) \end{aligned}$$

Gemäß der Definition zur Vollständigkeit gilt $p(p(M, i), i) = p(M, i) \forall M \subseteq Z$. Insbesondere gilt dies für $(p, i) = (p_2, i_2)$ und $M = p_1(R, i_1)$, was zu zeigen war.

2. Beispielhaft für zwei Plausibilisierungstupel (p_1, i_1) und (p_2, i_2) vom Typ R wird gezeigt, dass diese beiden Tupel permutierbar sind, d.h. $p_1(p_2(R, i_2), i_1) = p_2(p_1(R, i_1), i_2)$ für alle geometrischen Rekonstruktionen $R \subseteq Z$ der Rekonstruktionszone $Z \subset \mathbb{R}^n$. Außerdem wird gezeigt, dass das Ergebnis plausibel bezüglich beider Plausibilisierungstupel ist, d.h. $p_1(p_2(R, i_2), i_1)$ bzw. $p_2(p_1(R, i_1), i_2)$ plausibel bezüglich (p_1, i_1) und (p_2, i_2) . Sei daher $R \subseteq Z$ eine beliebige geometrische Rekonstruktion, aber fest.

Analog zum Beweis bezüglich Typ E, jedoch mit Regionen.

3. Beispielhaft für zwei Plausibilisierungstupel (p_E, i_E) vom Typ E und (p_R, i_R) vom Typ R wird gezeigt, dass das Plausibilisierungstupel vom Typ E vor dem Plausibilisierungstupel

vom Typ R angewendet werden muss, damit das Resultat plausibel bezüglich beider Tupel ist, d.h. $p_R(p_E(R, i_E), i_R)$ ist plausibel bezüglich (p_E, i_E) und (p_R, i_R) für alle geometrischen Rekonstruktionen $R \subseteq Z$ der Rekonstruktionszone $Z \subset \mathbb{R}^n$. Sei daher $R \subseteq Z$ eine beliebige geometrische Rekonstruktion, aber fest.

- $p_R(p_E(R, i_E), i_R)$ ist plausibel bezüglich (p_R, i_R) , d.h. zu zeigen ist

$$p_R(p_R(p_E(R, i_E), i_R), i_R) = p_R(p_E(R, i_E), i_R)$$

Gemäß der Definition zur Vollständigkeit gilt $p(p(M, i), i) = p(M, i) \forall M \subseteq Z$. Insbesondere gilt dies für $(p, i) = (p_R, i_R)$ und $M = p_E(R, i_E)$, was zu zeigen war.

- $p_R(p_E(R, i_E), i_R)$ ist plausibel bezüglich (p_E, i_E) , d.h. zu zeigen ist:

$$p_E(p_R(p_E(R, i_E), i_R), i_E) = p_E(p_E(R, i_E), i_R)$$

Zunächst wird gezeigt, dass eine Teilmenge $R_1 \subseteq R$ einer bezüglich (p_E, i_E) plausiblen Rekonstruktion R ebenfalls plausibel bezüglich (p_E, i_E) ist, und daher folgende Inklusion gilt, wobei $R = R_1 \cup R_2$:

$$(p_E(R_1 \cup R_2, i_E) = R_1 \cup R_2) \Rightarrow (p_E(R_1, i_E) = R_1)$$

Angenommen R_1 sei nicht plausibel bezüglich (p_E, i_E) , so hieße dies, dass es ein Element in R_1 gibt, welches als leere Menge ausgewertet wird: $p_E(R_1, i_E) \subsetneq R_1 \Leftrightarrow \exists r \in R_1 : p_E(\{r\}, i_E) = \emptyset$

Da aber R plausibel bezüglich (p_E, i_E) gilt, müssen alle Elemente in R wieder als solches ausgewertet werden: $p_E(R_1 \cup R_2, i_E) = R \Rightarrow \forall r \in (R_1 \cup R_2) = \{r\}$. Dies stellt einen Widerspruch dar. Insofern muss auch die Teilmenge R_1 plausibel bezüglich (p_E, i_E) sein.

Weil nun $p_R(p_E(R, i_E), i_R) \subseteq p_E(R, i_E)$ und $p_E(p_E(R, i_E), i_E) = p_E(R, i_E)$ (Vollständigkeit) gilt, gilt auch

$$p_E(p_R(p_E(R, i_E), i_R), i_E) = p_E(p_E(R, i_E), i_E)$$



Folgendes Kapitel beschreibt die unterschiedlichsten Plausibilisierungsfunktionen und untersucht deren Eigenschaften.

3.2. Plausibilisierungsfunktionen

Nachdem im vorangegangenen Kapitel das Prinzip der geometrischen Rekonstruktion und der Plausibilisierung eingeführt wurde, werden nun konkrete und praxisrelevante Plausibilisierungsfunktionen vorgestellt, die im nachfolgenden Kapitel 3.3 im Anwendungsbeispiel *Mensch/Roboter-Koexistenz* angewandt werden. Die einzelnen Plausibilisierungsfunktionen werden im Folgenden immer nach dem gleichen Schema beschrieben: Zunächst wird motiviert, dann formalisiert, schließlich einem der beiden Typen E bzw. R zugeordnet und gegebenenfalls weiter erörtert.

3.2.1. Ausschließbarer Raumbereich

Oftmals existieren in einer Rekonstruktionszone $Z \subset \mathbb{R}^n$ Raumbereiche $A \subseteq Z$, an deren Stelle sich keine gesuchten Objekte $U \subset \mathbb{R}^n$ aufhalten können, beispielsweise weil sich dort ein solider Gegenstand (z.B. Regal, Roboter, Tisch, ...) befindet, weil es keinen Zugang zu diesem Bereich gibt (z.B. unzugänglicher Käfig) oder weil ein Sensor die Abwesenheit von gesuchten Objekten garantiert (z.B. via Kamera, Anwesenheitssensor, Näherungssensor, ...). Für gesuchte Objekte U in Bezug auf diese Bereiche A gilt damit also $U \cap A = \emptyset$. Aus diesem Grund ist eine geometrische Rekonstruktion $R \subseteq Z$ von U in Z abzüglich A immer noch eine geometrische Rekonstruktion.

Definition 11. Sei $R \subseteq Z \subset \mathbb{R}^n$ eine geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z , ausserdem sei $A \subseteq Z$ ein ausschließbarer Raumbereich – gilt also $U \cap A = \emptyset$, dann ist $p_a : 2^Z \times 2^Z \rightarrow 2^Z$, $p_a(R, A) := R \setminus A$ eine Plausibilisierungsfunktion und heißt *Ausschließbarkeitsfunktion* (Abbildung 3.6).

Alle nicht-plausiblen Elemente der gegebenen geometrischen Rekonstruktion R , also alle Elemente, an denen sich kein gesuchtes Objekt aufhalten kann, werden durch diese Plausibilisierungsfunktion verworfen.

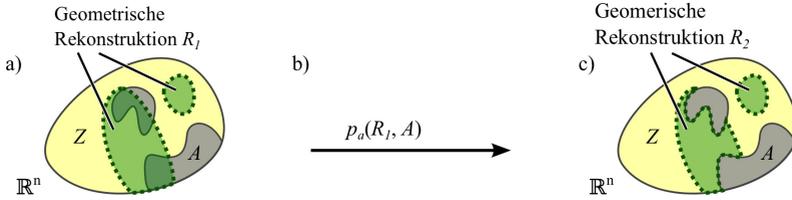


Abbildung 3.6.: Illustration der Ausschließbarkeitsfunktion. Die geometrische Rekonstruktion R_1 (a) wird mittels der Ausschließbarkeitsfunktion (b) in eine geometrische Rekonstruktion R_2 (c) überführt. Die von der auszuschließenden Menge A überlappenden Bereiche der geometrischen Rekonstruktion R_1 werden dabei verworfen, da sich hier keine gesuchten Objekte aufhalten können.

Es folgt nun die Zuordnung der Plausibilisierungsfunktion zu einem der beiden Typen gemäß Definition 10. Es handelt sich hierbei um eine *elementbasierte* Plausibilisierungsfunktion, da allgemein $\bigcup_{r \in R} (\{r\} \setminus A) = R \setminus A$, mit $R, A \subseteq Z \subset \mathbb{R}^n$ gilt: Die Vereinigung aller ein-elementigen Mengen aus einer geometrischen Rekonstruktion, welche nicht im ausschließbaren Raumbereich liegen entspricht genau der geometrischen Rekonstruktion ohne den ausschließbaren Bereich. Für jedes $r \in R$ kann einzeln entschieden werden, ob es zur auszugebenden geometrischen Rekonstruktionen hinzugehört. Des Weiteren ist diese Plausibilisierungsfunktion *vollständig*, da $R \setminus A = (R \setminus A) \setminus A$ gilt. Es handelt sich also um eine Plausibilisierungsfunktion vom *Typ E* (Definition 10).

Im Falle eines soliden Gegenstandes oder eines unzugänglichen Bereiches lässt sich A direkt angeben. Im folgenden Kapitel 3.2.2 wird gezeigt, wie A im konkreten Fall eines Anwesenheitssensors, einer Farb- und einer Tiefenkamera angegeben werden kann.

3.2.2. Sensorbasierter ausschließbarer Raumbereich

In diesem Kapitel wird anhand der drei Sensoren *Anwesenheitssensor*, *Farb-* und *Tiefenkamera* gezeigt, wie aus deren Auswertung ein ausschließbarer Raumbereich gewonnen werden kann. Dieser kann dann gemäß Kapitel 3.2.1 direkt in der Ausschließbarkeitsfunktion eingesetzt werden. Insofern stellt dieses Kapitel vielmehr eine Erweiterung des vorherigen, als die Angabe einer neuen, weiteren Plausibilisierungsfunktion dar.

3.2.2.1. Anwesenheitssensor

Als einführendes, besonders einfaches Beispiel dient der Anwesenheitssensor. Dieser beschreibt für einen bestimmten Raumbereich die An- bzw. Abwesenheit von gesuchten Objekten. Ein solches Verhalten ist durch folgende Anwesenheitsfunktion definiert:

Definition 12. Seien $U \subset \mathbb{R}^n$ die Menge der gesuchten Objekte und $W \subseteq Z \subset \mathbb{R}^n$ eine Teilmenge einer Rekonstruktionszone Z , so heisst

$$a : 2^{\mathbb{R}^n} \times 2^Z \rightarrow \{0, 1\}, \quad a(U, W) := \begin{cases} 0 & , U \cap W = \emptyset \text{ gemessen} \\ 1 & , U \cap W \neq \emptyset \text{ gemessen} \end{cases}$$

Anwesenheitsfunktion für U in W .

Die Anwesenheitsfunktion liefert den Wert 1 zurück, wenn im zugehörigen Detektionsbereich W ein gesuchtes Objekt U gemessen wird. Ansonsten liefert sie den Wert 0 zurück.

Insofern lässt sich dieser Raumbereich genau dann ausschließen, wenn der Anwesenheitssensor die Abwesenheit eines gesuchten Objektes garantieren kann:

Definition 13. Seien $U \subset \mathbb{R}^n$ die Menge der gesuchten Objekte, $W \subseteq Z \subset \mathbb{R}^n$ eine Teilmenge einer Rekonstruktionszone Z und a eine Anwesenheitsfunktion für U in W , dann ist

$$A_A := \begin{cases} W & , a(U, W) = 0 \\ \emptyset & , a(U, W) \neq 0 \end{cases}$$

ausschließbarer Raumbereich eines Anwesenheitssensors.

Liefert die dem Anwesenheitssensor zugrunde liegende Anwesenheitsfunktion a den Wert 0 zurück, so heißt dies, dass im zugehörigen Detektionsbereich W keine gesuchten Objekte U gemessen werden und somit W als ausschließbarer Raumbereich A_A (des Anwesenheitssensors) in der Ausschließbarkeitsfunktion (Definition 11) verwendet werden kann.

Das SafetyEye der Firma Pilz [Pilz 11] stellt im Grunde einen solchen Anwesenheitssensor dar, da dieses das Betreten von Zonen detektiert und signalisiert. Eine einzelne solche Zone entspricht in obiger Definition damit dem W . Ähnlich zu diesem einfachen Beispiel, wird im Folgenden die Verwendung von Farb- und Tiefenkameras zur Ermittlung eines ausschließbaren Raumbereiches erläutert.

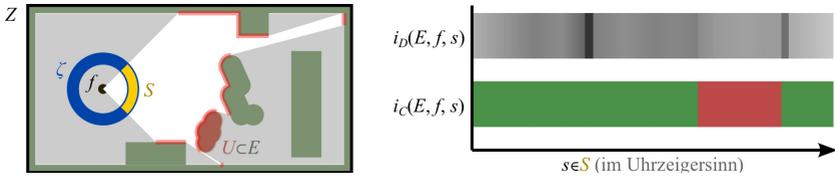


Abbildung 3.7.: Darstellung einer kalibrierten Kamera (f, S) , bestehend aus einem Projektionszentrum $f \in \mathbb{R}^n$ und den Sichtrichtungen $S \subseteq \zeta$ (gelb), in einer Umwelt $E \subseteq \mathbb{R}^n$, bestehend aus gesuchten Objekten $U \subseteq \mathbb{R}^n$ (rot) und bekannten Gegenständen $E \setminus U$ (grün). Im rechten Bild sind die resultierende Tiefenkamera-Pixelfunktion i_D und die resultierende Farbkamera-Pixelfunktion i_C "entlang" aller $s \in S$ "im Uhrzeigersinn" aufgetragen. Die Tiefenwerte der Tiefenkamera-Pixelfunktion sind durch Grauwerte beschrieben, wobei ein heller Grauwert eine kürzere Distanz als ein dunkler Grauwert bedeutet.

3.2.2.2. Kameras

Tiefenkameras liefern für jedes Pixel einen Wert, der die kürzeste Distanz zwischen der Kamera und allen Gegenständen, welche auf der Rückprojektion des Pixels liegen, beschreibt. Farbkameras liefern bekanntermaßen für jedes Pixel einen Farbwert, der sich aus der Beleuchtungssituation und Materialbeschaffenheit der Gegenstände auf der Rückprojektion des Pixels ergibt. Eine detailliertere Analyse und Beschreibung der physikalischen Zusammenhänge ist im Rahmen dieser Arbeit nicht notwendig. Eine Grauwert-Kamera kann als Spezialfall der Farbkamera betrachtet werden und wird daher nicht gesondert behandelt.

Damit es überhaupt möglich ist Aussagen über Raumbereiche treffen zu können, muss die Kamera mit dem betrachteten Raum in Beziehung gesetzt werden. Daher wird zunächst die kalibrierte Kamera eingeführt (Abbildung 3.7):

Definition 14. Sei $\zeta := \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$ die Oberfläche der Einheitskugel,

- dann ist das Tupel $K := (f, S)$, mit $f \in \mathbb{R}^n$ einem Projektionszentrum, und $S \subseteq \zeta$ den Sichtrichtungen, eine *kalibrierte Kamera*.
- Die Elemente $s \in S$ heißen *Pixel*.

Gegenüber realer Kamerahardware, werden Pixel in diesem Kapitel nicht als diskret definiert. Dies ist naheliegend, nachdem die Betrachtungen zur geometrische Rekonstruktion ebenfalls

3. Modellierung

bereits im nicht-diskreten \mathbb{R}^n liegen. Eine Diskretisierung von Bild und Raum erfordert dagegen zusätzliche, spezielle Untersuchungen gefolgt von Maßnahmen, die gewährleisten, dass die geometrische Rekonstruktion am Ende tatsächlich konservativ ist – also keine gesuchten Objekte aus dieser herausragen. Solche speziellen Untersuchungen müssen hier nicht angestellt werden und vereinfachen damit die Betrachtungen der geometrischen Zusammenhänge. Sie werden allerdings in der Umsetzung in Kapitel 4 ausführlich thematisiert.

Die Werte eines Pixels hängen vom Kameratyp ab. Eine Tiefenkamera liefert pro Pixel einen positiven skalaren Distanz-Wert, wohingegen eine Farbkamera einen Wert aus der Menge ihrer unterscheidbaren Farben liefert. Der Wert lässt sich mittels einer typabhängigen Pixelfunktion beschreiben:

Definition 15. Seien $E \subset \mathbb{R}^n$ die Menge aller Gegenstände und $K = (f, S)$ eine kalibrierte Kamera, dann ist

- $i : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \overline{\mathbb{R}}^+$ eine *tiefenbeschreibende Pixelfunktion* und
- $i_D : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \overline{\mathbb{R}}^+$, $i_D(E, f, s) := \inf\{x \in \overline{\mathbb{R}}^+ \mid f + x \cdot s \in E\}$, mit $\inf \emptyset := \infty$ eine konkrete tiefenbeschreibende Pixelfunktion, welche als *Tiefenkamera-Pixelfunktion* bezeichnet wird. Sie beschreibt den Wert – in diesem Fall also die Tiefe – eines Pixels einer Tiefenkamera. (Abbildung 3.7).

Der Wert eines Pixels einer Tiefenkamera ist damit die kleinste Distanz zwischen der Kamera und allen Gegenstands-Punkten entlang der Pixel-Rückprojektion. Enthält die Pixel-Rückprojektion jedoch keinen Gegenstands-Punkt sieht die Kamera bis ins Unendliche. Allgemein wird eine Funktion deren Wert pro Pixel eine “Sichttiefe” beschreibt hier als tiefenbeschreibende Pixelfunktion bezeichnet.

Die Pixelwerte einer Farbkamera stellen keine tiefenbeschreibende Pixelfunktion dar und werden wie folgt definiert:

Definition 16. Seien $E \subset \mathbb{R}^n$ die Menge aller Gegenstände, $K = (f, S)$ eine kalibrierte Kamera und C die Menge der von dieser Kamera unterscheidbaren Farben, dann ist $i_C : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow C$, $i_C(E, f, s) := c$ die *Farbkamera-Pixelfunktion*, welche den Wert – in diesem Fall also die Farbe – eines Pixels einer Farbkamera beschreibt (Abbildung 3.7). Ohne auf physikalische Details einzugehen, ergibt sich der Farbwert aus der Beleuchtungssituation und der Materialbeschaffenheit der Gegenstände in der Rückprojektion des Pixels.

Vergleicht man die Tiefenkamera-Pixelfunktion mit der Farbkamera-Pixelfunktion, so stellt man fest, dass bei Tiefenkameras bereits eine geometrische Information vorliegt, welche man in direkter Weise zur Beschreibung eines ausschließbaren Raumbereiches nutzen kann:

Definition 17. Seien $E \subset \mathbb{R}^n$ die Menge aller Gegenstände, $K = (f, S)$ eine kalibrierte Kamera und $Z \subset \mathbb{R}^n$ eine Rekonstruktionszone. Der *ausschließbare Raumbereich einer Tiefenkamera* wird beschrieben durch

$$A_D^K := \{z \in Z \mid \exists s \in S \exists x \in \mathbb{R}^+ : z = f + x \cdot s \text{ mit } x < i_D(E, f, s)\}$$

(Abbildung 3.8, a, c, e, g, i).

Alle Raumelemente die auf dem rückprojizierten Strahl eines Pixels liegen und zudem näher an der Kamera liegen als der nächstliegende Punkt der Umwelt – bestimmt mittels der Tiefenkamera-Pixelfunktion – zählen zum ausschließbaren Raumbereich der Tiefenkamera.

Im Fall einer Farbkamera muss man jedoch einen Umweg gehen, um mithilfe der Farbwerte dennoch zu einer geometrischen Aussage zu gelangen. Zwei solche Umwege berücksichtigen die Geometrie einer bekannten Umwelt sowie das Wissen, entweder über die Farben der gesuchten Objekte oder deren farbliche Unterscheidbarkeit bezüglich eines Hintergrundmodells der bekannten Umwelt, und werden im Folgenden beschrieben.

Betrachtet man nun den ersten Fall, in welchem die Farben gesuchter Objekte im Bild einer Farbkamera bekannt sind. Sei dazu die Menge aller Farben der gesuchten Objekte wie folgt definiert:

Definition 18. Sei $U \subset \mathbb{R}^n$ die Menge der gesuchten Objekte, $K = (f, S)$ die kalibrierte Kamera und C die Menge der von dieser Kamera unterscheidbaren Farben, so ist $F_U := \{i_C(U, f, s) \in C \mid s \in S\}$ die *Menge aller Farben gesuchter Objekte U* .

Kennt man nun also die Menge aller Farben gesuchter Objekte, so lässt sich ein Pixel zu einem potentiell gesuchten Objekt P bzw. nicht-gesuchten Objekt L zuordnen:

Definition 19. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera und $F_U \subseteq C$ die Menge aller Farben gesuchter Objekte U aus der Menge der von dieser Kamera unterscheidbaren Farben

3. Modellierung

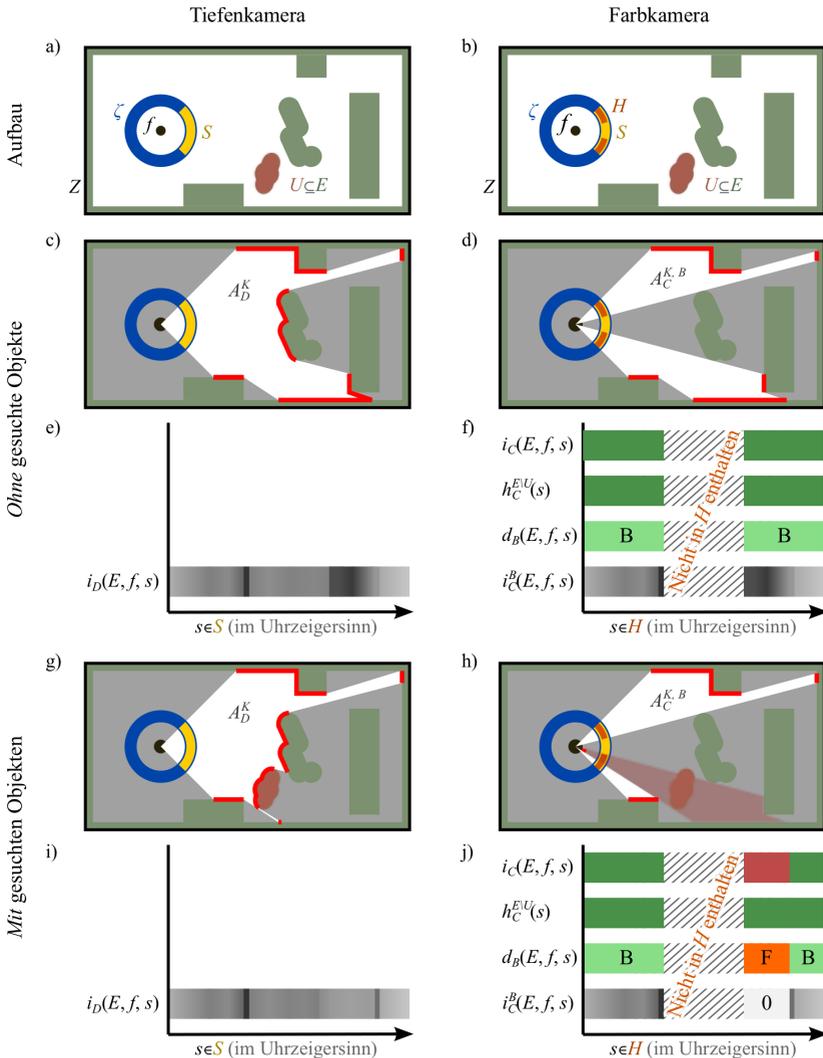


Abbildung 3.8.: Darstellung der Zusammenhänge der Pixelfunktionen und des ausschließbaren Raumbereiches einer Tiefenkamera (a, c, e, g, i) und einer Farbkamera mittels eines Hintergrundmodells (b, d, f, h, j) in einer gegebenen Umwelt. Aufbau der Beispielszene (a, b); Ausschließbarer Raumbereich ohne gesuchte Objekte (c - f); Ausschließbarer Raumbereich mit gesuchten Objekten (g - j).

C , dann heißt $d_F : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \{P, L\}$,

$$d_F(E, f, s) := \begin{cases} P & , i_C(E, f, s) \in F_U \\ L & , i_C(E, f, s) \notin F_U \end{cases}$$

farbbasierte Segmentierung mittels F_U .

Wird ein Pixel als potentiell gesuchtes Objekt klassifiziert, so heißt dies, dass sich irgendwo innerhalb der Rückprojektion dieses Pixels ein gesuchtes Objekt befinden könnte oder aber, dass die bekannten Gegenstände eine Farbe aus der Menge aller Farben der gesuchten Objekte besitzen. Es bleibt nichts anderes übrig, als in diesem Fall davon auszugehen, dass sich ein gesuchtes Objekt direkt vor der Kamera befindet. Geometrisch ausgedrückt heißt dies, dass man von einer Distanz von "null" zwischen Kamera und gesuchten Objekten für dieses Pixel ausgehen muss. Wird ein Pixel jedoch als nicht-gesuchtes Objekt klassifiziert, so kann man sicher sein, dass man bis hin zu den bekannten Gegenständen blicken kann. Ist die Geometrie dieser Gegenstände bekannt, so ist auch die Distanz zu diesen bekannt und kann wie folgt genutzt werden, um aus der reinen Farbkamera-Pixelfunktion eine tiefenbeschreibende Pixelfunktion zu formulieren:

Definition 20. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera, C die Menge der von dieser Kamera unterscheidbaren Farben, $F_U \subseteq C$ die Menge aller Farben der gesuchten Objekte U und d_F die farbbasierte Segmentierung mittels F_U . Dann ist $i_C^F : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \bar{\mathbb{R}}^+$,

$$i_C^F(E, f, s) := \begin{cases} \inf\{x \in \bar{\mathbb{R}}^+ \mid f + x \cdot s \in E \setminus U\} & , d_F(E, f, s) = L \\ 0 & , d_F(E, f, s) = P \end{cases}$$

die *mittels Farbmodell tiefenbeschreibende Farbkamera-Pixelfunktion*.

Eine solche tiefenbeschreibende Pixelfunktion lässt sich genauso wie die Tiefenkamera-Pixelfunktion zur Beschreibung eines ausschließbaren Raumbereichs nutzen:

Definition 21. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera, $Z \subset \mathbb{R}^n$ eine Rekon-

3. Modellierung

struktionszone und i_C^E die mittels Farbmodell tiefenbeschreibende Farbkamera-Pixelfunktion. Der mittels Farbmodell ausschließbare Raumbereich einer Farbkamera wird beschrieben durch

$$A_C^{K,F} := \{z \in Z \mid \exists s \in S \exists x \in \mathbb{R}^+ : z = f + x \cdot s \text{ mit } x < i_C^E(E, f, s)\}.$$

Jedes Raumelement, welches auf einem rückprojizierten Strahl eines Pixels und gleichzeitig näher an der Kamera liegt als der zugehörigen Pixel-Wert der mittels Farbmodell tiefenbeschreibenden Farbkamera-Pixelfunktion, gehört damit zum mittels Farbmodell ausschließbaren Raumbereich einer Farbkamera.

Statt die Farbe der gesuchten Objekte lässt sich aber auch die Eigenschaft der Unterscheidbarkeit gesuchter Objekte vor einem Hintergrund aus bekannten Gegenständen nutzen:

Definition 22. Seien $E \subset \mathbb{R}^n$ eine Menge von Gegenständen, $K = (f, S)$ eine kalibrierte Kamera und C die Menge der von dieser Kamera unterscheidbaren Farben, dann heißt $h_C^E : H \rightarrow C$, $h_C^E(s) := c$ Hintergrundmodell der Pixelmenge $H \subseteq S$ der Gegenstände E , falls $h_C^E(s) = i_C(E, f, s) \forall s \in H$ gilt (Abbildung 3.8, f).

Das Hintergrundmodell enthält die Farbwerte für eine Teilmenge aller Pixel einer Kamera K für den Fall, dass sich kein gesuchtes Objekt im Sichtbereich der Kamera befindet. Die bewusste Wahl einer Teilmenge aller Pixel einer Kamera kommt der Überlegung nach, dass nicht immer für alle Pixel ein Hintergrund beschrieben werden kann – beispielsweise wenn sich dieser ständig ändert oder aus einem anderen Grund nicht bestimmt werden kann. Konkrete Beispiele dafür sind ein Bildschirm, dessen Bild sich “unvorhersehbar” ändert oder ein Roboter, dessen Erscheinungsbild an seiner aktuellen Position nicht bekannt ist, er aber nicht als gesuchtes Objekt interpretiert werden soll (Abbildung 3.8, b, d).

Unter der Annahme, dass gesuchte Objekte vor bekannten Gegenständen im Kamerabild sich gegenüber dem Erscheinungsbild dieser bekannten Gegenständen unterscheiden, lassen sich die Pixel in Vordergrund F bzw. Hintergrund B wie folgt klassifizieren:

Definition 23. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera, C die Menge der von dieser Kamera unterscheidbaren Farben und $h_C^{E \setminus U}$ das Hintergrundmodell der Menge $E \setminus U$,

dann heißt $d_B : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times H \rightarrow \{F, B\}$,

$$d_B(E, f, s) := \begin{cases} F & , h_C^{E \setminus U}(s) \neq i_C(E, f, s) \\ B & , h_C^{E \setminus U} = i_C(E, f, s) \end{cases}$$

Vordergrund-Hintergrund-Segmentierung mittels $h_C^{E \setminus U}$ (Abbildung 3.8, h, j).

Wenn also der Pixelwert des aktuellen Bildes von dem des Hintergrundmodells abweicht, so “sieht” der Pixel den Hintergrund nicht und wird damit als Vordergrund klassifiziert. Wenn sich der Pixelwert des aktuellen Bildes nicht von dem des Hintergrundmodells unterscheidet, so wird der Pixel als Hintergrund klassifiziert. Diese Segmentierung kann ebenfalls genutzt werden um eine tiefenbeschreibende Farbkamera-Pixelfunktion zu formulieren:

Definition 24. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera, C die Menge der von dieser Kamera unterscheidbaren Farben und d_B eine Vordergrund-Hintergrund-Segmentierung mittels $h_C^{E \setminus U}$. Dann ist $i_C^B : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \bar{\mathbb{R}}^+$,

$$i_C^B(E, f, s) := \begin{cases} \inf\{x \in \bar{\mathbb{R}}^+ \mid f + x \cdot s \in E \setminus U\} & , d_B(E, f, s) = B \\ 0 & , d_B(E, f, s) = F \end{cases}$$

die *mittels Hintergrundmodell tiefenbeschreibende Farbkamera-Pixelfunktion* (Abbildung 3.8, j).

Diese Funktion setzt voraus, dass wie oben $E \setminus U$ ebenfalls geometrisch bekannt und somit auch die Distanzen pro Pixel zwischen Kamera und $E \setminus U$ bekannt sind. Der ausschließbare Raumbereich ergibt sich wieder mittels:

Definition 25. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera, C die Menge der von dieser Kamera unterscheidbaren Farben und d_B eine Vordergrund-Hintergrund-Segmentierung mittels $h_C^{E \setminus U}$. Außerdem sei $Z \subset \mathbb{R}^n$ eine Rekonstruktionszone. Der *mittels Hintergrundmodell ausschließbare Raumbereich einer Farbkamera* (Abbildung 3.8, h) wird beschrieben durch

$$A_C^{K,B} := \{z \in Z \mid \exists s \in H \exists x \in \mathbb{R}^+ : z = f + x \cdot s \text{ mit } x < i_C^B(E, f, s)\}.$$

3. Modellierung

In allen Fällen hat sich gezeigt, dass zunächst eine geometrische Aussage, typischerweise eine Sichttiefe pro Pixel vorliegen muss, bevor man diese dann auf gleichartige Weise zur Beschreibung eines ausschließbaren Raumbereichs nutzt. Im Fall einer Tiefenkamera steht diese Sichttiefe direkt zur Verfügung, während im Fall einer Farbkamera zusätzliches Wissen benötigt wird, beispielsweise in Form der Menge aller Farben gesuchter Objekte oder der Eigenschaft der Unterscheidbarkeit von gesuchten Objekten vor bekannten Gegenständen. In beiden Fällen müssen die bekannten Gegenstände geometrisch modelliert sein, um eine Distanz zu diesen in den tiefenbeschreibenden Pixelfunktionen verwenden zu können.

Weitere Sensoren lassen sich teilweise auf die hier beschriebenen zurückführen – beispielsweise lässt sich der ausschließbare Raumbereich eines Näherungssensors auf ähnliche Weise wie bei einer Tiefenkamera modellieren.

Varianten obiger Methoden zur Ermittlung geometrischer Zusammenhänge sind denkbar, zum Beispiel bei Farbkameras: Für einen Teil der Pixel kommt eine bestimmte Menge von Farben gesuchter Objekte vor und für einen anderen Teil der Pixel ist eine andere Menge von Farben gesuchter Objekte beobachtbar. Dadurch werden unter Umständen ausschließbare Räume auf zulässige Weise größer, jedoch ist die Angabe solcher spezieller Zusammenhänge in der Praxis eher schwierig.

Im Hinblick auf die voxelbasierte Umsetzung (Kapitel 4) ist es geschickt und zulässig, die tiefenbeschreibenden Pixelfunktionen auf spezielle Weise um Teile anderer ausschließbarer Raumbereiche zu erweitern. Während nun also die drei eingeführten tiefenbeschreibenden Pixelfunktionen um Teile weiterer ausschließbarer Volumina erweitert und beschrieben werden, wird der eigentliche Grund dafür erst später im Rahmen der voxelbasierten Umsetzung diskutiert.

Die Idee für die Erweiterung ist die, dass wenn eine tiefenbeschreibende Pixelfunktion die Abwesenheit gesuchter Objekte bis zur Oberfläche bekannter Gegenstände garantieren kann, sie dies auch bis zum Austritt des rückprojizierten Pixelstrahls aus jenen bekannten Gegenständen könnte, sofern diese ebenfalls einen ausschließbaren Raumbereich darstellen (Abbildung 3.9). Im Falle der Tiefenkamera wird damit das Wissen über die Geometrie bekannter Gegenstände zusätzlich notwendig. Die reine Sensorinformation reicht nun nicht mehr aus, da sie die Austrittstiefe eines rückprojizierten Pixels aus dem bekannten Gegenstand nicht enthält. Bei Farbkameras ist dies – wie oben beschrieben – ohnehin schon notwendig, weil sie die Eintrittstiefe nicht selbständig liefern können. Die drei angepassten tiefenbeschreibenden Pixelfunktionen ergeben sich wie folgt und können wie oben direkt zur Beschreibung eines ausschließbaren

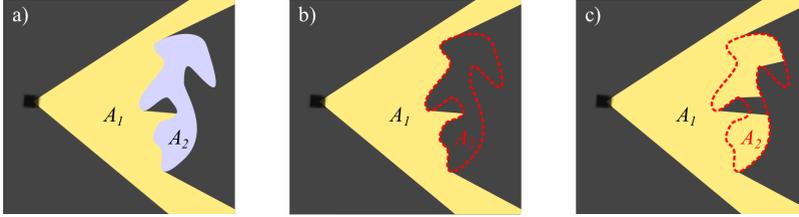


Abbildung 3.9.: Illustration der Idee zur Erweiterung von tiefenbeschreibenden Pixelfunktionen: a) Der gelbe ausschließbare Raumbereich A_1 resultiert aus einer tiefenbeschreibenden Pixelfunktion. Der hellblaue Raumbereich A_2 ist ebenfalls ein ausschließbarer Raumbereich und stammt beispielsweise von einem soliden bekannten Gegenstand. b) Alleinige Darstellung des Raumbereiches A_1 vor der Erweiterung. c) Die tiefenbeschreibenden Pixelfunktionen lassen sich gegenüber der bisherigen Betrachtung derart erweitern, dass sie, sofern sie an die Oberfläche von A_2 anschließen, diesen Raumbereich A_2 entlang der Pixelrückprojektion bis zum ersten Austritt als ausschließbar zusätzlich berücksichtigen dürfen.

Raumbereichs genutzt werden.

Die tiefenbeschreibende Farbkamera-Pixelfunktion welche auf der Menge aller Farben gesuchter Objekte basiert, wird wie folgt erweitert:

Definition 26. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera, C die Menge der von dieser Kamera unterschiedlichen, wahrnehmbaren Farben, $F_U \subseteq C$ die Menge aller Farben gesuchter Objekte U und d_F die farbbaasierte Segmentierung mittels F_U . Dann ist $\hat{i}_C^F : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \overline{\mathbb{R}}^+$,

$$\hat{i}_C^F(E, f, s) := \begin{cases} \inf\{x \in \overline{\mathbb{R}}^+ \mid \exists y < x : f + y \cdot s \in E \setminus U, f + x \cdot s \notin E \setminus U\} & , d_F(E, f, s) = L \\ 0 & , d_F(E, f, s) = P \end{cases}$$

die erweiterte, mittels Farbmodell tiefenbeschreibende Farbkamera-Pixelfunktion.

Es wird das Infimum aller Raumelemente außerhalb der bekannten Gegenstände verwendet, zwischen denen und der Kamera jedoch mindestens ein Raumelement der bekannten Gegenstände liegen muss. Die auf der Eigenschaft der Unterscheidbarkeit gesuchter Objekte vor bekanntem Hintergrund beruhende tiefenbeschreibende Farbkamera-Pixelfunktion wird auf

3. Modellierung

gleiche Weise erweitert:

Definition 27. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$, $K = (f, S)$ eine kalibrierte Kamera, C die Menge der von dieser Kamera unterscheidbaren Farben und d_B eine Vordergrund-Hintergrund-Segmentierung mittels $h_C^{E \setminus U}$. Dann ist $\hat{i}_C^B : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \bar{\mathbb{R}}^+$,

$$\hat{i}_C^B(E, f, s) := \begin{cases} \inf\{x \in \bar{\mathbb{R}}^+ \mid \exists y < x : f + y \cdot s \in E \setminus U, f + x \cdot s \notin E \setminus U\} & , d_B(E, f, s) = B \\ 0 & , d_B(E, f, s) = F \end{cases}$$

die *erweiterte, mittels Hintergrundmodell tiefenbeschreibende Farbkamera-Pixelfunktion*.

Die erweiterte Tiefenkamera-Pixelfunktion ist etwas aufwändiger im Vergleich zur ursprünglichen, da nun geprüft werden muss, ob ein Pixel auch bis zur Oberfläche bekannter Gegenstände “sehen” kann. Erst dann ist es zulässig entlang der Rückprojektion bis zum Austritt aus jenen bekannten Gegenständen Raumelemente in der Distanzbestimmung zu ignorieren:

Definition 28. Sei $E \subset \mathbb{R}^n$ die Umwelt, bestehend aus den gesuchten Objekten $U \subseteq E$ und den bekannten Gegenständen $E \setminus U$ und $K = (f, S)$ eine kalibrierte Kamera. Gelte außerdem das Prädikat $\Pi_i(E, f, s) := i_D(E, f, s) = \inf\{x \in \bar{\mathbb{R}}^+ \mid f + x \cdot s \in E \setminus U\}$, dann ist $\hat{i}_D : 2^{\mathbb{R}^n} \times \mathbb{R}^n \times S \rightarrow \bar{\mathbb{R}}^+$,

$$\hat{i}_D(E, f, s) := \begin{cases} \inf\{x \in \bar{\mathbb{R}}^+ \mid \exists y < x : f + y \cdot s \in E \setminus U, f + x \cdot s \notin E \setminus U\} & , \Pi_i(E, f, s) \\ i_D(E, f, s) & , \text{sonst} \end{cases}$$

die *erweiterte Tiefenkamera-Pixelfunktion*.

In der ersten Zeile wird geprüft, ob der Ausgabewert eines Pixels der Tiefenkamera gleich der “berechneten” Distanz zu den bekannten Gegenständen entspricht. In diesem Fall darf die Rückprojektion bis zum ersten Austritt aus jenen Gegenständen hinzugenommen werden. In der zweiten Zeile wird der bisherige Wert verwendet. Dies entspricht vergleichbar zur zweiten Methode bei Farbkameras einem Hintergrundmodell: Dort wo sich keine gesuchten Objekte vor den bekannten Gegenständen befinden, kann eine Tiefe bis zum ersten Austritt der Rückprojektion verwendet werden.

Im Endeffekt erreicht man durch diese drei erweiterten tiefenbeschreibenden Pixelfunktionen eine Überlappung ausschließbarer Raumbereiche, welche im Rahmen der voxelbasierten Umsetzung enorm hilfreich sein werden.

3.2.3. Erfassbarkeit

Die Erfassbarkeit nutzt die Eigenschaft, dass jede Region gesuchter Objekte in einer bestimmten Mindestanzahl von Sensoren zumindest teilweise innerhalb einer Rekonstruktionszone erfasst wird: Lässt sich beispielsweise garantieren, dass ein Mensch in einem Raum von mindestens m der insgesamt k Sensoren teilweise erfasst wird, so sind diejenigen Regionen einer geometrischen Rekonstruktion von Menschen in der Rekonstruktionszone nicht plausibel, die von weniger als m Sensoren teilweise erfasst werden. Zunächst wird dazu das “teilweise Erfassen gesuchter Objekte in einer Rekonstruktionszone” unter Zuhilfenahme der sensorbasierten ausschließbaren Raumbereiche (hier jedoch *ohne* die beschriebene Erweiterung um Teile anderer ausschließbarer Raumbereiche, vgl. Kapitel 3.2.2) definiert:

Definition 29. Seien $Z \subset \mathbb{R}^n$ die Menge, welche die Rekonstruktionszone beschreibt und $E \subset \mathbb{R}^n$ die Menge, welche die Umwelt beschreibt, wobei $U \subseteq E$ die Menge der gesuchten Objekte und $E \setminus U$ die Menge der bekannten Gegenstände darstellt. Seien $A^{E \setminus U}, A^E \subseteq Z$ die ausschließbaren Raumbereiche eines Sensors – ohne Erweiterung um sonstige ausschließbare Raumbereiche – einzig und allein unter Berücksichtigung der bekannten Gegenstände $E \setminus U$ bzw. der gesamten Umwelt E . Die Menge $D = A^{E \setminus U} \setminus A^E$ beschreibt Raumelemente, welche vom Sensor als *gesuchte Objekte erfasst* werden (Abbildung 3.10).

Einfach gesagt handelt es sich bei dieser Menge D um jene Raumelemente, die bei Abwesenheit gesuchter Objekte zum sensorisch ausschließbaren Raumbereich gehören, nicht aber mehr bei Anwesenheit der gesuchten Objekte. Gilt für alle Regionen ϱ (gemäß Definition 5) gesuchter Objekte $U \subset \mathbb{R}^n$ und von $k \in \mathbb{N}$ Sensoren als gesuchte Objekte erfassten Mengen $D_1, \dots, D_k \subseteq Z$ also $|\{D_i \in \{D_1, \dots, D_k\} \mid D_i \cap \varrho \neq \emptyset\}| \geq m \in \mathbb{N}_0$, so sind alle Regionen einer geometrischen Rekonstruktion von U in Z nicht plausibel, welche in weniger als m Sensoren als gesuchte Objekte erfasst werden. Diese Eigenschaft wird nun als Plausibilisierungsfunktion formuliert:

Definition 30. Sei $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subseteq E \subset \mathbb{R}^n$ in Z , wobei U die Menge der gesuchten Objekte und $E \setminus U$ die Menge der bekannten Gegenstände

3. Modellierung

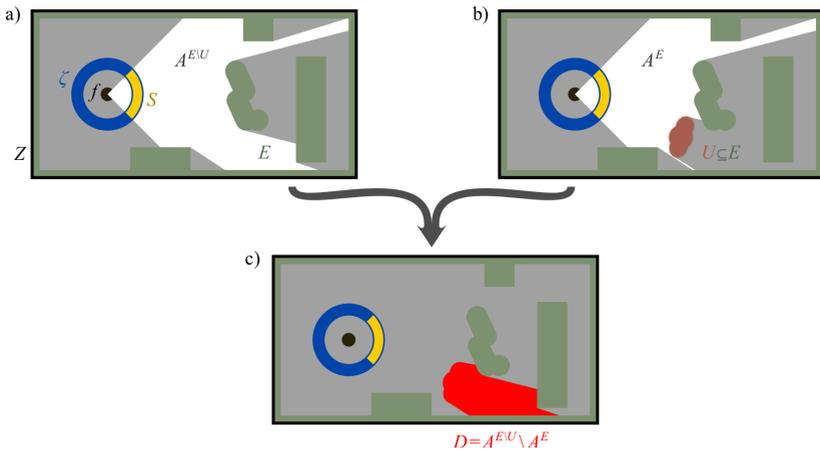


Abbildung 3.10.: Illustration sensorisch erfasster gesuchter Objekte: Die Umwelt $E \subset \mathbb{R}^n$ besteht aus gesuchten Objekten $U \subseteq E$ (dunkelrot) und aus bekannten Gegenständen $E \setminus U$ (dunkelgrün). a) Die Menge $A^{E \setminus U} \subseteq Z$ (weiß) stellt den ausschließbaren Raumbereich in einer Umwelt ohne gesuchte Objekte $E \setminus U$ dar. b) Die Menge $A^E \subseteq Z$ (weiß) stellt den ausschließbaren Raumbereich mit gesuchten Objekten $U \subseteq E$ dar. c) Die Menge $D = A^{E \setminus U} \setminus A^E$ (rot) beschreibt Raumelemente, welche vom Sensor als *gesuchte Objekte erfasst* werden.

darstellen. Des Weiteren seien $D_1, \dots, D_k \subseteq Z$ die von $k \in \mathbb{N}$ Sensoren als gesuchtes Objekt erfassten Mengen und es gelte das Prädikat $\Pi_e(\varrho, D_1, \dots, D_k, m) := |\{D_i \in \{D_1, \dots, D_k\} \mid D_i \cap \varrho \neq \emptyset\}| \geq m$, mit $m \in \mathbb{N}_0$ und $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_e : 2^Z \times 2^{\mathbb{R}^n} \times \underbrace{2^Z \times \dots \times 2^Z}_k \times \mathbb{N}_0 \rightarrow 2^Z, p_e(R, Z, D_1, \dots, D_k, m) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_e(\varrho, D_1, \dots, D_k, m)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *allgemeine globale Erfassbarkeitsfunktion* (Abbildung 3.11 a, b, c).

Da ein gesuchtes Objekt über den Rand der Rekonstruktionszone $\text{Rand}(Z)$ eindringen kann, werden die Regionen einer geometrischen Rekonstruktion, die am Rand der Rekonstruktionszone liegen, ebenfalls als plausibel behandelt. Würde man Regionen am Rand nicht automatisch als plausibel betrachten und würde ein kleiner Teil des Randes von keinem Sensor eingesehen werden, so müsste die Mindestanzahl an Sensoren, die ein gesuchtes Objekt als solches erfassen direkt auf Null gesetzt werden, denn im nicht-ingesehenen Teil könnte kein Sensor das gesuchte Objekt erfassen.

Ist hingegen bekannt, dass alle Regionen ϱ gesuchter Objekte an der *selben* Stelle von einer Mindestanzahl $m \in \mathbb{N}_0$ von maximal $k \in \mathbb{N}$ Sensoren als gesuchtes Objekt erfasst werden; gilt also $\exists M \subseteq \{1, \dots, k\} : (|M| \geq m) \wedge (\bigcap_{i \in M} D_i \cap \varrho \neq \emptyset)$, so resultiert daraus die *spezielle* globale Erfassbarkeitsfunktion:

Definition 31. Sei $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subseteq E \subset \mathbb{R}^n$ in Z , wobei U die Menge der gesuchten Objekte und $E \setminus U$ die Menge der bekannten Gegenstände darstellen. Des Weiteren seien $D_1, \dots, D_k \subseteq Z$ die von $k \in \mathbb{N}$ Sensoren als gesuchtes Objekt erfassten Mengen und gelte das Prädikat

$$\Pi_{\hat{e}}(\varrho, D_1, \dots, D_k, m) := \exists M \subseteq \{1, \dots, k\} : (|M| \geq m) \wedge \left(\bigcap_{i \in M} D_i \cap \varrho \neq \emptyset \right),$$

mit $m \in \mathbb{N}_0$ und $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_{\hat{e}} : 2^Z \times 2^{\mathbb{R}^n} \times \underbrace{2^Z \times \dots \times 2^Z}_k \times \mathbb{N}_0 \rightarrow 2^Z, p_{\hat{e}}(R, Z, D_1, \dots, D_k, m) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_{\hat{e}}(\varrho, D_1, \dots, D_k, m)}} \varrho$$

3. Modellierung

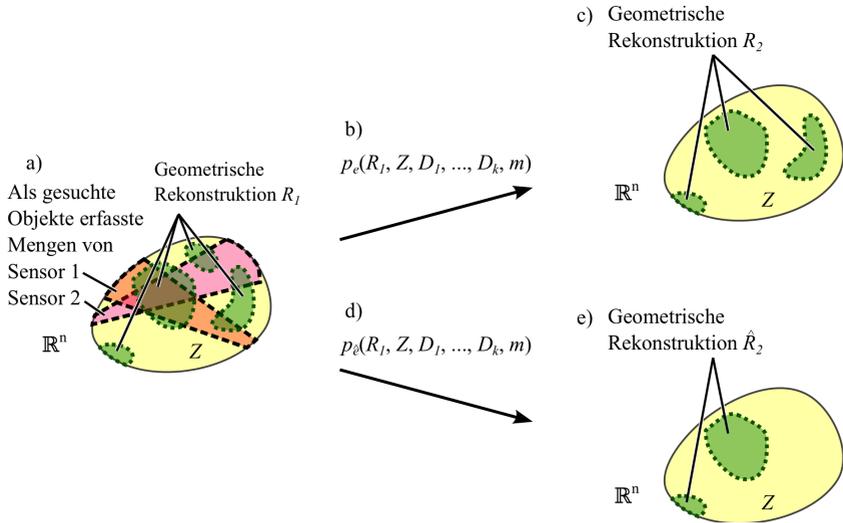


Abbildung 3.11.: Illustration der allgemeinen globalen Erfassbarkeitsfunktion (a, b, c) und der speziellen globalen Erfassbarkeitsfunktion (a, d, e). Die allgemeine globale Erfassbarkeitsfunktion verwirft diejenigen Regionen der geometrischen Rekonstruktion R_1 , die von weniger als m Sensoren jeweils teilweise detektiert werden, sofern diese nicht am Rand von Z liegen (hier: $m = 2$). Im dargestellten Beispiel werden Regionen (grün) verworfen, wenn sie in weniger als in zwei Sensoren als gesuchte Objekte erfasst werden. Bei der *allgemeinen* globalen Erfassbarkeitsfunktion ist eine Region bereits dann plausibel, wenn sie an *verschiedenen Stellen* von m Sensoren detektiert wird, bei der *speziellen* globalen Erfassbarkeitsfunktion muss sie an den *gleichen Stellen* von m Sensoren detektiert werden.

eine Plausibilisierungsfunktion und heißt *spezielle globale Erfassbarkeitsfunktion* (Abbildung 3.11 a, d, e).

Da die Mindestanzahl an Sensoren in beiden Fällen für die gesamte Rekonstruktionszone gilt, werden diese Erfassbarkeitsfunktionen als *global* bezeichnet. Ist nun jedoch ein Teil der Rekonstruktionszone mit mehr Sensoren ausgestattet als ein anderer Teil, so könnte der andere Teil sich limitierend für die Mindestanzahl an Sensoren in Hinsicht auf den ersten Teil auswirken. Um dem entgegen zu wirken, ist es daher wünschenswert unterschiedliche Mindestanzahlen an Sensoren für diese beiden Teil-Rekonstruktionszonen festlegen zu können. Aus dieser Forderung heraus resultiert die sogenannte *lokale* Variante der Erfassbarkeit, in welcher die Mindestanzahl an Sensoren ortsabhängig betrachtet wird. Es kann damit also festgelegt werden, dass in unterschiedlichen Bereichen der Rekonstruktionszone unterschiedliche Mindestanzahlen an Sensoren die gesuchten Objekte erfassen.

Definition 32. Sei $R \subseteq Z \subseteq \mathbb{R}^n$ eine geometrische Rekonstruktion von $U \subseteq \mathbb{R}^n$ in Z und sei außerdem $f : Z \rightarrow \mathbb{N}_0$ mit $f(z) = m$ eine Funktion, die jedem $z \in Z$ in der Rekonstruktionszone Z eine Mindestanzahl $f(z)$ von maximal $k \in \mathbb{N}$ Sensoren zuordnet, welche die gesuchten Objekte U erfassen und $\Delta = \{(z, m) \in Z \times \mathbb{N}_0 \mid f(z) = m\}$ die linkstotale und rechtseindeutige Relation, die diese Funktion beschreibt. Ausserdem seien $D_1, \dots, D_k \subseteq Z$ die von den k Sensoren als gesuchtes Objekt erfassten Mengen und gelte das Prädikat $\Pi_E(\varrho, D_1, \dots, D_k, \Delta) := |\{D_i \in \{D_1, \dots, D_k\} \mid D_i \cap \varrho \neq \emptyset\}| \geq \min\{f(z) \mid z \in \varrho\}$, mit $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_E : 2^Z \times \underbrace{2^Z \times \dots \times 2^Z}_k \times 2^{Z \times \mathbb{N}_0} \rightarrow 2^Z, p_E(R, D_1, \dots, D_k, \Delta) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \Pi_E(\varrho, D_1, \dots, D_k, \Delta)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *allgemeine lokale Erfassbarkeitsfunktion*.

Da in der Definition jedem $z \in Z \subseteq \mathbb{R}^n$ eine Mindestanzahl von $f(z)$ Sensoren zugeordnet wird, mit denen plausible Regionen gesuchter Objekte $U \subseteq \mathbb{R}^n$ detektiert werden, muss eine Region $\varrho \subseteq R$ einer geometrischen Rekonstruktion $R \subseteq Z$ als plausibel gewertet werden, sobald die kleinste Mindestanzahl $f(z)$ aller $z \in \varrho$ aus der Region ϱ erreicht wird. Aus diesem Grund ist die min-Funktion zu verwenden.

Würde die max-Funktion verwendet werden, könnte es passieren, dass eine Region einer geometrischen Rekonstruktion, welche eine Region gesuchter Objekte enthalten kann, in einen

3. Modellierung

Bereich mit einer höheren Mindestanzahl hineinreicht. Dies hätte zur Folge, dass entsprechend mehr Sensoren die Region der geometrischen Rekonstruktion als gesuchtes Objekt erfassen müssten, was aber aufgrund einer möglicherweise entarteten Form dieser Region der geometrischen Rekonstruktion nicht zwangsläufig erfüllt sein muss. In dem Fall könnte also eine plausible Region verworfen werden.

Analog zu oben gibt sich die spezielle lokale Erfassbarkeitsfunktion wie folgt:

Definition 33. Sei $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z und sei außerdem $f : Z \rightarrow \mathbb{N}_0$, $f(z) = m$ eine Funktion, die jedem $z \in Z$ in der Rekonstruktionszone eine Mindestanzahl $f(z)$ von maximal $k \in \mathbb{N}$ Sensoren zuordnet, welche die gesuchten Objekte U erfassen und $\Delta = \{(z, m) \in Z \times \mathbb{N}_0 \mid f(z) = m\}$ die linkstotale und rechtseindeutige Relation, die diese Funktion beschreibt. Außerdem seien $D_1, \dots, D_k \subseteq Z$ die von den k Sensoren als gesuchtes Objekt erfassten Mengen und gelte das Prädikat

$$\Pi_{\hat{E}}(\varrho, D_1, \dots, D_k, \Delta) := \exists M \subseteq \{1, \dots, k\} : |M| \geq \min\{f(z) \mid z \in \varrho\} \wedge \bigcap_{i \in M} D_i \cap \varrho \neq \emptyset$$

mit $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_{\hat{E}} : 2^Z \times \underbrace{2^Z \times \dots \times 2^Z}_k \times 2^{Z \times \mathbb{N}_0} \rightarrow 2^Z, p_{\hat{E}}(R, D_1, \dots, D_k, \Delta) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \Pi_{\hat{E}}(\varrho, D_1, \dots, D_k, \Delta)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *spezielle lokale Erfassbarkeitsfunktion*.

Im Vergleich zu den beiden globalen Erfassbarkeitsfunktionen wird in den lokalen Varianten der Rand bei der Prüfung auf Plausibilität außen vor gelassen, da dieser aufgrund der ortsabhängigen Angabe der Mindestanzahl explizit berücksichtigt werden kann – im Gegensatz zu den globalen Erfassbarkeitsfunktionen kann man hier konkret die Mindestanzahl am Rand reduzieren oder gar auf Null setzen, wenn kein Sensor in der Lage ist, dort gesuchte Objekte zu erfassen.

Es folgt nun die Zuordnung der vorgestellten Plausibilisierungsfunktionen zu einem der beiden Typen gemäß Definition 10. Alle vorgestellten Erfassbarkeitsfunktionen sind, wie man anhand der jeweiligen Definition ablesen kann, *regionenbasiert*, da sie für jede Region einzeln und unabhängig von anderen Regionen entscheiden, ob diese Region zur auszugebenden geometrischen Rekonstruktion gehören oder nicht. Dieses Schema ist typisch für viele regionenbasierte Plau-

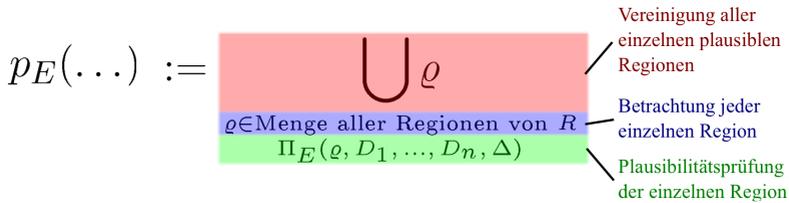


Abbildung 3.12.: Darstellung einer typischen regionenbasierten Plausibilisierungsfunktion: Regionen werden separat betrachtet (blau) und separat auf Plausibilität geprüft (grün). Alle einzelne Region, welche als plausibel ausgewertet werden, werden zur auszugebenden geometrischen Rekonstruktion vereint (rot).

sibilisierungsfunktion und wird in Abbildung 3.12 veranschaulicht. Außerdem sind alle Erfassbarkeitsfunktionen *Teilmengen-verwerfend*, weil eine kleinere Teilmenge maximal von der gleichen Anzahl von Sensoren teilweise als gesuchtes Objekt erfasst wird und Regionen mit einer zu kleinen Anzahl von Sensoren, welche diese teilweise als gesuchtes Objekt erfassen, als nicht-plausibel gelten. Erfassbarkeitsfunktionen sind *vollständig*, da eine erneute Auswertung dieser kein unterschiedliches Ergebnis liefern kann – die als gesuchtes Objekt detektierten Mengen der Sensoren sind nach wie vor die gleichen und wirken sich somit gleich im Rahmen der Plausibilitätsprüfung aus. Eine vormals als plausibel klassifizierte Region kann daher nicht plötzlich nicht-plausibel sein. Aufgrund dieser drei Eigenschaften sind damit alle vorgestellten Erfassbarkeitsfunktionen vom *Typ R* (Definition 10).

Insgesamt erweist sich die korrekte Wahl der Mindestanzahl von Sensoren in der Praxis selbst für die “einfache” allgemeine globale Erfassbarkeitsfunktion als sehr schwierig, da abgeschätzt werden muss in wie vielen Sensoren gesuchte Objekte teilweise erfasst werden. In dem Beispiel der Mensch/Roboter-Koexistenz müsste dazu der Mensch in der gesamten Rekonstruktionszone und in allen möglichen Posen in Verbindung mit allen möglichen Roboterposen auf Erfassbarkeit in den Sensoren überprüft werden, um so die korrekte Mindestanzahl an Sensoren zu ermitteln. Dies führt dazu, dass die Erfassbarkeitsfunktionen nur eingeschränkt sinnvoll einsetzbar sind. Im Zweifelsfall ist die Wahl einer geringeren Mindestanzahl die sicherere, da weniger Regionen verworfen werden. Dennoch müsste die Wahl überprüft werden, um tatsächlich weiterhin eine geometrische Rekonstruktion gewährleisten zu können. Denkbar, aber in dieser Arbeit nicht weiter untersucht, ist der Einsatz von geeigneten Heuristiken, um eine konservative Abschätzung dieses Parameters mit angemessenem Rechenaufwand im Rahmen einer Simulation zu ermitteln.

3.2.4. Form bezüglich Raumbereich

Auch läßt sich oftmals ein Raumbereich $M \subset \mathbb{R}^n$ identifizieren, in welchen keine Region der gesuchten Objekte $U \subset \mathbb{R}^n$ aufgrund ihrer Form hineinpasst ohne dabei herauszuragen. Umgekehrt kann man sagen, dass jede Region der gesuchten Objekte aus dem Raumbereich M herausragt, und damit $\varrho \setminus M \neq \emptyset$, wobei $\varrho \in$ Menge aller Regionen von U gilt. Beispielsweise wird ein durchschnittlich großer, erwachsener Mensch aus einem schmalen Quader – egal wo sich dieser Mensch befindet – immer herausragen, da er sich nicht so bewegen und verformen kann, dass er ohne herauszuragen komplett hinein passt.

Zu beachten ist, dass eine Region eines gesuchten Objektes aus M herausragen kann, ohne dass der herausragende Teil innerhalb der Rekonstruktionszone Z liegt. Das ist dann der Fall, wenn sich M am $\text{Rand}(Z)$ der Rekonstruktionszone Z befindet. Aus diesem Grund dürfen Regionen einer geometrischen Rekonstruktion auch dann nicht verworfen werden, wenn sie am Rand von Z liegen, da mittels einer geometrischen Rekonstruktion nur ein Herausragen innerhalb Z berücksichtigt werden kann.

Definition 34. Sei $R \subseteq Z \subset \mathbb{R}^n$ eine geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z und außerdem seien $M \subset \mathbb{R}^n$ und das Prädikat $\Pi_f(\varrho, M) := \varrho \setminus M \neq \emptyset$ mit $\varrho \in$ Menge aller Regionen von U gegeben. Dann ist

$$p_f : 2^Z \times 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \rightarrow 2^Z, p_f(R, Z, M) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_f(\varrho, M)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *Formfunktion* (Abbildung 3.13, a, b, c).

Alle nicht-plausiblen Regionen einer gegebenen geometrischen Rekonstruktion, also alle Regionen, die sich weder mit dem Rand der Rekonstruktionszone überschneiden, noch aus dem spezifizierten Raumbereich herausragen, werden durch diese Plausibilisierungsfunktion verworfen.

Es ist möglich, solche Raumbereiche, aus welchen Regionen gesuchter Objekte immer herausragen werden zu generieren: Da ein Mensch eine gewisse “Mindestdicke” besitzt, kann man um jeden bekannten Gegenstand innerhalb der Rekonstruktionszone eine “dünnere” Hülle legen, aus der der Mensch aufgrund seiner “Mindestdicke” immer herausragen wird.

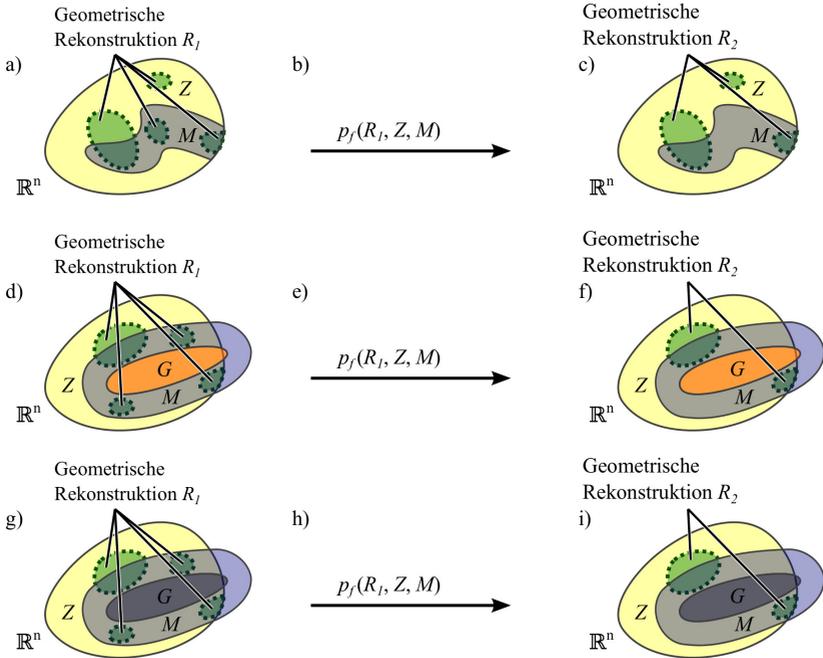


Abbildung 3.13.: Illustration der Formfunktion. Die geometrischen Rekonstruktionen R_1 (a, d, g) werden mittels der Formfunktion (b, e, h) in geometrische Rekonstruktionen R_2 (c, f, i) überführt. Es werden diejenigen Regionen verworfen, welche komplett innerhalb der spezifizierten Menge M liegen und sich gleichzeitig nicht am Rand von Z befinden. Zur einfachen Festlegung solcher Menge bietet sich die Verwendung der Hülle um Gegenstände an (d, e, f). Im Hinblick auf die spätere Umsetzung ist es vorteilhaft, die erweiterte Hülle um Gegenstände zu verwenden (Definition 36), sofern diese Gegenstände gleichzeitig einen ausschließbaren Raumbereich darstellen (g, h, i).

3. Modellierung

Definition 35. Sei $G \subset \mathbb{R}^n$ ein Gegenstand und $r \in \mathbb{R}^+$ ein Radius, dann heißt $M_r^G := \{x \in \mathbb{R}^n \setminus G \mid \exists g \in G : \|x - g\|_2 \leq r\}$ *Hülle um G mit Radius r* (Abbildung 3.13, d, e, f).

Um eine solche Hülle im Rahmen der Formfunktion einsetzen zu dürfen, muss nach wie vor gelten, dass Regionen gesuchter Objekte aus dieser herausragen.

Im Hinblick auf die voxelbasierte Umsetzung (Kapitel 4) ist es geschickt und zulässig, die Hülle um den bekannten Gegenstand selbst zu erweitern, sofern dieser als ausschließbarer Raumbereich verwendet wird. Ohne den genauen Grund an dieser Stelle zu diskutieren ist einleuchtend, dass das Resultat jedenfalls das selbe ist. Die erweiterte Hülle lautet wie folgt:

Definition 36. Sei $G \subset \mathbb{R}^n$ ein Gegenstand und $r \in \mathbb{R}^+$ ein Radius, dann heißt $\hat{M}_r^G := M_r^G \cup G$ *erweiterte Hülle um G mit Radius r* (Abbildung 3.13, g, h, i).

Ragen Regionen gesuchter Objekte aus der Hülle eines Gegenstandes, so ragen sie auch aus der erweiterten Hülle, sofern gewährleistet ist, dass sie nicht in den Gegenstand eindringen können.

Zu beachten ist, dass die Anwendung dieser Plausibilisierungsfunktion auf vereinigte Raumbereiche ein anderes Resultat haben kann, als die sukzessive Anwendung der Plausibilisierungsfunktion auf die einzelnen, getrennten Raumbereiche (Abbildung 3.14).

Es folgt nun die Zuordnung der Plausibilisierungsfunktion zu einem der beiden Typen gemäß Definition 10. Die Formfunktion weist das gleiche, typische Schema regionenbasierter Plausibilisierungsfunktionen auf (Abbildung 3.12) und ist folglich *regionenbasiert*. Weiterhin ist sie *Teilregionen-verwerfend*, da jede Teilmenge einer zu verwerfenden Region ebenfalls verworfen wird. Außerdem ist die Formfunktion *vollständig*, da eine zweite Auswertung keine weiteren Regionen verwerfen kann – die vormalig als plausibel ermittelten Regionen werden nach wie vor aus dem gegebenen Raumbereich herausragen und damit plausibel bleiben. Aufgrund der drei Eigenschaften ist die Formfunktion vom *Typ R* (Definition 10).

3.2.5. Volumen

Kennt man den minimalen Volumeninhalt $v \in \mathbb{R}^+$ aller Regionen gesuchter Objekte $U \subset \mathbb{R}^n$, also $v = \inf\{\lambda(\varrho) \mid \varrho \in \text{Menge aller Regionen von } U\}$, wobei λ das Lebesgue-Maß darstellt, so gilt $\lambda(\varrho) \geq v$, $\varrho \in \text{Menge aller Regionen von } U$. Auch hier ist zu beachten, dass das Volumen einer Region der gesuchten Objekte größer sein kann als das Volumen dieser Region geschnitten

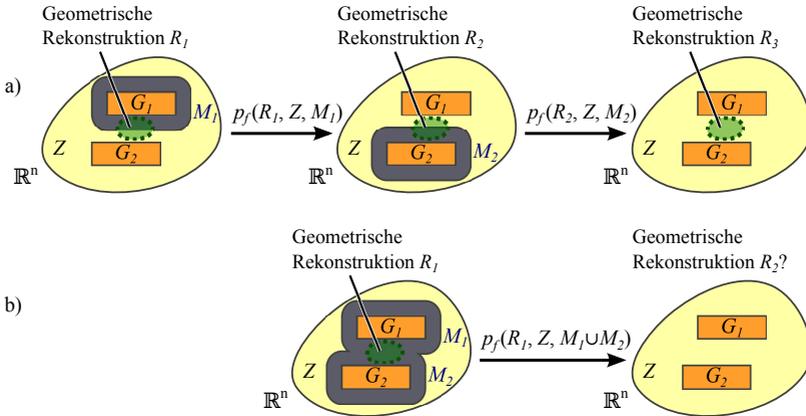


Abbildung 3.14.: Bei der Formfunktion ist zu beachten, dass die Anwendung dieser auf die Vereinigung zweier Raumbereiche (b) ein anderes Resultat haben kann, als die sukzessive Anwendung auf die einzelnen getrennten Raumbereiche (a).

mit der Rekonstruktionszone, nämlich dann, wenn diese Region aus der Rekonstruktionszone herausragt. Da durch eine geometrische Rekonstruktion ausschließlich innerhalb von Z Betrachtungen stattfinden, dürfen die Regionen einer gegebenen geometrischen Rekonstruktion, welche sich mit dem Rand von Z überschneiden nicht verworfen werden – eine sich darin befindende Region gesuchter Objekte könnte aus Z herausragen und somit ein größeres Volumen besitzen als es anhand der geometrischen Rekonstruktion bestimmt werden kann.

Definition 37. Seien $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z , außerdem $v \in \mathbb{R}^+$ ein Mindestvolumen und gelte das Prädikat $\Pi_v(\varrho, v) := \lambda(\varrho) \geq v$, wobei $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_v : 2^Z \times 2^{\mathbb{R}^n} \times \mathbb{R}^+ \rightarrow 2^Z, p_v(R, Z, v) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } U \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_v(\varrho, v)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *globale Volumenfunktion* (Abbildung 3.15, a, b, c).

Diese Plausibilisierungsfunktion verwirft alle nicht-plausiblen Regionen einer gegebenen geometrischen Rekonstruktion, also alle Regionen, die nicht am Rand der Rekonstruktionszone liegen und gleichzeitig einen geringeren Volumeninhalt besitzen als der minimale Volumeninhalt

3. Modellierung

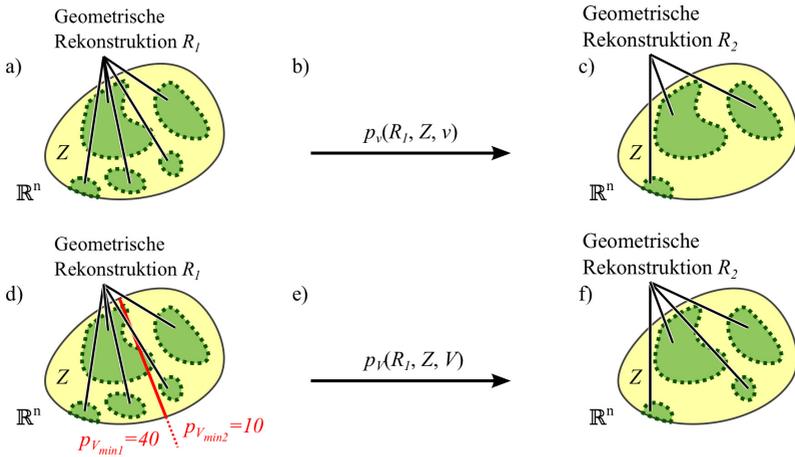


Abbildung 3.15.: Illustration der globalen (a, b, c) und lokalen (d, e, f) Volumenfunktion. Die geometrische Rekonstruktion R_1 (a, d) wird mittels der Volumenfunktion (b, e) in eine geometrische Rekonstruktion R_2 (c, f) überführt. Regionen mit zu kleinem Volumen werden dabei verworfen, sofern sie nicht am Rand von Z liegen. In der lokalen Variante lässt sich jedem $z \in Z$ ein lokales Mindestvolumen zuordnen – hier 40 bzw. 10.

aller Regionen gesuchter Objekte.

Die globale Volumenfunktion nutzt eine in der Rekonstruktionszone global geltende Aussage über den minimalen Volumeninhalt aller Regionen gesuchter Objekte. Im Gegensatz dazu ist denkbar, dass eine solche Information über den minimalen Volumeninhalt aller Regionen gesuchter Objekte lokal vorhanden ist – also abhängig vom Ort innerhalb der Rekonstruktionszone. Beispielsweise ist denkbar, dass in einem Teil der Rekonstruktionszone Menschen rekonstruiert werden sollen und in einem anderen Teil kleinere Werkstücke. Nun hat man die Möglichkeit das Mindestvolumen in dem Teil, in welchem Menschen rekonstruiert werden sollen, mit einem größeren Wert anzugeben, als in dem Teil, in welchem die kleineren Werkstücke rekonstruiert werden sollen. Würde man aber ein global geltendes Mindestvolumen ansetzen, so müsste man das der kleineren Werkstücke verwenden, da diese ansonsten “verloren” gingen. Folglich würde aber die Güte der Rekonstruktion in dem Teil, in dem nur der Mensch rekonstruiert werden muss, sinken, da Regionen, die eigentlich nicht plausibel sind beibehalten werden, weil sie ein größeres Volumen als die Werkstücke besitzen.

Definition 38. Sei $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z , und sei außerdem $f(z) \in \mathbb{R}^+$, mit $z \in Z \subset \mathbb{R}^n$ eine Funktion, die jedem $z \in Z$ in der Rekonstruktionszone Z einen minimalen Volumeninhalt zuordnet und $V = \{(z, v) \in Z \times \mathbb{R}^+ \mid f(z) = v\}$ die linkstotale und rechtseindeutige Relation, die diese Funktion beschreibt, sodass das Prädikat $\Pi_V(\varrho, V) := \lambda(\varrho) \geq \inf\{v \mid (z, v) \in V \wedge z \in \varrho\}$, mit ϱ Menge aller Regionen von U gilt, so ist

$$p_V : 2^Z \times 2^{\mathbb{R}^n} \times 2^{Z \times \mathbb{R}^+} \rightarrow 2^Z, p_V(R, Z, V) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_V(\varrho, V)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *lokale Volumenfunktion* (Abbildung 3.15, d, e, f).

In der lokalen Variante der Volumenfunktion wird eine Region der gegebenen geometrischen Rekonstruktion dann verworfen, wenn ihr Volumeninhalt das Minimum aller über die Funktion spezifizierten minimalen Volumeneinhalte an den Stellen der Raumelemente die dieser Region angehören unterschreitet, statt wie in der globalen Variante nur den einen (globalen) minimalen Volumeninhalt.

Es findet nun die Zuordnung der Plausibilisierungsfunktionen zu einem der beiden Typen gemäß Definition 10 statt. Wieder folgen die Definitionen beider Volumenfunktionen dem typischen Schema einer regionenbasierten Plausibilisierungsfunktion (Abbildung 3.12) und sind

3. Modellierung

daher ebenfalls *regionenbasiert*. Weiterhin sind beide Volumenfunktionen *Teilregionen-verwerfend*, da eine Teilregion in jedem Fall das gleiche oder ein geringeres Volumen besitzt und somit verworfen wird, sofern die zugrunde liegende Region verworfen wird. Außerdem sind beide Volumenfunktionen *vollständig*, da bei einer zweiten Auswertung keine weitere Region verworfen werden kann – die vormalig als plausibel ermittelten Regionen werden nach wie vor ein größeres Volumen besitzen, als das spezifizierte Volumen bzw. ortsabhängige Volumen. Beide Plausibilisierungsfunktionen sind vom *Typ R*, da sie die drei notwendigen Eigenschaften besitzen (Definition 10).

3.2.6. Distanz

Eine weitere Möglichkeit Regionen einer geometrischen Rekonstruktion als nicht-plausibel zu identifizieren und folglich verwerfen zu können beruht auf der Betrachtung von Abständen. Beispielsweise hat ein Mensch, sofern keine Schwerelosigkeit herrscht, immer Kontakt mit seiner Umwelt – typischerweise dem Boden – bzw. hat er eine gewisse Maximaldistanz zum Boden, wenn er springt.

Haben Regionen gesuchter Objekte $U \subset \mathbb{R}^n$ die Eigenschaft eine maximale Distanz $d \in \mathbb{R}_0^+$ zu einer nicht-leeren Menge $M \subset \mathbb{R}^n$ zu besitzen – gilt also $\inf\{\|u - m\|_2 \mid u \in \varrho, m \in M\} \leq d$ mit $\varrho \in$ Menge aller Regionen von U , so sind alle Regionen einer geometrischen Rekonstruktion $R \subseteq Z \subset \mathbb{R}^n$ von U in Z , welche sich nicht mit dem Rand von Z überschneiden und deren Abstand zu M größer ist als d nicht plausibel.

Definition 39. Seien $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z , außerdem $M \subset \mathbb{R}^n$ die nicht-leere Menge zu der Distanzen betrachtet werden, $d \in \mathbb{R}_0^+$ die maximale Distanz und gelte das Prädikat $\Pi_d(\varrho, M, d) := \inf\{\|u - m\|_2 \mid u \in \varrho, m \in M\} \leq d$ mit $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_d : 2^Z \times 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \times \mathbb{R}_0^+ \rightarrow 2^Z, p_d(R, Z, M, d) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_d(\varrho, M, d)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *globale Distanzfunktion* (Abbildung 3.16, a, b, c).

Ebenso wie bei der Volumenfunktion ist eine lokale Variante denkbar, die eine ortsabhängige Angabe der Maximaldistanz erlaubt:

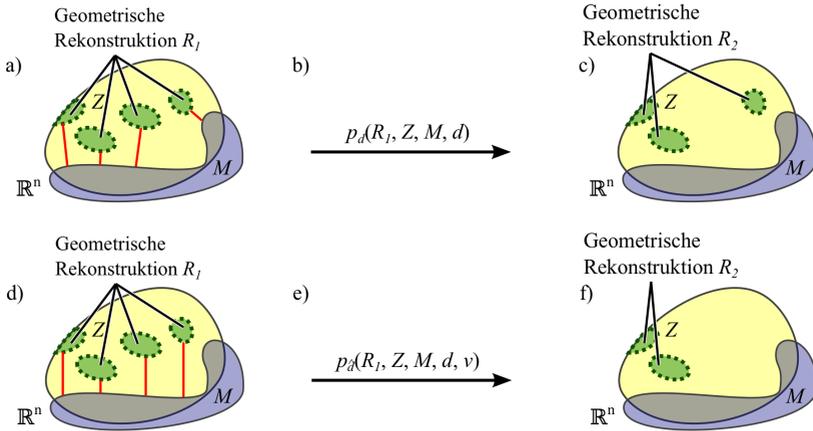


Abbildung 3.16.: Illustration der globalen Distanzfunktion (a, b, c) und der richtungsspezifischen globalen Distanzfunktion (d, e, f). Die geometrische Rekonstruktion R_1 (a, d) wird mittels der Distanzfunktion (b, e) in eine geometrische Rekonstruktion R_2 (c, f) überführt. Regionen mit zu großem Abstand zur Menge M werden dabei verworfen, sofern sie nicht am Rand von Z liegen. Bei der richtungsspezifischen Variante werden Distanzen nur in die spezifizierte Richtung v betrachtet.

3. Modellierung

Definition 40. Sei $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z , und sei außerdem $f(z) \in \mathbb{R}_0^+$ mit $z \in Z \subset \mathbb{R}^n$ eine Funktion, die jedem $z \in Z$ in der Rekonstruktionszone Z eine maximale Distanz zu einer Menge $M \subset \mathbb{R}^n$ zuordnet und $D = \{(z, d) \in Z \times \mathbb{R}_0^+ \mid f(z) = d\}$ die linkstotale und rechtseindeutige Relation, die diese Funktion beschreibt, sodass das Prädikat $\Pi_D(\varrho, M, D) := \inf\{\|u - m\|_2 \mid u \in \varrho, m \in M\} \leq \sup\{d \mid (z, d) \in D \wedge z \in \varrho\}$ mit ϱ Menge aller Regionen von U gilt, so ist

$$p_D : 2^Z \times 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \times 2^{Z \times \mathbb{R}_0^+} \rightarrow 2^Z, p_D(R, Z, M, D) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_D(\varrho, M, D)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *lokale Distanzfunktion*.

Die lokale Distanzfunktion verwirft Regionen einer geometrischen Rekonstruktion, die nicht am Rand der Rekonstruktionszone liegen und die gleichzeitig die maximale aller über die Funktion spezifizierten Distanzen an den Raumpositionen dieser Regionen zur gegebenen Menge überschreitet. Alle Regionen, die diese maximale Distanz nicht überschreiten, gelten als plausibel und werden beibehalten.

Soll die Distanz ausschließlich in einer bestimmten Richtung betrachtet werden, so benötigt man die im Folgenden definierte globale und lokale *richtungsspezifische* Distanzfunktion. Eine solche Funktion ist beispielsweise dann nützlich, wenn ausgedrückt werden soll, dass gesuchte Objekte nicht schweben können und daher eine maximale Distanz zu bekannten Gegenständen entgegen der Richtung der Erdanziehungskraft haben. In diesem Fall sind alle schwebenden Regionen einer geometrischen Rekonstruktion nicht plausibel und können damit verworfen werden.

Definition 41. Seien $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z , außerdem $M \subset \mathbb{R}^n$ die nicht-leere Menge zu der Distanzen betrachtet werden, $d \in \mathbb{R}_0^+$ die maximale Distanz, $v \in \mathbb{R}^n$ der Richtungsvektor und gelte das Prädikat

$$\Pi_d(\varrho, M, d, v) := \begin{cases} \inf\{\|u - m\|_2 \mid u \in \varrho, m \in M, \angle(u - m, v) = 0\} \leq d & , \exists u \in \varrho, \exists m \in M : \angle(u - m, v) = 0 \\ \perp & , \text{sonst} \end{cases}$$

wobei $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_d : 2^Z \times 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \times \mathbb{R}_0^+ \times \mathbb{R}^n \rightarrow 2^Z, p_d(R, Z, M, d, v) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_d(\varrho, M, d, v)}} \varrho$$

die zugehörige Plausibilisierungsfunktion und heißt *richtungsspezifische globale Distanzfunktion* (Abbildung 3.16, d, e, f).

Befindet sich in der spezifizierten Richtung, ausgehend von einer Region nicht die spezifizierte Menge, zu der die Distanz berücksichtigt werden soll, so liefert das Prädikat zur richtungsspezifischen globalen Distanzfunktion Π_d den Wert “false” (\perp) – eine solche Region wird demnach als nicht-plausibel gewertet.

Die richtungsspezifische globale Distanzfunktion gleicht der globalen Distanzfunktion mit dem Zusatz, dass ausschließlich Distanzen in Richtung eines spezifizierten Richtungsvektors betrachtet werden – der Winkel zwischen dem spezifizierten Richtungsvektor und dem Distanzvektor also 0 ist. Bei der lokalen Variante wird die maximale Distanz wieder ortsabhängig beschrieben:

Definition 42. Seien $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z , außerdem $f(z) \in \mathbb{R}_0^+$ mit $z \in Z$ eine Funktion, die jedem z in der Rekonstruktionszone Z eine maximale Distanz in der Richtung $v \in \mathbb{R}^n$ zu einer nicht-leeren Menge $M \subset \mathbb{R}^n$ zuordnet und $D = \{(z, d) \in Z \times \mathbb{R}_0^+ \mid f(z) = d\}$ die linkstotale und rechtseindeutige Relation, die diese Funktion beschreibt und gelte das Prädikat

$$\Pi_{\hat{D}}(\varrho, M, D, v) := \begin{cases} \inf\{\|u - m\|_2 \mid u \in \varrho, m \in M, \angle(u - m, v) = 0\} \\ \leq \sup\{d \mid (z, d) \in D, z \in \varrho\} \\ , \exists u \in \varrho, \exists m \in M : \angle(u - m, v) = 0 \\ \perp , \text{sonst} \end{cases}$$

wobei $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_{\hat{D}} : 2^Z \times 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \times 2^{Z \times \mathbb{R}_0^+} \times \mathbb{R}^n \rightarrow 2^Z$$

$$p_{\hat{D}}(R, Z, M, D, v) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \varrho \cap \text{Rand}(Z) \neq \emptyset \vee \Pi_{\hat{D}}(\varrho, M, D, v)}} \varrho$$

3. Modellierung

die zugehörige Plausibilisierungsfunktion, welche als *richtungsspezifische lokale Distanzfunktion* bezeichnet wird.

Der Unterschied zur richtungsspezifischen globalen Distanzfunktion besteht nun darin, dass die Distanz zwischen Region und Menge nicht gegenüber einer festen Maximaldistanz überprüft wird, sondern gegenüber dem Maximum aller ortsabhängigen Maximaldistanzen der Positionen in der Rekonstruktionszone, an denen sich die Region befindet.

An dieser Stelle wird deutlich, dass sich mühelos etliche Varianten von Plausibilisierungsfunktionen aus einer Grundidee – hier der Distanz von gesuchten Objekten zu bestimmten Raumbereichen – erzeugen lassen. Beispielsweise könnte man nun noch den Richtungsvektor ortsabhängig beschreiben, worauf allerdings hier aus fehlendem Bezug zur Praxis verzichtet wird.

Es findet nun die Zuordnung der Plausibilisierungsfunktionen zu einem der beiden Typen gemäß Definition 10 statt. Alle Distanzfunktionen folgen dem für regionenbasierte Plausibilisierungsfunktionen typischen Schema (Abbildung 3.12) und sind somit auch *regionenbasiert*. Des Weiteren sind alle Distanzfunktionen *Teilregionen-verwerfend*, da Teilregionen in jedem Fall die gleiche oder eine größere Distanz zu dem spezifizierten Raumbereich besitzen und damit verworfen werden, sofern die zugrundeliegende Region ebenfalls verworfen wird. Außerdem sind alle Distanzfunktionen *vollständig*, da eine weitere Auswertung keine Änderung des Ergebnisses zur Folge hat: Vormals als plausibel ermittelte Regionen, werden wieder die gleichen Distanzen haben, welche eben unter dem spezifizierten Schwellwert liegen und somit weiterhin als plausibel gelten. Aufgrund dieser drei Eigenschaften sind alle Distanzfunktionen vom *Typ R* (Definition 10).

3.2.7. Rand

Je nach Anwendung kann eine Rekonstruktionszone derart klein gewählt werden, dass Regionen gesuchter Objekte gar nicht komplett hineinpassen. Handelt es sich beispielsweise um die Hand- und Arm-Überwachung bei der Zusammenarbeit zwischen einem Mensch und einem Kleinroboter, so kann die Rekonstruktionszone so klein gewählt werden, dass der Mensch nicht komplett in diese Rekonstruktionszone hinein passt und somit immer mit dem Rand der Rekonstruktionszone “verbunden” ist, sobald er hinein greift. Dies lässt sich nun wie folgt als Plausibilisierungsfunktion nutzen: Haben Regionen gesuchter Objekte $U \subset \mathbb{R}^n$ innerhalb der Rekonstruktions-

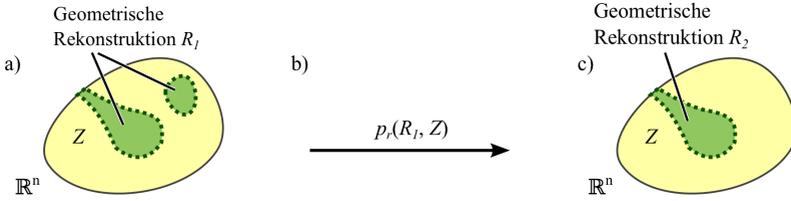


Abbildung 3.17.: Illustration der Randfunktion. Die geometrische Rekonstruktion R_1 (a) wird mittels der Randfunktion (b) in eine geometrische Rekonstruktion R_2 (c) überführt. Es werden diejenigen Regionen verworfen, welche keine Verbindung, d.h. keine Überschneidung mit dem Rand der Rekonstruktionszone Z haben.

zone $Z \subset \mathbb{R}^n$ also die Eigenschaft sich mit dem Rand der Rekonstruktionszone $\text{Rand}(Z)$ zu überschneiden – gilt also $\varrho \cap Z \neq \emptyset \Rightarrow \varrho \cap \text{Rand}(Z) \neq \emptyset$ mit $\varrho \in$ Menge aller Regionen von U , so sind alle Regionen einer geometrischen Rekonstruktion $R \subseteq Z$ von U in Z , welche sich nicht mit dem Rand von Z überschneiden nicht plausibel.

Definition 43. Sei $R \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktion von $U \subset \mathbb{R}^n$ in Z und gelte das Prädikat $\Pi_r(\varrho, Z) := \varrho \cap Z \neq \emptyset \Rightarrow \varrho \cap \text{Rand}(Z) \neq \emptyset$, mit $\varrho \in$ Menge aller Regionen von U , dann ist

$$p_r : 2^Z \times 2^{\mathbb{R}^n} \rightarrow 2^Z, p_r(R, Z) := \bigcup_{\substack{\varrho \in \text{Menge aller Regionen von } R \\ \Pi_r(\varrho, Z)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *Randfunktion* (Abbildung 3.17).

Es findet nun die Zuordnung der Plausibilisierungsfunktion zu einem der beiden Typen gemäß Definition 10 statt. Auch die Randfunktion folgt dem typischen Schema regionenbasierter Plausibilisierungsfunktionen (Abbildung 3.12) und ist daher selbst *regionenbasiert*. Sie ist *Teilregionen-verwerfend*, da jede Teilmenge einer Region, welche sich nicht mit dem Rand überschneidet und daher nicht plausibel ist, ebenfalls als nicht-plausibel ausgewertet wird. Des Weiteren ist die Randfunktion *vollständig*, da vormals als plausibel ermittelte Regionen in einer weiteren Auswertung sich immer noch mit dem Rand der Überwachungszone überschneiden werden und somit weiterhin als plausibel gelten. Die Randfunktion ist folglich vom *Typ R*, da sie alle dafür notwendigen Eigenschaften erfüllt (Definition 10).

Sind in einer Anwendung nur solche Regionen plausibel, welche sich mit dem Rand der Rekonstruktionszone überschneiden, so setzt man die Randfunktion ein. Alle regionenbasierten

3. Modellierung

Plausibilisierungsfunktionen, welche eine Region ohnehin als plausibel bewerten, wenn diese sich mit dem Rand der Rekonstruktionszone überschneidet, tragen nicht weiter zu einer geometrischen Rekonstruktion mit höherer Güte bei und können damit weg gelassen werden.

3.2.8. Zeit

Oftmals kann gesuchten Objekten $U \subset \mathbb{R}^n$ eine Maximalgeschwindigkeit, welche sie nicht überschreiten zugeordnet werden. Dieser Sachverhalt lässt sich insofern nutzen, als dass Regionen einer geometrischen Rekonstruktion, innerhalb eines Zeitintervalls sich nicht beliebig weit von ihrer ursprünglichen Position entfernt haben können und somit Regionen, welche sich nun doch weiter weg befinden, als nicht-plausibel bewertet und damit eliminiert werden können. Überschneiden sich beispielsweise Regionen gesuchter Objekte zu zwei Zeitpunkten, so können alle Regionen einer geometrischen Rekonstruktion, die aus dem "Nichts" entstanden sind – also keine Überschneidung aufweisen – eliminiert werden. Statt die Geschwindigkeit direkt zu verwenden, lässt sich auch einfachheitshalber die Distanz $d \in \mathbb{R}_0^+$ verwenden.

Zusätzlich ist zu beachten, dass Regionen gesuchter Objekte über den Rand der Rekonstruktionszone eindringen können. Dies bedeutet, dass eine solche Region im vorherigen Zeitschritt noch nicht detektiert werden konnte. Wenn sich eine Region aber maximal um eine Distanz d innerhalb eines Zeitintervalls von ihrer ursprüngliche Position entfernen kann, so ist eine neu eingedrungene Region auch nur dann plausibel, wenn sie eine maximale Distanz d zum Rand besitzt.

Definition 44. Seien $R, R_p \subseteq Z \subset \mathbb{R}^n$ die geometrische Rekonstruktionen jeweils von den gesuchten Objekten $U, U_p \subset \mathbb{R}^n$ in Z , außerdem $d \in \mathbb{R}_0^+$ eine Distanz und gelte das Prädikat $\Pi_t(\varrho, \varrho_p, d, Z) := \inf\{\|r - r_p\|_2 \mid r \in \varrho, r_p \in \varrho_p\} \leq d$, wobei $\varrho \in$ Menge aller Regionen von U und $\varrho_p \in$ Menge aller Regionen von U_p , dann ist

$$p_t : 2^Z \times 2^Z \times \mathbb{R}_0^+ \times 2^{\mathbb{R}^n} \rightarrow 2^Z, p_t(R, R_p, d, Z) := \bigcup_{\substack{\varrho \in \text{Menge aller Reg. von } R, \varrho_p \in \text{Menge aller Reg. von } R_p \\ \inf\{\|r - r_p\|_2 \mid r \in \varrho, r_p \in \varrho_p\} \leq d \vee \Pi_t(\varrho, \varrho_p, d, Z)}} \varrho$$

eine Plausibilisierungsfunktion und heißt *Temporalfunktion* (Abbildung 3.18).

Es findet nun die Zuordnung der Plausibilisierungsfunktion zu einem der beiden Typen gemäß Definition 10 statt. Die Temporalfunktion folgt wieder dem bekannten, typischen Schema re-

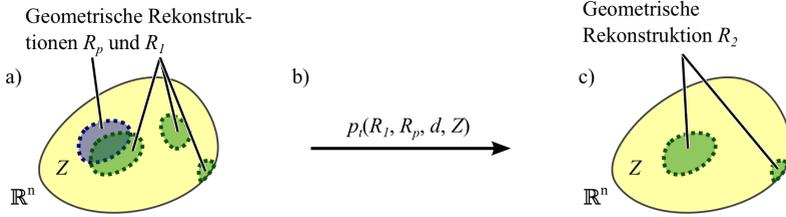


Abbildung 3.18.: Illustration der Temporalfunktion. Die geometrische Rekonstruktion R_1 (a) wird mittels der Temporalfunktion (b) in eine geometrische Rekonstruktion R_2 (c) überführt. Regionen einer geometrischen Rekonstruktion, die in keinem Zusammenhang (Abstand d) mit der vorherigen geometrischen Rekonstruktion stehen, werden verworfen, sofern sie nicht am Rand von Z liegen.

gionenbasierter Plausibilisierungsfunktionen (Abbildung 3.12) und ist daher ebenfalls *regionenbasiert*. Sie ist *Teilregionen-verwerfend*, da Teilmengen zu verwerfender Regionen mindestens den gleichen, wenn nicht sogar einen größeren Abstand zu R_p besitzen und somit auch als nicht-plausibel gelten und verworfen werden. Außerdem ist sie *vollständig*, da vormals als plausibel ermittelte Regionen weiterhin die gleiche Distanz zu R_p besitzen und somit auch weiterhin als plausibel gelten. Aufgrund dieser drei Eigenschaften, welche die Temporalfunktion erfüllt, ist diese vom *Typ R* (Definition 10).

Die Temporalfunktion hat eine gewisse Ähnlichkeit mit einer Sicherheitszaun-Lösung, ist jedoch viel fein-granularer: Angenommen es gäbe einen Raumbereich, der aufgrund seiner Größe und der dort fehlenden Sensorik als plausibel gilt und daher nicht verworfen werden kann, selbst wenn sich kein gesuchtes Objekt darin befindet. Gibt es einen Moment, zu welchem garantiert werden kann, dass sich kein gesuchtes Objekt darin befindet, so wird dieser Raumbereich durch die Temporalfunktion nach diesem Moment so lange als nicht-plausibel markiert, bis ein gesuchtes Objekt aufgrund seiner Nähe hätte "hineinschlüpfen" können. Danach lässt sich der Raumbereich nicht mehr "automatisch" als nicht-plausibel markieren, sondern muss stattdessen explizit wieder freigegeben werden, was der Nutzung eines Sicherheitszaunes gleichkommt – auch hier muss vor einer Freigabe geprüft werden, ob sich kein Mensch innerhalb des abgetrennten Bereiches befindet.

3.2.9. Zusammenfassung

Alle eingeführten Plausibilisierungsfunktionen werden in Tabelle 3.2 zur Übersicht zusammengefasst. Sofort wird deutlich, dass lediglich eine der Plausibilisierungsfunktionen vom Typ E ist, während alle anderen vom Typ R sind. Dies liegt daran, dass die Ausschließbarkeitsfunktion sehr allgemein ist und selbst sehr viele Untervarianten besitzt, um jeweils auf spezielle Weise ausschließbare Elemente zu beschreiben. Eine vergleichbare Zusammenfassung der regionenbasierten Plausibilisierungsfunktionen ist ebenso denkbar und machbar.

3.3. Anwendungsbeispiel: Mensch/Roboter-Koexistenz

Mit den beiden vorangegangenen Kapiteln 3.1 und 3.2 wurden Werkzeuge und Beispiele eingeführt, mit denen man nun in der Lage ist, Wissen und Sensoren zu nutzen um Raumbereiche zu ermitteln, in denen sich gesuchte Objekte aufhalten können. Dieses Kapitel beschreibt ein Anwendungsbeispiel im Bereich der Mensch/Roboter-Koexistenz, in welchem die Bewegungsgeschwindigkeit eines Roboter abhängig vom Abstand zwischen ihm und der wissens- und sensorbasierten geometrischen Rekonstruktion von Menschen angepasst werden soll. Dazu seien folgende anwendungsspezifische Informationen gegeben:

Bekannte Gegenstände (inkl. Geometrie): Roboterarbeitszelle mit Tischen, Regalen, Förderbändern, Boden, Wänden, Decke und Industrieroboter

Rekonstruktionszone: Gesamte Arbeitszelle mit Boden, Wänden und Decke umfassend

Sensoren: Zwei Tiefenkameras, eine Farbkamera, Näherungssensorhaut am Roboter

Gesuchte Objekte: Menschen

- Können sich nicht in bekannten Gegenständen aufhalten,
- auch nicht in sensorbasierten ausschließbaren Raumbereichen
- Volumeninhalt größer als 50l (entspricht etwa 50 kg)
- Befinden sich maximal 70 cm über dem Boden
- In mindestens zwei Sensoren gleichzeitig teilweise als gesuchtes Objekt erfassbar

Plausibilisierungsfunktionen und Ausprägungen	Funktion	Kurzbeschreibung	Typ
Ausschließbarkeitsfunktion über Geometrie von Gegenständen über Sensoren <ul style="list-style-type: none"> • Anwesenheitssensor • Tiefenkamera • Farbkamera via Farben gesuchter Objekte via Hintergrundmodell	$p_a(R, A)$	Schließt Raumbereich A aus geometrischer Rekonstruktion R aus. Der ausschließbare Raumbereich kann unterschiedliche Quellen haben.	E
Erfassbarkeitsfunktion Allgemeine globale ~ Spezielle globale ~ Allgemeine lokale ~ Spezielle lokale ~	$p_e(R, Z, D_1, \dots, D_n, m)$ $p_g(R, Z, D_1, \dots, D_n, m)$ $p_l(R, D_1, \dots, D_n, m)$ $p_{lg}(R, D_1, \dots, D_n, m)$	Verwirft Regionen einer geometrischen Rekonstruktion R (in Z), die von weniger als m Sensoren als gesuchte Objekte erfasst werden. D_1 bis D_n sind die die von n Sensoren als gesuchte Objekte erfassten Mengen.	R
Formfunktion über Hüllen	$p_f(R, Z, M)$	Verwirft Regionen einer geometrischen Rekonstruktion R (in Z), wenn diese nicht aus M herausragen.	R
Volumenfunktion Globale ~ Lokale ~	$p_v(R, Z, v)$ $p_{lv}(R, Z, V)$	Verwirft Regionen einer geometrischen Rekonstruktion R (in Z), wenn diese einen geringeren Volumeninhalt besitzen als v bzw. V .	R
Distanzfunktion Globale ~ Lokale ~ Richtungsspezifische globale ~ Richtungsspezifische lokale ~	$p_d(R, Z, M, d)$ $p_{ld}(R, Z, M, D)$ $p_{gd}(R, Z, M, D, v)$ $p_{lgd}(R, Z, M, D, v)$	Verwirft Regionen einer geometrischen Rekonstruktion R (in Z), wenn diese einen größere Distanz zu M aufweisen als d bzw. D ; ggf. unter Berücksichtigung einer Richtung v .	R
Randfunktion	$p_r(R, Z)$	Verwirft Regionen einer geometrischen Rekonstruktion R (in Z), wenn diese keine Verbindung zum Rand von Z haben.	R
Temporalfunktion	$p_t(R, R_p, d, Z)$	Verwirft Regionen einer geometrischen Rekonstruktion R (in Z), wenn diese weiter als d von einer (vorherigen) geometrischen Rekonstruktion R_p entfernt sind.	R

Tabelle 3.2.: Übersicht der eingeführten Plausibilisierungsfunktionen.

3. Modellierung

- Passen nicht in Hüllen mit Radius 23 cm um Gegenstände
- Maximale Distanz zu sich selbst im vorhergehenden Zeitschritt 10 cm.

Gesucht ist letztlich eine Geschwindigkeitsanpassung des Roboters abhängig vom Abstand zum Menschen. Dazu wird eine geometrische Rekonstruktion erzeugt und darauf aufbauend eine Abstandsberechnung zum Roboter durchgeführt. Der berechnete Abstand dient zur Bestimmung einer geeigneten Bewegungsgeschwindigkeit des Roboters.

Wissens- und sensorbasierte geometrische Rekonstruktion: Bestmögliche geometrische Rekonstruktion von Menschen gemäß oben spezifizierter Informationen.

Mensch-Roboter-Abstand: Abstand zwischen Roboter und wissens- und sensorbasierter geometrischer Rekonstruktion.

Geschwindigkeits-Anpassung: Je geringer der Mensch-Roboter-Abstand, desto langsamer der Roboter. Die Geschwindigkeitsanpassung wird durch einen Skalierungsfaktor im Intervall $[0, 1]$ ausgedrückt, wobei der Roboter bei einem Abstand von 20 cm oder kleiner still stehen soll und bei einer Distanz von 250 cm oder größer mit Maximalgeschwindigkeit verfahren darf.

Es können damit also alle vorgestellten Plausibilisierungsfunktionen, abgesehen von der Randfunktion verwendet werden. Die bestmögliche geometrische Rekonstruktion liegt dann vor, wenn sie plausibel bezüglich aller verwendeter Plausibilisierungsfunktionen ist. In diesem Beispiel wird zusätzlich eine Funktion benötigt, welche einen Abstand zwischen geometrischer Rekonstruktion und Roboter beschreibt und eine weitere, welche daraufbasierend eine Geschwindigkeitsskalierung der Roboterbewegung bewirkt. Zunächst wird schrittweise jede Information verwertet und im Zuge dessen auch die benötigten Symbole eingeführt, welche dann bis zum Ende dieses Kapitels ihre Gültigkeit behalten. Im Anschluss werden dann in aller Kürze die verwendeten Plausibilisierungsfunktionen skizzenhaft zusammengefasst.

Ausschließbarkeitsfunktion über Geometrie bekannter Gegenständen

Seien $Z \subset \mathbb{R}^3$ die Menge, welche die Rekonstruktionszone und $U \subseteq E \subset \mathbb{R}^3$ die Mengen, welche die Umwelt repräsentieren, wobei U die Menge der gesuchten Objekte – also Menschen – und $E \setminus U$ die Menge der bekannten Gegenstände – also Tisch, Regale, Förderbänder, Boden, Wände,

Decke und Industrieroboter – darstellen. Da gesuchte Objekte sich nicht in den Volumina der bekannten Gegenstände aufhalten können, lässt sich die Ausschließbarkeitsfunktion auf die Standardrekonstruktion $R_0 = Z$ anwenden und man erhält eine bezüglich dieser Information plausible geometrische Rekonstruktion R_1 :

$$R_1 = p_a(R_0, E \setminus U)$$

Weiterhin sind verschiedene Sensoren innerhalb der Arbeitszelle angebracht, sodass man deren garantierten Freiraum ebenfalls ausschließen kann:

Ausschließbarkeitsfunktion über sensorbasierte ausschließbare Raumbereiche

Tiefenkameras, Farbkameras und Näherungssensoren lassen sich unter Verwendung einer tiefenbeschreibenden Pixelfunktion auf gleiche Art und Weise nutzen. Insofern wird an dieser Stelle nur auf Kapitel 3.2.2 verwiesen und direkt der ausschließbare Raumbereich der Sensoren verwendet. Die sensorbasierten ausschließbaren Raumbereiche seien $A_1, \dots, A_4 \subseteq Z$ – erste Tiefenkamera, zweite Tiefenkamera, Farbkamera und Näherungssensorhaut. Die Näherungssensorhaut wird hier als ein einziger Sensor aufgefasst. Diese ausschließbaren Raumbereiche werden wieder mithilfe der Ausschließbarkeitsfunktion zur Plausibilisierung verwendet und man erhält schließlich die geometrische Rekonstruktion R_5 :

$$R_2 = p_a(R_1, A_1)$$

$$R_3 = p_a(R_2, A_2)$$

$$R_4 = p_a(R_3, A_3)$$

$$R_5 = p_a(R_4, A_4)$$

Globale Volumenfunktion

Da Regionen gesuchter Objekte mindestens 50l Volumeninhalt haben, und dies für die gesamte Rekonstruktionszone gilt, kann die globale Volumenfunktion verwendet werden. Daraus resultiert dann die geometrische Rekonstruktion R_6 :

$$R_6 = p_v(R_5, Z, 50l)$$

Richtungsspezifische globale Distanzfunktion

Weil Regionen gesuchter Objekte sich maximal 70 cm über den Boden $B \subseteq E \setminus U$ befinden, und auch dieses für die gesamte Rekonstruktionszone gilt, kann die richtungsspezifische globale Distanzfunktion verwendet werden. Es resultiert die geometrische Rekonstruktion R_7 :

$$R_7 = p_d(R_6, Z, B, 70 \text{ cm}, (0, 0, 1))$$

Allgemeine globale Erfassbarkeitsfunktion

Seien $D_1, \dots, D_4 \subseteq Z$ die von den Sensoren als gesuchtes Objekt erfassten Mengen, dann resultiert aus der allgemeinen globalen Erfassbarkeitsfunktion mit dem spezifizierten Wert von 2 die geometrische Rekonstruktion R_8 :

$$R_8 = p_e(R_7, Z, D_1, D_2, D_3, D_4, 2)$$

Form bezüglich Raumbereich über Hüllen

Bestünde die bekannte Umgebung beispielhaft aus nur drei Gegenständen $G_1, G_2, G_3 \subseteq E \setminus U$, $G_1 \cap G_2 = G_1 \cap G_3 = G_2 \cap G_3 = \emptyset$, $G_1 \cup G_2 \cup G_3 = E \setminus U$, so kann die Formfunktion in Verbindung mit Hüllen $H_r^{G_1}, H_r^{G_2}, H_r^{G_3}$ um die drei Gegenstände mit Radius $r = 23 \text{ cm}$ eingesetzt werden. Daraus resultiert schließlich die geometrische Rekonstruktion R_{11} :

$$R_9 = p_f(R_8, Z, H_{23 \text{ cm}}^{G_1})$$

$$R_{10} = p_f(R_9, Z, H_{23 \text{ cm}}^{G_2})$$

$$R_{11} = p_f(R_{10}, Z, H_{23 \text{ cm}}^{G_3})$$

Temporalfunktion

Sei R_{12}^P die geometrische Rekonstruktion zeitlich direkt vor der aktuell zu bestimmenden R_{12} , so kann die Temporalfunktion mit dem angegebenen Wert von 10 cm verwendet werden:

$$R_{12} = p_t(R_{11}, R_{12}^P, 10 \text{ cm}, Z)$$

Distanz und Geschwindigkeitsskalierung

Mit R_{12} steht nun eine geometrische Rekonstruktion der Menschen in der Roboterarbeitszelle zur Verfügung. Benötigt wird nun noch eine Funktion d , mit deren Hilfe ein Abstand, und eine Funktion v mittels derer die Geschwindigkeitsskalierung der Roboterbewegung berechnet werden kann. Seien diese beschrieben durch

$$d : 2^{\mathbb{R}^3} \times 2^{\mathbb{R}^3} \rightarrow \mathbb{R}, \quad d(A, B) := \inf\{\|a - b\|_2 \mid a \in A, b \in B\},$$

$$v : \mathbb{R} \rightarrow \mathbb{R}, \quad v(a) := \begin{cases} 0 & , a < 20 \text{ cm} \\ (a - 20 \text{ cm})/230 \text{ cm} & , 20 \text{ cm} \leq a \leq 250 \text{ cm} \\ 1 & , 250 \text{ cm} < a \end{cases}$$

Zusammenfassung der Funktionen

Alle zur Verfügung stehenden Informationen wurden mithilfe von Plausibilisierungsfunktionen schrittweise genutzt um zu der plausiblen geometrischen Rekonstruktion R_{12} zu gelangen. Da jeweils das Ergebnis einer Plausibilisierungsfunktion, abgesehen von der ersten, für die die Standardrekonstruktion verwendet wird, in die nächste Plausibilisierungsfunktion einfließt, lässt sich unter Hinzunahme des Robotermodells $M \subseteq E \setminus U$ damit in aller Kürze die gesamte Plausibilisierung inklusive Abstandsbestimmung und Geschwindigkeitsskalierung wie folgt in eine Formel fassen:

$$s = v(d(p_t(p_f \dots (p_e(R_0, \dots), \dots), \dots), R_{12}^{-1}, 10 \text{ cm}, Z), M))$$

Diese zusammengesetzte Formel beschreibt eine Geschwindigkeitsskalierung der Roboterbewegung in Abhängigkeit zur kürzesten Distanz zu allen Raumbereichen, in welchen sich Menschen innerhalb der Rekonstruktionszone befinden könnten. Zu beachten ist, dass wenn beispielsweise ein Mensch mit einem geringeren Volumen als dem oben spezifizierten in die gemeinsame Arbeitszelle eindringt, er nicht geschützt ist, da Regionen mit solch kleinem Volumen verworfen werden. In diesem Fall müsste also das Mindestvolumen geeignet nach unten korrigiert werden.

Dieses Beispiel demonstriert die Einfachheit und Allgemeinheit der wissens- und sensorbasierten geometrischen Rekonstruktion und wird später in ähnlicher Form angewendet (Kapi-

tel 4.4 und Kapitel 5).

3.4. Zusammenfassung und Schlussfolgerungen

Zunächst wurden Grundlagen und Werkzeuge zur geometrischen Rekonstruktion und Plausibilisierung im Kontinuierlichen eingeführt. Darauf aufbauend wurde anhand mehrerer praxisrelevanter Plausibilisierungsfunktionen demonstriert, wie sich Wissen und Sensorinformation im Plausibilisierungsprozess integrieren lassen. Im Anschluss daran wurden die eingeführten Plausibilisierungsfunktionen an einem Beispiel im Bereich der Mensch/Roboter-Koexistenz, nämlich der abstandsabhängigen Geschwindigkeitsregelung des Roboters angewendet, was sich schließlich in einer einzigen Formel komfortabel ausdrücken ließ.

Die in diesem Kapitel vorgestellte systematische Herangehensweise nach dem Ausschlussprinzip ermöglicht erstmals die einheitliche Integration unterschiedlichsten Wissens und unterschiedlichster Sensoren, um Raumbereiche geometrisch zu identifizieren, innerhalb derer sich gesuchte Objekte befinden können.

Nach der formalen Abhandlung der wissens- und sensorbasierten geometrischen Rekonstruktion im Kontinuierlichen wird im folgenden Kapitel eine voxelbasierte Umsetzung präsentiert.

4. Umsetzung

Zur Umsetzung eignet sich insbesondere die Diskretisierung des kontinuierlichen dreidimensionalen Raumes mittels Voxeln, da diese aufgrund ihres *Raumelement-Charakters* der mengenbasierten Betrachtungen im Kontinuierlichen (Kapitel 3) sehr nahe kommen und gleichzeitig *in angemessener Zeit* und *robust* verarbeitet werden können. Zeitliche Optimierung wird in diesem Kapitel nur sehr rudimentär und erst im Rahmen eines Ausblicks detailliert diskutiert. Stattdessen werden im ersten Unterkapitel 4.1 Grundlagen und Operationen zum Umgang mit Voxeln präsentiert. Im anschließenden Unterkapitel 4.2 werden die Definitionen zur geometrischen Rekonstruktion und zur Plausibilisierung in voxelbasierte Varianten überführt. Daran schließt sich das Unterkapitel 4.3 an, welches die meisten der im Kontinuierlichen beschriebenen konkreten Plausibilisierungsfunktionen in voxelbasierte überführt und deren Implementierung mittels Pseudocode skizziert. Im Unterkapitel 4.4 wird anhand einer Beispielsequenz experimentell untersucht, wieviele Voxel durch die implementierten Plausibilisierungsfunktionen im realen Umfeld tatsächlich verworfen werden können. Abschließend finden sich in Unterkapitel 4.5 eine Zusammenfassung und Schlussfolgerungen. Die eingeführten Symbole und Bezeichnungen sind in Anhang B zur Übersicht alphabetisch aufgelistet und können dort nachgeschlagen werden.

4.1. Voxel: Grundlagen und Operationen

Voxelräume stellen eine gitterförmige Diskretisierung des kontinuierlichen dreidimensionalen Raumes dar. Sie werden hier über einen minimalen und einen maximalen Punkt im kontinuierlichen dreidimensionalen Raum sowie einer äquidistanten Unterteilung des pro Dimension jeweilig resultierenden Intervalls definiert (Abbildung 4.1):

Definition 45. Sei $b, t \in \mathbb{R}^3$ ein minimaler und ein maximaler Punkt im dreidimensionalen Raum, d.h. $b_i < t_i$, $i \in \{1, 2, 3\}$ und $r \in \mathbb{N}^3$ die Anzahl der äquidistanten Unterteilungen pro

4. Umsetzung

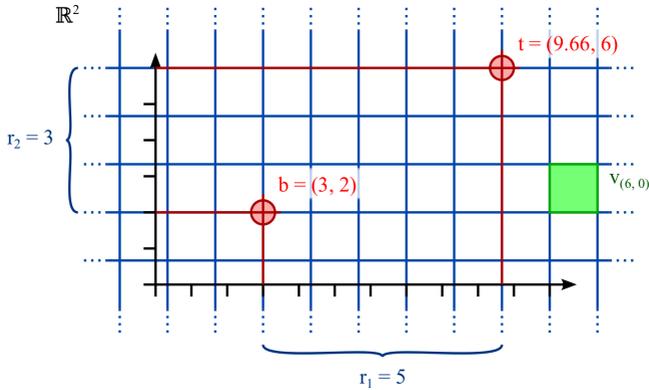


Abbildung 4.1.: Ins Zweidimensionale übertragene, beispielhafte Darstellung der Voxelraumdefinition: Ein Voxelraum ist durch einen Minimum-Punkt b , einen Maximum-Punkt t und eine Anzahl von äquidistanten Unterteilungen pro Dimension (r_1, r_2) innerhalb der resultierenden Intervalle definiert. Einem Voxel kann eine Voxelposition und ein Volumen (grün) zugeordnet werden.

Dimension, dann heißt das Tripel $V = (b, t, r)$ *Voxelraum*.

Eine Zelle dieser gitterförmigen Diskretisierung wird als Voxel bezeichnet. Jedem Voxel lässt sich eine eindeutige Voxelposition und ein quaderförmiges Volumen zuordnen (Abbildung 4.1):

Definition 46. Sei $V = (b, t, r)$ ein Voxelraum, so sind $x \in \mathbb{Z}^3$ die *Voxelpositionen* und $v_x = \{z \in \mathbb{R}^3 \mid b_i + x_i(t_i - b_i) \leq z \leq b_i + (x_i + 1)(t_i - b_i), i \in \{1, 2, 3\}\}$ das *Volumen des Voxels* mit der Voxelposition x . Sei $X \subset \mathbb{Z}^3$ eine Menge von Voxelpositionen, dann ist $v_X = \bigcup_{x \in X} v_x$ das *Volumen der Voxelmenge* mit den Voxelpositionen X .

Mengenoperationen wie beispielsweise Differenz, Schnitt oder Vereinigung können direkt auf den Mengen der Voxelpositionen durchgeführt werden, deren resultierendes zugehöriges Volumen dem Abschluss der gleichen Operation auf den jeweils zugehörigen Volumina dieser Mengen von Voxelpositionen entspricht.

Mit Definition 46 wurde gewissermaßen eine Transformation von einer Voxelrepräsentation zu einer kontinuierlichen Betrachtung im \mathbb{R}^3 gegeben. Aber auch umgekehrt lassen sich solche Transformation definieren, also ausgehend von einer Menge im \mathbb{R}^3 hin zu einer Voxelrepräsentation. Da eine Menge im Kontinuierlichen typischerweise nicht exakt auf die durch Voxel

hervorgerufene diskrete Struktur “passt”, müssen geeignete Transformationen gefunden werden, deren Ergebnisse, zurück ins Kontinuierliche transformiert, eine möglichst gute Approximation der Ausgangsmenge ergibt. Im Folgenden werden zwei naheliegende Transformationen definiert:

Definition 47. Seien $V = (b, t, r)$ ein Voxelraum und $M \subset \mathbb{R}^3$ eine Menge im Kontinuierlichen. Eine Menge $M^{V^+} = \{z \in \mathbb{Z}^3 \mid M \cap \text{Volumen des Voxels mit der Voxelcoordinate } z \neq \emptyset\}$ heißt *vergrößerte Voxelapproximation* von M auf V . Eine Menge $M^{V^-} = \{z \in \mathbb{Z}^3 \mid M \supseteq \text{Volumen des Voxels mit der Voxelcoordinate } z\}$ heißt *verkleinerte Voxelapproximation* von M auf V .

Die vergrößerte Voxelapproximation einer Menge im Kontinuierlichen beinhaltet damit diejenigen Voxel, deren Volumen die Menge M schneidet, während die verkleinerte Voxelapproximation einer Menge im Kontinuierlichen nur diejenigen Voxel enthält, deren Volumen komplett innerhalb M liegen (Abbildung 4.2). Folglich ist das Volumen der vergrößerten Voxelapproximation einer Menge $A \subset \mathbb{R}^3$ immer Obermenge der Menge A selbst und das Volumen der verkleinerten Voxelapproximation der Menge A immer Teilmenge der Menge A selbst. Im Anhangskapitel A werden Algorithmen präsentiert, mit denen man verschiedene geometrische Modelle in solche Voxelapproximationen konvertieren kann.

Ein Voxel hat zunächst drei unterschiedliche Typen von Nachbarn: Bei Flächennachbarn unterscheiden sich nur eine, bei Kantennachbarn zwei und bei Eckennachbarn alle drei Komponenten der Voxelposition jeweils um den Wert 1:

Definition 48. Seien $x, y \in \mathbb{Z}^3$ zwei Voxelpositionen. Sie werden als *Flächennachbarn* bezeichnet, wenn $\exists i, j, k \in \{1, 2, 3\} : i \neq j, j \neq k, k \neq i \wedge |x_i - y_i| = 1, |x_j - y_j| = |x_k - y_k| = 0$ gilt, als *Kantennachbarn*, wenn $\exists i, j, k \in \{1, 2, 3\} : i \neq j, j \neq k, k \neq i \wedge |x_i - y_i| = |x_j - y_j| = 1, |x_k - y_k| = 0$ gilt und als *Eckennachbarn*, wenn $|x_1 - y_1| = |x_2 - y_2| = |x_3 - y_3| = 1$ gilt.

Mittels eines Graphen lässt sich eine Nachbarschaft zwischen Voxeln explizit beschreiben, indem Voxelpositionen als Knoten und Nachbarschaften als Kanten aufgefasst werden. Auf diese Weise lassen sich neben der üblichen Flächen-, Kanten- und Eckennachbarschaft auch beliebig andere Nachbarschaftsbeziehungen modellieren. In Abbildung 4.3 ist in (b) die Kantennachbarschaft explizit modelliert und in (c) modifiziert. Etwas weiter unten wird gezeigt, dass es hilfreich sein kann, aus einer ursprünglich reinen Flächennachbarschaft Kanten zu entfernen. Unter Zuhilfenahme der Modellierung der Nachbarschaftsbeziehung als Graph lässt sich der Begriff der Region (siehe Definition 5) sehr einfach auf Voxelräume übertragen:

4. Umsetzung

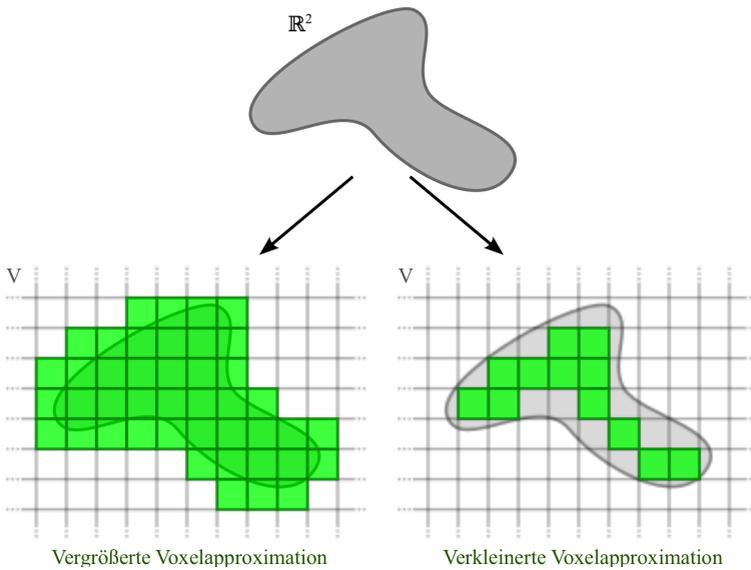


Abbildung 4.2.: Ins Zweidimensionale übertragene, beispielhafte Darstellung der vergrößerten und der verkleinerten Voxelapproximation: Das Volumen der vergrößerten Voxelapproximation ist eine Obermenge und das der verkleinerten Voxelapproximation eine Teilmenge der ursprünglichen Menge.

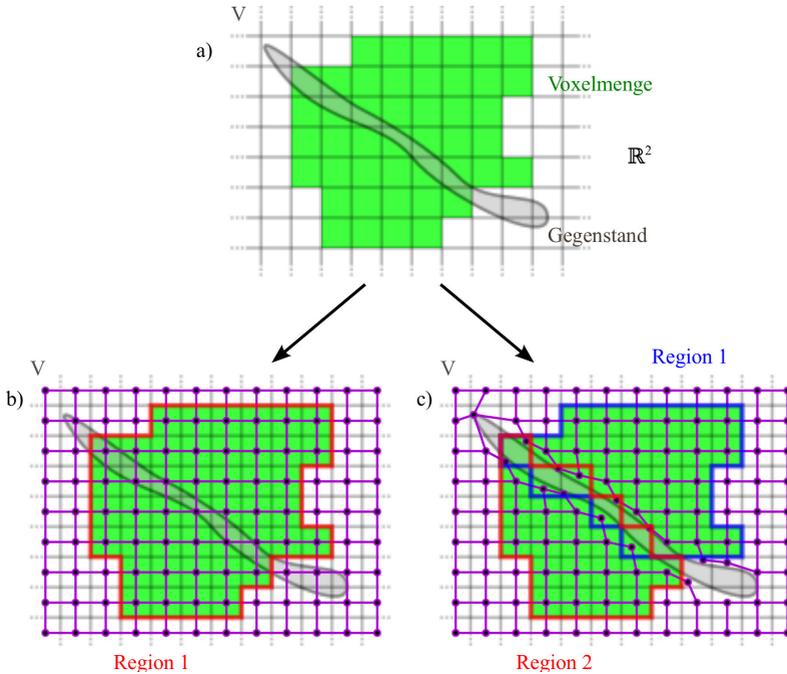


Abbildung 4.3.: Ins Zweidimensionale übertragene, beispielhafte Darstellung unterschiedlicher Voxelnachbarschaften: Teilabbildung a) zeigt eine Voxelmenge (grün) in einem Voxelraum und einen Gegenstand. Teilabbildung b) zeigt eine reine Kantennachbarschaft (im 3-D: eine Flächennachbarschaft), dargestellt mittels des lila-farbenen Graphen. Teilabbildung c) zeigt die gleiche Kantennachbarschaft, welche jedoch in der Nähe des Gegenstandes modifiziert ist. In b) besteht die Voxelmenge aus einer Voxelregion, in c) aus zwei Voxelregionen.

4. Umsetzung

Definition 49. Sei $G = (N^V, E)$ ein Graph, welcher die Nachbarschaftsbeziehungen $E \subseteq N^V \times N^V$ der Voxelpositionen $N^V \subset \mathbb{Z}^3$ beschreibt, und $M^V \subset \mathbb{Z}^3$ eine Menge von Voxelpositionen.

- Eine *Voxelregion* der Menge M^V ist eine zusammenhängende Menge $Q_1^V \subseteq M^V$ in G mit der Eigenschaft Q_1^V maximal, das heißt $\forall Q_2^V \subseteq M$ mit Q_2^V zusammenhängend in G gilt
 1. $Q_2^V \subseteq Q_1^V$ oder
 2. $Q_2^V \cap Q_1^V = \emptyset$
- Eine Menge $Q_M^V = \{Q_1^V \subseteq M^V \mid Q_1^V \text{ Voxelregion von } M^V\}$ heißt *Menge aller Voxelregionen von M^V* .

Ist eine Menge von Voxelpositionen $R \subset \mathbb{Z}^3$ und ein Graph $G = (M, E)$ zur Beschreibung von Nachbarschaftsbeziehungen der Voxelpositionen $M \subset \mathbb{Z}^3$ gegeben, so lässt sich die Menge aller Voxelregionen $P \subseteq 2^R$ von R mittels mehrfacher Anwendung eines Floodfill-Algorithmus ermitteln: Der Algorithmus 4.1 liefert die Menge von Voxelregionen für eine gegebene Menge

Algorithmus 4.1 Ermittlung der Menge aller Voxelregionen P für eine gegebene Menge R von Voxelpositionen auf einem Nachbarschafts-beschreibenden Graphen G .

```

1: procedure CLUSTER( $R, G$ )
2:    $P \leftarrow \text{EMPTYSETOFVOXELREGIONS}$     ▷ Initialisierung der Menge von Voxelregionen
3:   for all  $v \in R$  do                        ▷ Prüfe alle Voxelpositionen in  $R$ 
4:     if  $\forall p \in P : v \notin p$  then          ▷ Falls Voxelposition zu keiner Voxelregion gehörend
5:        $r \leftarrow \text{FLOODFILL}(v, R, G)$     ▷ Ermittle Voxelregion  $r$ , startend mit  $v$ 
6:        $\text{ADDVOXELREGION}(r, P)$               ▷ Füge  $r$  zur Menge aller Voxelregionen  $P$ 
7:     end if
8:   end for
9:   return  $P$ 
10: end procedure

```

an Voxelpositionen zurück. In Zeile 2 wird dazu zunächst eine solche leere Menge initialisiert, die im weiteren Verlauf gefüllt wird. Dann wird über alle Voxel der gegebenen Menge von Voxelpositionen iteriert (Zeile 3) und jeder dieser Voxel auf Zugehörigkeit zu bereits ermittelten Voxelregion überprüft (Zeile 4). Gehört er keiner Voxelregion an, so ist er Teil einer noch nicht

ermittelten Voxelregion, die sodann mittels eines Floodfill-Algorithmus bestimmt wird (Zeile 5) und diese anschließend der Menge von Voxelregionen hinzugefügt wird (Zeile 6).

Der Floodfill-Algorithmus 4.2 ermittelt eine komplette Voxelregion, basierend auf der gegebenen Menge von Voxelpositionen, dem Nachbarschafts-beschreibenden Graphen und einer gegebenen Start-Voxelposition: Die Funktion “Neighbours” in Zeile 8 liefert die Menge der benach-

Algorithmus 4.2 Ermittlung einer Voxelregion r für eine gegebene Menge von Voxelpositionen R auf einem Nachbarschafts-beschreibenden Graphen G , startend mit einer Voxelposition v innerhalb der Voxelregion r .

```

1: procedure FLOODFILL( $v, R, G$ )
2:    $s \leftarrow \text{EMPTYSTACK}()$                                 ▷ Initialisiere leeren Stack
3:    $r \leftarrow \text{EMPTYREGION}()$                                ▷ Initialisiere leere Voxelregion
4:   ADDVOXELPOSITION( $v, r$ )                                   ▷ Füge Voxelposition  $v$  zur Voxelregion  $r$ 
5:   PUSH( $v, s$ )                                               ▷ Lege Voxelposition  $v$  auf Stack  $s$ 
6:   while NOTEMPTY( $s$ ) do                                     ▷ Solange der Stack nicht leer ist
7:      $c \leftarrow \text{POP}(s)$                                      ▷ Nimm oberes Element  $c$ 
8:      $N \leftarrow \text{NEIGHBOURS}(c, G)$                          ▷ Ermittle Nachbarn  $N$  von  $c$  auf Graphen  $G$ 
9:     for all  $n \in N$  do                                       ▷ Prüfe alle Nachbarn
10:      if  $n \in R \wedge n \notin r$  then                       ▷ Falls  $n$  in  $R$  aber noch nicht in Voxelregion  $r$ 
11:        ADDVOXELPOSITION( $n, r$ )                               ▷ Füge  $n$  zur Voxelregion  $r$ 
12:        PUSH( $n, s$ )                                           ▷ Lege  $n$  auf Stack
13:      end if
14:    end for
15:  end while
16:  return  $r$ 
17: end procedure

```

barten Voxelpositionen zurück. Gehört einer dieser Nachbarvoxel noch nicht der Voxelregion an, wohl aber der eingegebenen Menge an Voxelpositionen (Zeile 10), so wird dieser der Voxelregion hinzugefügt (Zeile 11) und zusätzlich auf den Stack gelegt (Zeile 12).

Ob ein Voxel bereits zu einer Voxelregion gehört, lässt sich im Rahmen einer Implementierung schnell prüfen, indem man ein dreidimensionales Feld verwendet, auf welches man direkt per Voxelposition zugreift und darin einen Bitschalter (engl. Flag) setzt bzw. liest. Ist die Anzahl der Nachbarn eines Voxels durch $n \in \mathbb{N}^+$ für alle Voxel der gegebenen Menge von Voxelpositionen beschränkt, so lässt sich der Algorithmus 4.1 mit einem Rechenaufwand von $O(v \cdot n)$

4. Umsetzung

implementieren, wobei $v = |R|$ die Anzahl der Voxelpositionen in R ist. Der Aufwand lässt sich damit erklären, dass man jeden Voxel innerhalb von R markiert – also insgesamt v Stück – und alle seine Nachbarn – also maximal n Stück – prüft. Typischerweise ist die Anzahl der Nachbarn der Voxel konstant, beispielsweise gibt es bei einer Flächennachbarschaft im \mathbb{Z}^3 für jeden Voxel genau sechs Nachbarn. Insofern lässt sich der Aufwand dann mit $O(v)$ angeben.

In Abbildung 4.3 lässt sich schnell nachvollziehen, weshalb eine einfache Nachbarschaftsbetrachtung ausschließlich über Flächen, Kanten und Ecken nicht immer zielführend ist: Hat man innerhalb einer Rekonstruktionszone einen dünnen soliden Gegenstand, so wirkt sich dieser, bei Anwendung der Ausschließbarkeitsfunktion “trennend” auf darüber und darunter liegende Regionen im Kontinuierlichen aus. Im Rahmen der Voxelbetrachtung dürfen allerdings nur Voxel ausgeschlossen werden, die komplett innerhalb des Gegenstandes liegen – also in Abbildung 4.3 kein Einziger. Dies hat zur Folge, dass unter Verwendung einer reinen Flächennachbarschaft keine Trennung in ober- und unterhalb liegende Voxelregionen geschieht und somit eine große Voxelregion, statt zweier kleiner entsteht. Da die große Voxelregion eher plausibel ist als die beiden kleineren, lässt sich bei einer angepassten Nachbarschaft eine (im nachfolgenden Kapitel definierte) voxelbasierte geometrische Rekonstruktion (Definition 51) meist mit höherer Güte (Definition 52) aber mindestens mit gleicher Güte gegenüber einer voxelbasierten geometrischen Rekonstruktion mit reiner Flächennachbarschaft erreichen. Betrachtet man beispielsweise die Volumenfunktion, welche so parametrisiert sei, dass sie Regionen mit einem kleineren Volumen als 501 als nicht-plausibel bewertet und verwirft. Lässt man nun eine Region mit einem Volumen von 801 durch diese Volumenfunktion bewerten, so erweist sich diese Region als plausibel. Ist jedoch diese Region eigentlich durch einen Gegenstand getrennt – liegen also tatsächlich zwei Regionen vor, beispielsweise mit einem Volumen von 201 bzw. 601 – so wird diejenige Region mit dem Volumen von 201 verworfen. Die Güte der resultierenden geometrischen Rekonstruktion hat sich in dem Fall somit erhöht.

Der Rand $\text{Rand}(R)$ einer Menge von Voxelpositionen $R \subset \mathbb{Z}^3$, lässt sich ebenfalls unter Zuhilfenahme eines nachbarschaftsbeschreibenden Graphen $G = (M, E)$ der Voxelpositionen $M \subset \mathbb{Z}^3$ definieren:

Definition 50. Seien $G = (M, E)$ ein Graph, welcher die Nachbarschaftsbeziehungen der Voxelpositionen $M \subset \mathbb{Z}^3$ beschreibt und $R \subset \mathbb{Z}^3$ eine Menge von Voxelpositionen, dann heißt $\text{Rand}(R) := \{r \in R \mid \exists s \notin R : \{r, s\} \in E\}$ *Rand* der Voxelmenge R , wobei $\{r, s\}$ die

Nachbarschaft zweier Voxelpositionen darstellt.

Existiert also ein Nachbarvoxel, welcher nicht zur Voxelmenge gehört, so gehört der betrachtete Voxel zum Rand der Voxelmenge.

4.2. Voxelbasierte geometrische Rekonstruktion und Plausibilisierung

An dieser Stelle werden nun die Konzepte zur wissens- und sensorbasierten geometrischen Rekonstruktion in die voxelbasierte Betrachtung übertragen. Die voxelbasierte geometrische Rekonstruktion lautet demnach:

Definition 51. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine Menge von Voxelpositionen, deren zugehöriges Volumen die Rekonstruktionszone darstellt und $U \subset \mathbb{R}^3$ die Menge, welche die gesuchten Objekte repräsentiert. Eine Menge von Voxelpositionen $R^V \subseteq Z^V$ heißt *voxelbasierte geometrische Rekonstruktion (VGR)* von U in Z^V , falls das Volumen von R^V eine geometrische Rekonstruktion von U im Volumen von Z^V darstellt. Die Menge Z^V wird als *voxelbasierte Rekonstruktionszone* von V bezeichnet (Abbildung 4.4).

Die Definition der voxelbasierten geometrischen Rekonstruktion entspricht der der geometrischen Rekonstruktion (Definition 1), mit dem Unterschied, dass im voxelbasierten Fall die zugehörigen Volumina von Voxelpositionen betrachtet werden. Die voxelbasierte Güte lässt sich dagegen direkt auf der Menge der Voxelpositionen definieren, was natürlich einer Betrachtung auf dem zugehörigen Volumen gleichkommt (vgl. Definition 2).

Definition 52. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone und $R_1^V, R_2^V \subseteq Z^V$ zwei voxelbasierte geometrische Rekonstruktionen von $U \subset \mathbb{R}^3$ in Z^V , dann ist R_1^V von *höherer Güte*, falls $R_1^V \subsetneq R_2^V$, von *geringerer Güte*, falls $R_1^V \supsetneq R_2^V$ und von *gleicher Güte*, falls $R_1^V = R_2^V$ gilt.

Die voxelbasierte Optimalrekonstruktion und voxelbasierte Standardrekonstruktion lassen sich wie folgt formulieren:

Definition 53. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $R^V \subseteq Z^V$ eine voxelbasierte geometrische Rekonstruktion von $U \subset \mathbb{R}^3$ in Z^V und

4. Umsetzung

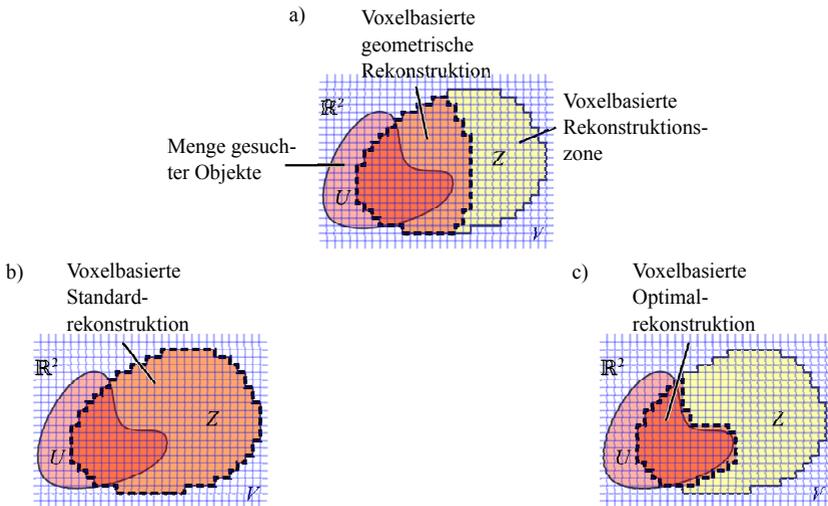


Abbildung 4.4.: Ins Zweidimensionale übertragene, beispielhafte Darstellung der voxelbasierten geometrischen Rekonstruktion (a), der voxelbasierten Standardrekonstruktion (b) und der voxelbasierten Optimalrekonstruktion (c). Die Menge gesuchter Objekte liegt natürlich nach wie vor im Kontinuierlichen vor. Die voxelbasierte Optimalrekonstruktion ist eine Obermenge der kontinuierlichen Optimalrekonstruktion.

$U^{V+} \subset \mathbb{Z}^3$ die vergrößerte Voxelapproximation von U auf V , dann heißt R^V *voxelbasierte Optimalrekonstruktion*, falls $R^V = Z^V \cap U^{V+}$ gilt und *voxelbasierte Standardrekonstruktion*, falls $R^V = Z^V$ gilt (Abbildung 4.4).

Auch hier lässt sich die Ähnlichkeit zu Definition 3 direkt erkennen. Der Unterschied liegt darin, dass ein Voxel der voxelbasierten Rekonstruktionszone bereits dann schon zur Optimalrekonstruktion gehört, sobald ein Teil seines Volumens den gesuchten Objekten angehört. Aus diesem Grund ist die Optimalrekonstruktion im Kontinuierlichen immer eine Teilmenge der voxelbasierten Optimalrekonstruktion.

Die voxelbasierte Plausibilisierungsfunktion sieht zwar genauso aus, wie die im Kontinuierlichen (vgl. Definition 4), abgesehen vom diskretisierten Definitions- und Wertebereich der Ein- und Ausgabe, muss aber im konkreten Fall Zusammenhänge im Kontinuierlichen berücksichtigen, was später bei der Überführung konkreter Beispiele in die voxelbasierten Varianten deutlich wird.

Definition 54. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subset \mathbb{R}^3$ die Menge, welche die unbekanntenen Objekte repräsentiert und I Platzhalter für spezifisches Wissen oder spezifische Sensorinformation (hier nicht weiter ausgeführt).

- Eine Funktion $p^V : 2^{Z^V} \times I \rightarrow 2^{Z^V}$ heißt *voxelbasierte Plausibilisierungsfunktion*, genau dann wenn $\forall R^V \subseteq Z^V$ mit R^V voxelbasierte geometrische Rekonstruktion und $\exists i \in I$ gilt:
 1. $p^V(R^V, i)$ ist wieder voxelbasierte geometrische Rekonstruktion
 2. $p^V(R^V, i) \subseteq R^V$
- Das Tupel (p^V, i) , bestehend aus voxelbasierter Plausibilisierungsfunktion und spezifischer Information $i \in I$ heißt *voxelbasiertes Plausibilisierungstupel*.
- Die Menge $M^V \subseteq Z^V$ heißt *plausibel bezüglich des voxelbasierten Plausibilisierungstupels* (p^V, i) , falls $p^V(M^V, i) = M^V$, $i \in I$ gilt. Dies gilt insbesondere auch für voxelbasierte geometrische Rekonstruktionen.
- Sei $P^V = \{(p_1^V, i_1), \dots, (p_n^V, i_n)\}$ eine Menge von $n \in \mathbb{N}$ voxelbasierten Plausibilisierungstupeln, dann heißt die Menge $M^V \subseteq Z^V$ *plausibel bezüglich der Menge P^V* , falls $\forall (p^V, i) \in P^V : p^V(M^V, i) = M^V$ gilt.

4. Umsetzung

Ebenso wie im Kontinuierlichen lassen sich die voxelbasierten Plausibilisierungsfunktionen, bzw. Plausibilisierungstupel charakterisieren (vgl. Definition 6):

Definition 55. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone und (p^V, i) ein voxelbasiertes Plausibilisierungstupel.

- p^V heißt *elementbasiert*, wenn folgende zwei Bedingungen gelten:

1. $p^V(\bigcup_{m \in M^V} \{m\}, i) = \bigcup_{m \in M^V} p^V(\{m\}, i), \forall M^V \subseteq Z^V$ (Additivität)
2. $p^V(\{z\}, i) \in \{\{z\}, \emptyset\}, \forall z \in Z^V$ (Vernichtung)

- p^V heißt *regionenbasiert*, wenn folgende zwei Bedingungen gelten:

1. $p^V(\bigcup_{\substack{Q \subset M^V \\ \text{Voxelregion}}} Q, i) = \bigcup_{\substack{Q \subset M^V \\ \text{Voxelregion}}} p^V(Q, i), \forall M^V \subseteq Z^V$ (Additivität)
2. $p^V(Q, i) \in \{Q, \emptyset\}, \forall Q \subseteq Z^V$ Voxelregion (Vernichtung)

Statt Raumelemente und Regionen, werden hier Voxel und Voxelregionen betrachtet. Ebenfalls lässt sich die Vollständigkeit (Definition 7) auf voxelbasierte Plausibilisierungsfunktionen übertragen:

Definition 56. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone und (p^V, i) ein voxelbasiertes Plausibilisierungstupel. Die voxelbasierte Plausibilisierungsfunktion p^V heißt *vollständig*, wenn $p^V(M^V, i) = p^V(p^V(M^V, i), i) \forall M^V \subseteq Z$ gilt.

Auch die Eigenschaft 'Teilregionen-verwerfend' (Definition 8) lässt sich direkt in eine voxelbasierte Variante übertragen:

Definition 57. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone und (p^V, i) ein voxelbasiertes Plausibilisierungstupel. Die Plausibilisierungsfunktion p^V heißt

- *Teilregionen-verwerfend für die Menge* $M^V \subseteq Z^V$, wenn $\forall Q^V \subseteq M^V$ mit Q^V Voxelregion die folgende Inklusion gilt: $p^V(Q^V, i) = \emptyset \Rightarrow \forall R^V \subseteq Q^V : p^V(R^V, i) = \emptyset$
- *Teilregionen-verwerfend*, wenn p^V Teilregionen-verwerfend $\forall M^V \subseteq Z^V$

Die Beweise zur Permutierbarkeit von Plausibilisierungsfunktionen gelten in gleicher Weise im Voxelbasierten und werden daher nicht erneut aufgeführt. Da nun die Konzepte der geometrischen Rekonstruktion und der Plausibilisierung im Voxelbasierten vollständig vorliegen, werden im Folgenden die konkreten Plausibilisierungsfunktionen aus Kapitel 3.2 in voxelbasierte Varianten überführt.

4.3. Voxelbasierte Plausibilisierungsfunktionen

Nachdem die grundlegenden Konzepte nun auch für die voxelbasierte Umsetzung zur Verfügung stehen, können die konkreten Plausibilisierungsfunktionen in ihre voxelbasierten Varianten transformiert werden.

4.3.1. Ausschließbarer Raumbereich

Ist im Kontinuierlichen ein Raumbereich $A \subset \mathbb{R}^3$ ausschließbar (vgl. Kapitel 3.2.1), so dürfen im voxelbasierten Fall im Voxelraum V maximal nur diejenigen Voxelpositionen $A^V \subseteq Z^V \subset \mathbb{Z}^3$ einer voxelbasierten Rekonstruktionszone Z^V ausgeschlossen werden, deren zugehöriges Volumen komplett innerhalb des kontinuierlich ausschließbaren Raumbereiches liegt, also $A^V = \{z \in Z^V \mid \text{Volumen des Voxels } z \subseteq A\}$ gilt. Eine Teilmenge von A^V ist daher auch zulässig. Gehören einige der in einem Voxel enthaltenen Raumelemente nicht zum ausschließbaren Raumbereich im Kontinuierlichen, so gehört auch dieser gesamte Voxel nicht zum voxelbasierten ausschließbaren Raumbereich, da es Raumelemente gibt, in denen sich ein gesuchtes Objekt aufhalten könnte. Die Menge A^V ist also (höchstens) eine verkleinerte Voxelapproximation der Menge A auf V .

Definition 58. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $R^V \subseteq Z^V$ eine voxelbasierte geometrische Rekonstruktion von $U \subset \mathbb{R}^3$ in Z^V und sei außerdem $A^V \subseteq Z^V$ mit $\Pi_a^V(U, A^V) := U \cap \text{Volumen der Voxelmenge mit den Voxelkoordinaten } A^V = \emptyset$, dann ist

$$p_a^V : 2^{Z^V} \times 2^{Z^V} \rightarrow 2^{Z^V}, p_a^V(R^V, A^V) := R^V \setminus A^V$$

die zugehörige voxelbasierte Plausibilisierungsfunktion und heißt *voxelbasierte Ausschließbarkeitsfunktion*.

4. Umsetzung

In der voxelbasierten Ausschließbarkeitsfunktion wird somit bereits von einer ausschließbaren *Voxelmenge* ausgegangen. Eine Implementierung der Ausschließbarkeitsfunktion ist in Algorithmus 4.3 aufgelistet. Nur wenn ein Voxel der gegebenen voxelbasierten geometrischen Rekon-

Algorithmus 4.3 Implementierung der Ausschließbarkeitsfunktion. Gegeben ist dafür die voxelbasierte geometrische Rekonstruktion (VGR) $R^V \subseteq Z^V$ und der ausschließbare Raumbereich A^V . Ausgabe des Algorithmus ist die voxelbasierte geometrische Rekonstruktion $R_2^V \subseteq R^V$ gemäß Definition 58.

```
1: procedure AUSSCHLIESSBARKEITSFUNKTION( $R^V, A^V$ )
2:    $R_2^V \leftarrow \text{EMPTYRECONSTRUCTION}()$            ▷ Initialisiere leere Ausgabe-VGR
3:   for all  $r \in R^V$  do                               ▷ Prüfe jeden Voxel in  $R^V$ 
4:     if  $r \notin A^V$  then                             ▷ Falls Voxel  $r$  nicht Teil des ausschließbaren Raumbereichs
5:        $\text{ADDVOXEL}(R_2^V, r)$                          ▷ Füge Voxel  $r$  der VGR  $R_2^V$  hinzu
6:     end if
7:   end for
8:   return  $R_2^V$                                        ▷ Liefere Ausgabe-VGR zurück
9: end procedure
```

struktion nicht dem gegebenen ausschließbaren Raumbereich angehört, ist er auch in der auszugehenden voxelbasierten geometrischen Rekonstruktion enthalten. Dies stellt (wie auch in der Formel der Ausschließbarkeitsfunktion) eine Differenz beider Mengen dar.

Liegt ein ausschließbarer Raumbereich im Kontinuierlichen vor, so muss dieser zunächst in eine Voxeldarstellung entsprechend der oben formulierten Anforderung transformiert werden, also so, dass das zugehörige Volumen der Voxelmenge eine Teilmenge des im Kontinuierlichen gegebenen ausschließbaren Raumbereiches darstellt.

Betrachtet man mehrere im Kontinuierlichen zusammenhängende ausschließbare Raumbereiche, so kann man feststellen, dass deren getrennte Diskretisierung gemäß obiger Anforderung zu nicht-zusammenhängenden ausschließbaren Raumbereichen im Diskreten führt (Abbildung 4.5). Zwischen eigentlich zusammenhängenden ausschließbaren Raumbereichen entsteht eine trennende Voxelschicht, welche selbst wiederum zusammenhängt und sich sehr störend im Rahmen einer weiteren Plausibilisierung auswirkt. Zumal aber die geometrische Rekonstruktion im Voxelraum berechnet wird, ist es nicht möglich "online" eine Vereinigung der kontinuierlichen ausschließbaren Raumbereiche zu berechnen. Handelt es sich um einzelne *statische* Raumbereiche

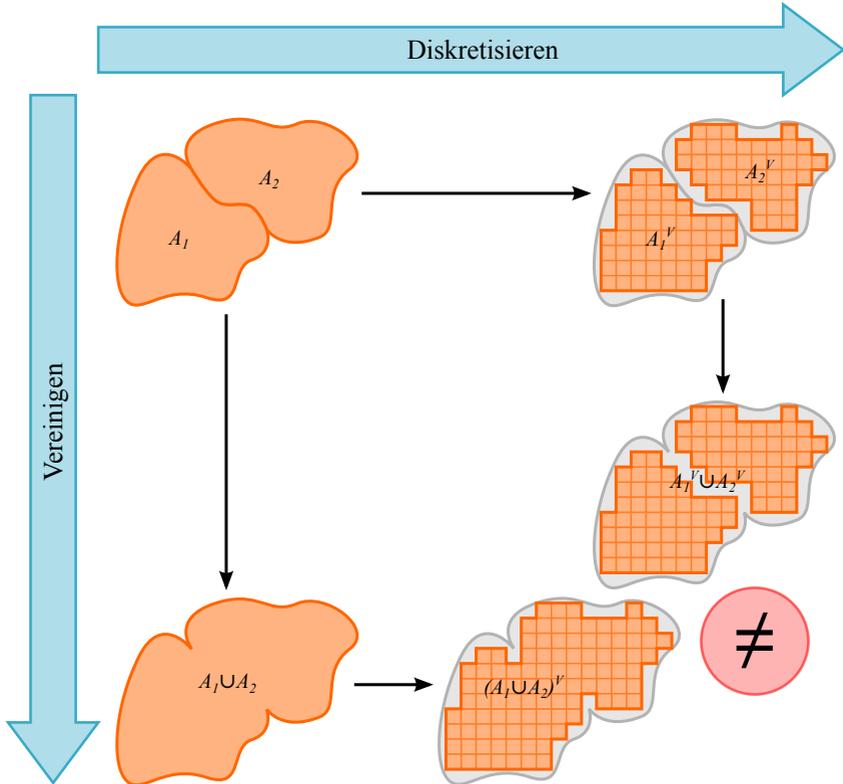


Abbildung 4.5.: Die Operationen *Diskretisieren* und *Vereinigen* sind nicht kommutativ. Wird zunächst diskretisiert und danach vereinigt, so entstehen aus den im Kontinuierlichen zusammenhängenden Mengen A_1 und A_2 im Diskreten zwei Voxelregionen. Wird jedoch zuerst vereinigt und danach diskretisiert, so erhält man eine einzelne Voxelregion. Dieses Ergebnis ist das zu bevorzugende. In vielen Fällen ist es allerdings nicht möglich schnell und robust die Vereinigung im Kontinuierlichen zu ermitteln.

4. Umsetzung

reiche, so können diese natürlich in einer “Offline”-Phase im Vorfeld vereinigt werden, deren Vereinigung diskretisiert und diese schließlich online verwendet werden. Handelt es sich jedoch um dynamische ausschließbare Raumbereiche, also um Raumbereiche, deren Geometrie sich ändert, so ist dieser Weg nicht möglich, außer man kennt im Voraus alle Konfigurationen dieser dynamischen ausschließbaren Raumbereiche und berechnet alle möglichen Vereinigungen ebenfalls im Vorfeld. Ein solcher dynamischer ausschließbarer Raumbereich liegt beispielsweise bei der Verwendung von Sensoren vor: Je nach gemessenem Sensorsignal sieht der ausschließbare Raumbereich anders aus. Insofern ist eine Berechnung der Vereinigung ausschließbarer Raumbereiche im Vorfeld meist *nicht* machbar. Dennoch lässt sich im Fall von Sensoren eine Lösung mittels Überlappung ausschließbarer Raumbereiche angeben, welche in Kapitel 3.2.2 angedeutet wurde und im nachfolgenden Unterkapitel präsentiert wird.

Typischerweise liegen die Geometrieinformationen über die Umwelt in Form von CAD-Daten vor, welche zu einer geeigneten Voxeldarstellung überführt werden müssen. Für den Fall, dass diese Daten in Form von Dreiecksoberflächennetzen gegeben sind, wird im Anhang (Kapitel A.1) konkret beschrieben, wie sich daraus eine solche Voxeldarstellung erzeugen lässt. Da die Erzeugung einer Voxeldarstellung aus Dreiecksoberflächennetzen relativ zeitaufwändig ist, ist die Anwendung von Kugelmodellen bei dynamischen Gegenständen vorteilhaft. Ein Algorithmus zur Transformation von Kugelmodellen zu Voxelmodellen wird im Anhang (Kapitel A.2) vorgestellt. Da jedoch ein Kugelmodell selbst bereits eine approximierende Darstellung ist, muss dessen Beschaffenheit derart sein, dass diese Transformation weiterhin zu einem Voxelmodell führt, dessen Volumen – wie oben beschrieben – eine Teilmenge der kontinuierlichen Darstellung ist. Nur so kann gewährleistet werden, dass die Anwendung der voxelbasierten Ausschließbarkeitsfunktion tatsächlich zu einer voxelbasierten geometrischen Rekonstruktion führt.

Im Folgenden wird der etwas komplexere Fall von Sensoren, insbesondere von Kameras präsentiert und gezeigt, wie sich trennende Voxelschichten vermeiden lassen.

4.3.2. Sensorbasierter ausschließbarer Raumbereich

Zunächst wird der Anwesenheitssensor, danach die Kameras in Bezug auf den ausschließbaren Raumbereich betrachtet.

4.3.2.1. Anwesenheitssensor

Sei $V = (b, t, r)$ ein Voxelraum, $R^V \subset \mathbb{Z}^3$ die Menge der Voxelpositionen, deren zugehöriges Volumen eine geometrische Rekonstruktion beschreibt und $M^V \subseteq \mathbb{Z}^3$ die Menge der Voxelpositionen, deren zugehöriges Volumen im Kontinuierlichen durch eine Anwesenheitsfunktion a auf gesuchte Objekte überwacht wird (Definition 12). Für den voxelbasierten ausschließbaren Raumbereich gilt dann genauso wie im Kontinuierlichen:

$$A_{A_1}^V := \begin{cases} M^V & , a(U, \text{Volumen der Voxelmenge } M^V) = 0 \\ \emptyset & , a(U, \text{Volumen der Voxelmenge } M^V) \neq 0 \end{cases}$$

Wird die Anwesenheit gesuchter Objekte innerhalb des Volumens von M^V detektiert, so kann M^V nicht ausgeschlossen werden, andernfalls schon.

Liegt das vom Anwesenheitssensor überwachte Volumen im Kontinuierlichen $W \subset \mathbb{R}^3$ vor, welches typischerweise nicht durch das Volumen einer Voxeldiskretisierung dargestellt werden kann, so ist – wie bereits oben erläutert – im Diskreten maximal die verkleinerte Voxelapproximation von W auf V zu wählen oder eine Teilmenge davon. Weitere Voxel gegenüber der verkleinerten Voxelapproximation von W auf V wären durch den überwachten Raumbereich des Sensors nicht komplett abgedeckt, sodass durch die Detektion der Abwesenheit gesuchter Objekte Raumbereiche ausgeschlossen würden, in welchen sich gesuchte Objekte aufhalten könnten. Den so ausschließbaren Raumbereich würde man also wie folgt beschreiben:

$$A_{A_2}^V := \begin{cases} W^{V-} & , a(U, W) = 0 \\ \emptyset & , a(U, W) \neq 0 \end{cases},$$

wobei W^{V-} die verkleinerte Voxelapproximation im Voxelraum V von W darstellt.

In beiden Fällen kann auch hier das Problem trennender Voxelschichten, welches in Kapitel 4.3.1 angesprochen wurde auftreten. Dieses Problem kann durch eine zulässige Überlapung mittels Vergrößerung des ausschließbaren Raumbereichs des Anwesenheitssensors um ausschließbare Raumbereiche statischer Gegenstände erreicht werden (siehe dazu auch Kapitel 3.8 und das folgende Kapitel 4.3.2.2).

4. Umsetzung

4.3.2.2. Kameras

Will man voxelbasierte ausschließbare Raumbereiche mittels realen Kameras bestimmen, so muss neben der Diskretisierung durch die Verwendung von Voxelräumen auch eine Diskretisierung im realen Kamerabild berücksichtigt werden. Diese zusätzliche Diskretisierung im Kamerabild hat zur Folge, dass die entsprechend zu diskretisierenden tiefenbeschreibenden Pixelfunktionen gewissen Anforderungen genügen müssen, um letztendlich auch tatsächlich ausschließbare Raumbereiche erzeugen zu können. Dies betrifft gleichermaßen die zu diskretisierenden erweiterten tiefenbeschreibenden Pixelfunktionen. Mit diesen Überlegungen kann schließlich eine Voxelauswertung betrachtet und ein Algorithmus angegeben werden.

Reale Pixel In realen Kameras besteht das Bild aus Pixeln, welchen eine gewisse Ausdehnung auf dem Sensor zugrunde liegt. Dadurch kann ein solches reales Pixel nicht mehr durch einen einzigen Strahl, sondern muss vielmehr durch ein Strahlenbündel beschrieben werden (Abbildung 4.6 a):

Definition 59. Sei $K = (f, S)$ eine kalibrierte Kamera (Definition 14), so heißt $K_r = (f, S_r)$ mit $S_r \subseteq 2^S$ und $\bigcup_{s_r \in S_r} s_r = S$ *kalibrierte reale Kamera* und $s_r \in S_r$ *reales Pixel*.

Reale Pixel sind damit lediglich eine Menge von Pixeln, gemäß der Definition 14 im Kontinuierlichen. Es bedarf hier keiner weiterer Einschränkungen bezüglich der realen Pixel, sodass beispielsweise zwei reale Pixel einen gemeinsamen Pixel gemäß Definition 14 haben könnten, was eine Art Überlappung bedeuten würde.

Tiefenbeschreibende Real-Pixelfunktion Ausgehend von einer tiefenbeschreibenden Pixelfunktion im Kontinuierlichen (Definition 15) lässt sich für den diskretisierten Fall die tiefenbeschreibende Real-Pixelfunktion definieren:

Definition 60. Sei $i : 2^{\mathbb{R}^3} \times \mathbb{R}^3 \times S \rightarrow \bar{\mathbb{R}}^+$ eine tiefenbeschreibende Pixelfunktion im dreidimensionalen Raum, so ist $i_r : 2^{\mathbb{R}^3} \times \mathbb{R}^3 \times 2^S \rightarrow \bar{\mathbb{R}}^+$, $i_r(E, f, s_r) := \inf\{i(E, f, s) \mid s \in s_r\}$ die zugehörige *tiefenbeschreibende Real-Pixelfunktion*.

Es wird das Minimum der tiefenbeschreibenden Pixelfunktion aller Pixel des Real-Pixels als Real-Pixelwert verwendet (Abbildung 4.6 b). Der Grund dafür ist naheliegend: Bis zu dieser Distanz, ausgehend vom Brennpunkt der Kamera, entlang aller enthaltenen Pixel befindet sich

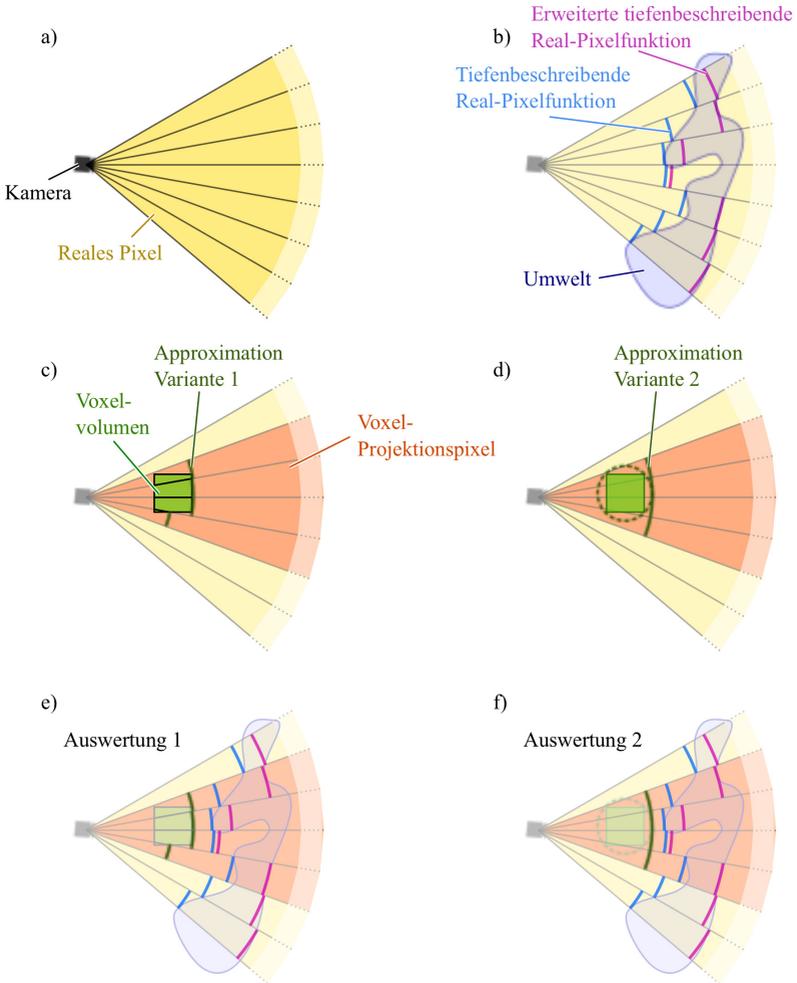


Abbildung 4.6.: Illustration einer realen Kamera und realen Pixeln (a), der (erweiterten) tiefenbeschreibenden Real-Pixelfunktion im Kontext einer Umwelt (b), einer Voxelapproximation nach Variante 1 (c) und Variante 2 (d) sowie der Auswertung unter zuhelfenahme der (erweiterten) tiefenbeschreibenden Real-Pixelfunktion für Variante 1 (e) und Variante 2 (f).

4. Umsetzung

garantiert kein Gegenstand. Für jeden größeren Wert trifft dies nicht zu. Dieser Sachverhalt lässt sich einfacher mittels des im Folgenden definierten rückprojizierten freien Volumens eines realen Pixels beschreiben (Abbildung 4.7):

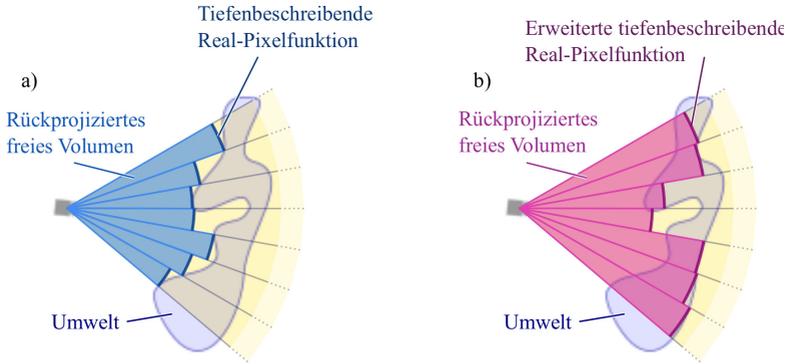


Abbildung 4.7.: Illustration des rückprojizierten freien Volumens eines jeden Pixels der tiefenbeschreibenden Real-Pixelfunktion (a) und der erweiterten tiefenbeschreibenden Real-Pixelfunktion (b).

Definition 61. Seien $K_r = (f, S_r)$ eine kalibrierte reale Kamera, $E \subset \mathbb{R}^3$ eine Umwelt und $\tilde{i}_r : 2^{\mathbb{R}^3} \times \mathbb{R}^3 \times S_r \rightarrow \overline{\mathbb{R}^+}$ eine (ggf. erweiterte) tiefenbeschreibende Real-Pixelfunktion im dreidimensionalen Raum, so heißt

$$V_{s_r} = \{z \in \mathbb{R}^3 \mid \exists s \in s_r \exists x \in \mathbb{R}^+ : z = f + x \cdot s \text{ mit } x < \tilde{i}_r(E, f, s_r)\}$$

rückprojiziertes freies Volumen des realen Pixels $s_r \in S_r$.

Verwendet man nun eine reale Tiefenkamera, so lässt sich eine Subpixel-genaue Betrachtung nicht ohne Weiteres durchführen. Wird beispielsweise eine Oberfläche auf einen realen Pixel projiziert, deren Distanz innerhalb dieser Projektion stark variiert, so erhält man dennoch nur einen einzigen Tiefenwert für den gesamten realen Pixel. Aus diesem Grund besteht die Anforderung an die Tiefenkamerahardware selbst, dass sie in der Lage sein muss, die tiefenbeschreibende

Real-Pixelfunktion gemäß der Definition zu erfüllen und eine Distanz pro realem Pixel zu liefern, deren rückprojiziertes freies Volumen tatsächlich keinen Gegenstand enthält.

Auch bei Farbkameras sind Subpixel-genaue Betrachtungen nicht ohne Weiteres machbar – man erhält nur einen einzigen Farbwert für den gesamten realen Pixel. Da bei Farbkameras in beiden vorgestellten Methoden (Definition 21 und Definition 25) jeweils über eine Segmentierung des Bildes eine tiefenbeschreibende Pixelfunktion entwickelt wurde, muss das Segmentierungsverfahren schon dann ein gesuchtes Objekt für den gesamten Real-Pixel detektieren, wenn dieses nur auf einen kleinen Teil projiziert wird. In diesem Fall wird in beiden tiefenbeschreibenden Pixelfunktionen ein Wert von 0 für den Pixel gesetzt.

Genaugenommen handelt es sich bei diesen “realen” Kameramodellen immer noch um optimale Kameras, bei denen Rauschen und Messfehler nicht berücksichtigt werden. Solche Untersuchungen liegen jedoch nicht im Fokus dieser Arbeit und werden auch daher nicht weiter erörtert. Sie sind vor allem aber dann notwendig, wenn ein solches System *sicher* gestaltet werden soll, wie es im Bereich der Mensch/Roboter-Koexistenz zumeist benötigt wird (vergleiche Ausblick, Kapitel 7.5).

Erweiterte tiefenbeschreibende Real-Pixelfunktion Für die erweiterten tiefenbeschreibenden Pixelfunktionen gilt gleiches wie für die normalen tiefenbeschreibenden Pixelfunktionen:

Definition 62. Sei $\hat{i} : 2^{\mathbb{R}^3} \times \mathbb{R}^3 \times S \rightarrow \bar{\mathbb{R}}^+$ eine erweiterte tiefenbeschreibende Pixelfunktion im dreidimensionalen Raum, so ist $\hat{i}_r : 2^{\mathbb{R}^3} \times \mathbb{R}^3 \times 2^S \rightarrow \bar{\mathbb{R}}^+$, $\hat{i}_r(E, f, s_r) := \inf\{\hat{i}(E, f, s) \mid s \in s_r\}$ die zugehörige *erweiterte tiefenbeschreibende Real-Pixelfunktion*.

Es wird wieder das Infimum aller im realen Pixel enthaltenen Pixelwerte aus gleichem Grund wie oben benötigt (Abbildung 4.6 b): Nur so ist gewährleistet, dass sich im rückprojizierten freien Volumen kein Gegenstand befinden kann.

Voxelauswertung und voxelbasierter ausschließbarer Raumbereich Ein Voxel mit einer Position x in einem Voxelraum muss entweder als komplett *frei* oder als *potentiell gesuchtes Objekt enthaltend* ausgewertet werden und kann, falls er als frei ausgewertet wird, dem ausschließbaren Raumbereich hinzugefügt werden. Ein Voxel wird als frei ausgewertet, falls alle in seinem zugehörigen Volumen v_x enthaltenen Raumelemente als frei ausgewertet werden können, also $\forall z \in v_x, \exists s_r \in S_r : z \in V_{s_r}$, mit V_{s_r} rückprojiziertes freies Volumen des realen Pixels s_r

4. Umsetzung

gilt. Gibt es einen Teil des Voxelvolumens, welcher auf *keinen einzigen* Pixel projiziert wird, so kann dieser Voxel offensichtlich *nie* als frei ausgewertet werden.

Um nun die explizite Betrachtung einzelner Raumelemente zu vermeiden, bietet sich an, die geometrischen Zusammenhänge im Dreidimensionalen zu berücksichtigen: So liegen die Elemente des rückprojizierten Volumens eines realen Pixels innerhalb eines Kugelsektors, und das Volumen eines Voxels ist quaderförmig. Außerdem wird im Allgemeinen das Volumen eines Voxels auf mehrere reale Pixel projiziert – und nicht nur auf einen einzigen Pixel. Verändert sich zudem die Kameraplatzierung gegenüber dem Voxelraum nicht, lassen sich verschiedene Vorberechnungen anstellen, die schließlich eine konservative und schnelle Voxelauswertung ermöglichen. Im Folgenden werden zwei Varianten beschrieben, um eine solche Voxelauswertung vorzunehmen.

In der ersten Variante (Abbildung 4.6 c) wird für alle Voxel vorberechnet, auf welche realen Pixel sie projiziert werden (orange-farbener Bereich) und wie weit das dazugehörige Projektionsvolumen (rechte, dunkel-grüne Voxelbegrenzung) maximal vom Brennpunkt der Kamera entfernt ist. Werden Teile des Voxels auf kein reales Pixel projiziert, so wird dieser Voxel als nicht auswertbar markiert. Pro Voxel erhält man also eine Liste von Paaren, jeweils bestehend aus realem Pixel und Distanz und zudem einem Auswertbarkeits-Bitschalter (Flag). Ein Voxel kann dann als frei ausgewertet werden und dem voxelbasierten ausschließbaren Raumbereich hinzugefügt werden, wenn dieser Voxel gemäß des Bitschalters auswertbar ist und für jedes reale Pixel die Voxel­distanz kleiner ist als die Distanz der tiefenbeschreibenden Real-Pixelfunktion bzw. der erweiterten tiefenbeschreibenden Real-Pixelfunktion. Dies ist dann der Fall, wenn in Abbildung 4.6 e alle grünen Voxelbegrenzungen links von den zugehörigen blauen, bzw. roten visualisierten Tiefenwerten liegen. Würde eine Voxelbegrenzung rechts davon liegen, so hieße dies, dass für einen Teil des Voxelvolumens kein Freiraum garantiert werden kann.

In der zweiten Variante (Abbildung 4.6 d), wird lediglich vorberechnet, auf welche realen Pixel ein Voxel projiziert wird (orange-farbener Bereich). Außerdem wird wieder wie oben berechnet, ob ein Voxel auswertbar ist. Pro Voxel erhält man also nun eine Liste von realen Pixeln und dem Bitschalter zur Auswertbarkeit. Um nun zu bestimmen, ob ein Voxel als frei ausgewertet und dem voxelbasierten ausschließbaren Raumbereich hinzugefügt werden kann, wird eine Maximaldistanz (rechte, dunkel-grüne Voxelbegrenzung) des Voxels über seine Umkugel ermittelt (gestrichelter, dunkel-grüner Kreis). Ist diese Maximaldistanz des Voxels kleiner als alle Werte der realen Pixel in der Liste, so gilt der Voxel als frei – in Abbildung 4.6 f muss also die grüne

Voxelbegrenzung links von allen blauen, bzw. roten visualisierten Tiefenwerten der Projektionspixel liegen. Diese zweite Variante, welche sowohl eine akzeptable Ausführungsgeschwindigkeit als auch einen vertretbaren Speicherplatzverbrauch gewährleistet, wird im folgenden Algorithmus zur Ermittlung des kamerabasierten ausschließbaren Raumbereichs genutzt.

Algorithmus Nachdem nun bekannt ist, wie die (erweiterte) tiefenbeschreibende Real-Pixelfunktion beschaffen sein muss und wie ein Voxel ausgewertet werden kann, lässt sich der Algorithmus 4.4 zur Ermittlung des voxelbasierten ausschließbaren Raumbereichs angeben. Vom

Algorithmus 4.4 Implementierung zur Ermittlung des ausschließbaren Raumbereiches einer Kamera. Gegeben ist dafür der Brennpunkt f einer Kamera, das aktuelle Bild I , in welchem für jedes reale Pixel der zugehörige Tiefenwert mittels einer (ggf. erweiterten) tiefenbeschreibenden Real-Pixelfunktion eingetragen wurde, die Datenstruktur D , welche jedem Voxel neben einem Auswertbarkeits-Bitschalter auch eine Liste von Projektionspixeln zuordnet und eine Voxelmenge V , deren Voxel auf Abwesenheit gesuchter Objekte geprüft werden. Ausgabe des Algorithmus ist eine Teilmenge $A \subseteq V$, deren Voxel dem ausschließbaren Raumbereich angehören.

```

1: procedure KAMERABASIERTERAUSSCHLIESSBARERRAUMFUNKTION( $f, I, D, V$ )
2:    $A \leftarrow$  EMPTYVOXELSET()           ▷ Initialisiere leeren ausschließbaren Raum
3:    $d \leftarrow$  VOXELDIAGONALLENGTH/2   ▷ Ermittle halbe Länge der Voxeldiagonalen
4:   for all  $q \in V$  do                   ▷ Prüfe jeden Voxel in  $V$ 
5:     if not EVALUATABLE( $D, q$ ) then     ▷ Falls Auswertbarkeits-Bitschalter false
6:       CONTINUE                         ▷ Werte nächsten Voxel aus
7:     end if
8:      $m \leftarrow$  ||VOXELCENTER( $q$ ) -  $f$ ||2 +  $d$    ▷ Ermittle maximale Voxelvolumen-Distanz
9:      $a \leftarrow$  true                     ▷ Annahme, das Voxel frei
10:    for all  $p \in$  PROJECTIONPIXELS( $D, q$ ) do   ▷ Prüfe ihn in allen Projektionspixeln
11:      if  $m \geq$  VALUE( $I, p$ ) then           ▷ Falls konservatives Voxelvolumen hinter Tiefe
12:         $a \leftarrow$  false                   ▷ Voxelvolumen doch nicht komplett frei
13:      BREAK
14:    end if
15:  end for
16:  if  $a$  then                               ▷ Falls Voxelvolumen komplett frei
17:    ADDVOXEL( $A, q$ )                         ▷ Füge Voxel zu ausschließbarem Raum  $A$  hinzu
18:  end if
19: end for
20: return  $A$                                 ▷ Liefere ausschließbaren Raum zurück
21: end procedure

```

4. Umsetzung

Prinzip wird jeder einzelne Voxel einer gegebenen Voxelmenge V auf Abwesenheit gesuchter Objekte geprüft und in jenem Fall dem voxelbasierten ausschließbaren Raumbereich A hinzugefügt. Das Voxelvolumen wird dabei in all seinen Projektionspixeln auf die relative Lage der für jeden dieser Real-Pixel eingetragene Tiefenwerte konservativ – wie oben beschrieben – untersucht.

Diskussion Vergleicht man die Resultate der tiefenbeschreibenden und der erweiterten tiefenbeschreibenden Pixelfunktionen, so kann man auch hier sehen, dass zunächst trennende Schichten entstehen (Abbildung 4.8). In Teilabbildung (a) werden beide Raumbereiche A_1 und A_2 dargestellt. Teilabbildung (e) und (h) enthalten den Raumbereich A_1 in herkömmlicher Ausprägung bzw. gemäß der erweiterten tiefenbeschreibenden Pixelfunktion im Kontinuierlichen. Teilabbildung (b) zeigt den Raumbereich A_2 im Kontinuierlichen. Die Diskretisierung der Raumbereiche A_1 und A_2 (c, d, f) resultiert in getrennten Voxelregionen (g). Durch die erweiterten tiefenbeschreibenden Pixelfunktionen erreicht man eine Überlappung seitens des (via Sensor bestimmten) ausschließbaren Raumbereichs, sodass die Problematik einer solchen trennenden Schicht massiv reduziert werden kann (j). Der Nachteil liegt jedoch darin, dass dadurch der sensorbasierte ausschließbare Raumbereich nicht mehr für sich alleine steht, sondern zusätzlich noch die Information über ausschließbare Raumbereiche über Volumina bekannter (statischer) Gegenstände benötigt. Diese Abhängigkeit war im Kontinuierlichen nicht gegeben.

4.3.3. Erfassbarkeit

Lässt sich voraussetzen, dass Regionen gesuchter Objekte in einer Mindestanzahl von Sensoren als solche innerhalb der Rekonstruktionszone erfasst werden, so lassen sich alle Regionen einer geometrischen Rekonstruktion verwerfen, die von weniger als dieser Mindestanzahl von Sensoren als gesuchte Objekte erfasst werden. Was dabei *erfassen* bedeutet, wurde mittels der ausschließbaren Raumbereiche eines Sensors in einer Umwelt ohne und mit gesuchten Objekten definiert (Definition 29, Abbildung 4.9, a, b, c).

Die Erfassbarkeit ist nicht ganz problemlos auf Voxelräume und reale Kameras übertragbar. Der Grund hierfür liegt in der Diskretisierung des Raumes und des Bildes und wird im Folgenden diskutiert. In beiden Fällen wird es zu zwei Alternativen kommen, wobei sich im Anschluss Gründe angeben lassen, welche Alternative jeweils die zu bevorzugende ist.

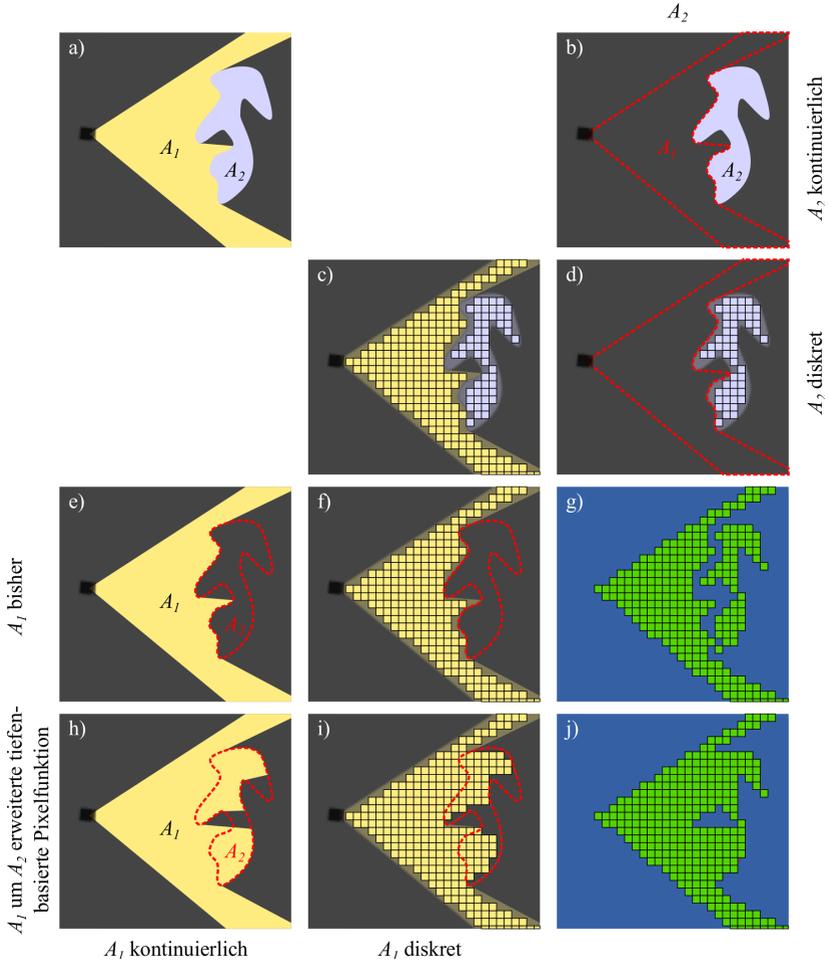


Abbildung 4.8.: Vergleich der tiefenbeschreibenden und der erweiterten tiefenbeschreibenden Pixelfunktion im Kontext einer voxelbasierten Umsetzung. Im Kontinuierlichen zusammenhängende Raumbereiche (a) werden durch die Diskretisierung getrennt (g). Erst eine Überlappung des Raumbereiches A_1 mit dem Raumbereich A_2 (h) entschärft die Situation, sodass im diskreten Fall ebenfalls zusammenhängende Raumbereiche entstehen (j).

4. Umsetzung

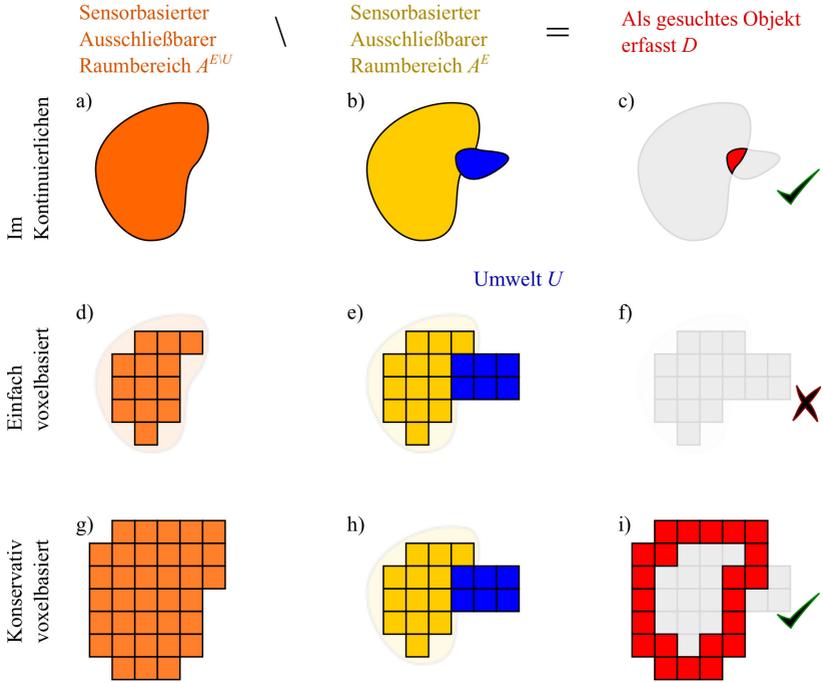


Abbildung 4.9.: Darstellung der Menge D , welche vom Sensor als gesuchtes Objekt erfasst wird (c, f, j), basierend auf dem sensorbasierten ausschließbaren Raumbereich $A^{E \setminus U}$ (a, d, g) und A^E (b, e, h). Im einfach voxelbasierten Fall, wird das gesuchte Objekt nicht als solches erfasst (f), im konservativ voxelbasierten schon (j). Bei Untersuchung der Erfassbarkeit einer Region, wird in der konservativ voxelbasierten Variante die Anzahl der Sensoren, die das gesuchte Objekt erfassen in jedem Fall korrekt um eins erhöht.

4.3.3.1. Diskretisierung durch Voxelräume

Zunächst wird die Definition des *Erfassens* direkt auf Voxel übertragen:

Definition 63. Seien $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subseteq E \subset \mathbb{R}^3$ Mengen, welche die Umwelt beschreiben, wobei U die Menge der gesuchten Objekte und $E \setminus U$ die Menge der bekannten Gegenstände darstellt. Seien $A^{V, E \setminus U}, A^{V, E} \subseteq Z^V$ verkleinerte Voxelapproximationen ausschließbarer Raumbereiche eines Sensors (vgl. Kapitel 4.3.1) – ohne Erweiterung um sonstige ausschließbare Raumbereiche – einzig und allein unter Berücksichtigung der bekannten Gegenstände $E \setminus U$, bzw. der gesamten Umwelt E . Die Menge $D^V = A^{V, E \setminus U} \setminus A^{V, E}$ beschreibt Voxel, welche vom Sensor als *gesuchte Objekte einfach voxelbasiert erfasst* werden (Abbildung 4.9, d, e, f).

Diese Definition unterscheidet sich von Definition 29 dadurch, als dass nicht mehr die kontinuierlichen ausschließbaren Raumbereiche verwendet werden, sondern die daraus berechenbaren voxelbasierten ausschließbaren Raumbereiche. Bestimmt man nun den Parameter, welcher die Anzahl von Sensoren, in welchen Regionen gesuchter Objekte mindestens als solche erfasst werden im Kontinuierlichen, so ist dieser nach der Diskretisierung nicht mehr zulässig, da dadurch Regionen gesuchter Objekte von weniger als dieser Mindestanzahl von Sensoren als gesuchtes Objekt erfasst werden könnten (Abbildung 4.9, d, e, f). Der Grund hierfür liegt darin, dass das Volumen der Voxelmenge D^V eine Teilmenge der im Kontinuierlichen Menge D ist: Der im Kontinuierlichen ausschließbare Raumbereich ist größer als der voxelbasierte. Es könnte also passieren, dass eine Region in D , aber nicht in D^V liegt (Abbildung 4.9, c, f).

Dem lässt sich entgegenwirken, indem man die *vergrößerte* Voxelapproximation von $A^{E \setminus U}$ verwendet und davon wie gehabt lediglich die verkleinerte Voxelapproximation von A^E abzieht (Abbildung 4.9, g, h, i). Man erreicht dadurch, dass genau die Raumbereiche im Kontinuierlichen in welchen Regionen gesuchter Objekte durch die Diskretisierung nicht berücksichtigt würden, nun doch berücksichtigt werden, indem sie *immer* als gesuchte Objekte interpretiert werden (Abbildung 4.9, i). Dies ist eine konservative Betrachtung, da dadurch die Anzahl an Sensoren, welche eine Region gesuchter Objekte als solche erfassen immer mindestens genauso hoch ist wie bei einer kontinuierlichen Betrachtung. Dazu sei definiert:

Definition 64. Seien $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subseteq E \subset \mathbb{R}^3$ Mengen, welche die Umwelt beschreiben, wobei U die Menge der gesuchten Objekte und $E \setminus U$ die Menge

4. Umsetzung

der bekannten Gegenstände darstellt. Seien $A^{V,E\setminus U} \subseteq Z^V$ vergrößerte Voxelapproximation und $A^{V,E} \subseteq Z^V$ verkleinerte Voxelapproximationen ausschließbarer Raumbereiche eines Sensors (vgl. Kapitel 4.3.1) – ohne Erweiterung um sonstige ausschließbare Raumbereiche – einzig und allein unter Berücksichtigung der bekannten Gegenstände $E \setminus U$, bzw. der gesamten Umwelt E . Die Menge $D^V = A^{V,E\setminus U} \setminus A^{V,E}$ beschreibt Raumelemente, welche vom Sensor als *gesuchte Objekte konservativ voxelbasiert erfasst* werden (Abbildung 4.9, g, h, i).

Der Erfassbarkeitsparameter, also die Mindestanzahl an Sensoren, in welchen Regionen gesuchter Objekte als solche erfasst werden, lässt sich durch diese Definition unabhängig von einer Voxelraumauflösung anwenden.

Legt man jedoch das Volumen der Voxelmenge der verkleinerten Voxelapproximation von $A^{E\setminus U}$ zur Bestimmung des Erfassbarkeitsparameters zugrunde, so ist es natürlich zulässig die erste Definition zu verwenden. Nachteil dabei ist allerdings, dass der Erfassbarkeitsparameter von der Voxelraumauflösung abhängig ist und bei einer Änderung dieser neu bestimmt werden muss.

Man hat nun also die Wahl, ob man den Parameter, entsprechend unter Berücksichtigung der Voxelraumauflösung festlegt oder ob man den im Kontinuierlichen bestimmten Parameter konservativ nutzen will. Die zu bevorzugende Alternative wird im Anschluss an die nun folgende Betrachtung realer Kameras mit realen Pixeln erörtert.

4.3.3.2. Diskretisierung durch reale Kameras

Reale Pixel haben eine gewisse Ausdehnung auf dem Sensor der realen Kamera (Definition 59). Aus diesem Grund war es notwendig die im Kontinuierlichen vorliegenden tiefenbeschreibenden Pixelfunktionen auf tiefenbeschreibende Real-Pixelfunktionen im Diskreten mittels einer Minimumsbestimmung anzupassen (Kapitel 4.3.1), was natürlich Auswirkungen auf den ausschließbaren Raumbereich hat, der direkt zur Definition des *Erfassens* verwendet wurde. In Abbildung 4.10 werden die Folgen einer solchen Betrachtung verglichen: Im Fall einer Tiefenkamera (Abbildung 4.10 a-f) könnte sich ein gesuchtes Objekt in einer Vertiefung der Umwelt aufhalten (Abbildung 4.10 b, e), ohne dass es eine Auswirkung auf das Resultat der tiefenbeschreibenden Real-Pixelfunktion hätte (Abbildung 4.10, f). Anders jedoch bei einer Farbkamera (Abbildung 4.10 g-l) – deren tiefenbeschreibende Real-Pixelfunktion würde mit den beiden vorgestellten Verfahren nun für die Real-Pixel auf die das gesuchte Objekt projiziert wird einen Wert

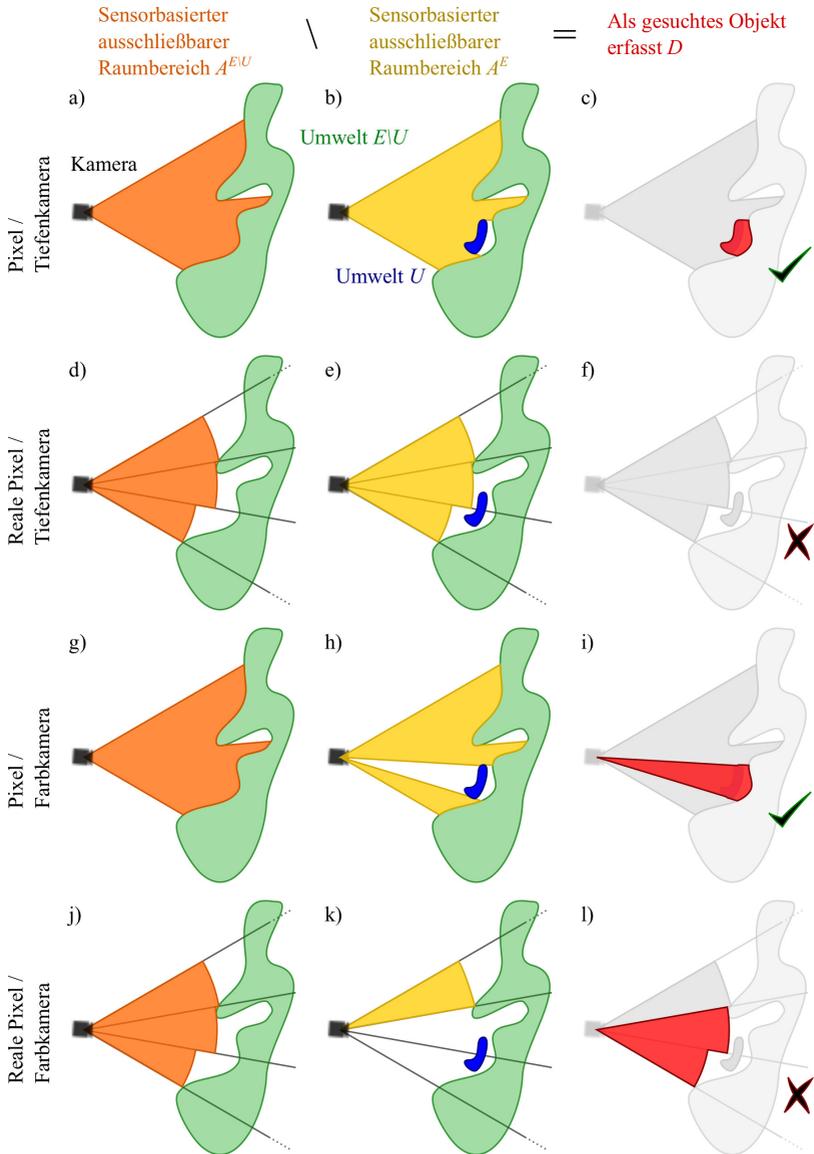


Abbildung 4.10.: Darstellung der Menge D , welche vom Sensor als gesuchtes Objekt erfasst wird (c, f, i, l), basierend auf dem sensorbasierten ausschließbaren Raumbereich $A^{E \setminus U}$ (a, d, g, j) und A^E (b, e, h, k) im Kontinuierlichen (Pixel) und im Diskreten (Reale Pixel).

4. Umsetzung

von 0 liefern (Abbildung 4.10 k). Schaut man sich nun jedoch den Raumbereich an, welcher von der Farbkamera als gesuchtes Objekt erfasst wird (Abbildung 4.10 l), so muss man feststellen, dass ausgerechnet der Raumbereich in welchem sich das gesuchte Objekt befindet überhaupt nicht zu dem als gesuchtes Objekt erfassten Bereich gehört. Die einfache Lösung ist die, dass man beim Berechnen des Erfassbarkeitsparameters ausschließlich die realen Pixel und deren rückprojiziertes Volumen berücksichtigt, mit dem Nachteil, dass bei Änderung der Kameraauflösung der Erfassbarkeitsparameter neu bestimmt werden muss. Eine andere Lösung ist, dass man als Referenztiefe die maximal sichtbare Tiefe verwendet und diese immer mit dem Wert der tiefenbeschreibenden Real-Pixelfunktion vergleicht (Abbildung 4.11). Dies hat den Nachteil,

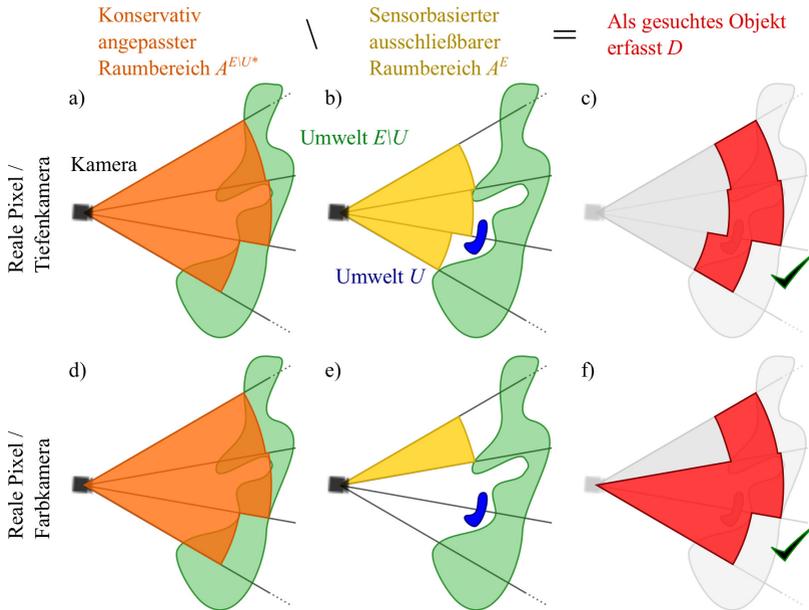


Abbildung 4.11.: Darstellung der konservativen Lösung zur Ermittlung der Menge D , welche vom Sensor als gesuchtes Objekt erfasst wird (c, f), basierend auf dem konservativ angepassten Raumbereich $A^{E \setminus U^*}$ (a, d) und dem sensorbasierten ausschließbaren Raumbereich A^E (b, e) im Diskreten (Reale Pixel).

das an Gegenständen Voxel *immer* als gesuchtes Objekt interpretiert werden und man somit auch tendenziell weniger Regionen einer voxelbasierten geometrischen Rekonstruktion verwerfen kann.

4.3.3.3. Auswahl der Alternativen

Man erkennt direkt, dass sich beide Diskretisierungs-Probleme zur Transformation in die voxelbasierte Erfassbarkeit mit realen Kameras ähneln. Weil erfahrungsgemäß die Voxelraumauflösung häufiger als ein Sensor, bzw. dessen Auflösung geändert wird, ist es sinnvoll sich im ersten Fall für die konservative Lösung (vgl. Abbildung 4.9 g, h, i) und im zweiten Fall für die einfache Lösung (vgl. Abbildung 4.10 d-f, j-l, mit geeignet gewähltem Erfassbarkeitsparameter) zu entscheiden. Damit ist die Angabe des Erfassbarkeitsparameters unabhängig von der Voxelraumauflösung, wohl aber von der Auflösung der Kamera. Diese Entscheidung wird im Folgenden verwendet.

4.3.3.4. Umsetzung am Beispiel realer Kameras

Basierend auf einer gegebenen Voxelmenge muss für jeden enthaltenen Voxel berechnet werden, ob dieser dem sensorbasierten ausschließbaren Raumbereich in einer Umwelt *ohne* gesuchte Objekte *angehört* und gleichzeitig dem sensorbasierten ausschließbaren Raumbereich in einer Umwelt *mit* gesuchten Objekten *nicht angehört*. Genau dann wird dieser Voxel vom Sensor als gesuchtes Objekt erfasst und der Zähler für eine Voxelregion kann um diesen Sensor erhöht werden. Wie im vorhergehenden Kapitel entschieden, ist der Erfassbarkeitsparameter abhängig von der Kameraauflösung, aber unabhängig von der Voxelraumauflösung angegeben. Aufgrund der Unabhängigkeit bezüglich der Voxelraumauflösung müssen gesuchte Objekte konservativ voxelbasiert erfasst werden (Definition 64), was die Berechnung, ob ein Voxel dem sensorbasierten ausschließbaren Raumbereich in einer Umwelt *ohne* gesuchte Objekte angehört, beeinflusst, da hier nicht mehr die verkleinerte Voxelapproximation verwendet werden darf. Stattdessen müssen hierfür auch Voxel verwendet werden die bereits zum Teil im kontinuierlichen ausschließbaren Raumbereich liegen. Das sind solche Voxel, deren Volumen zum Teil in mindestens einem ihrer Projektionspixel näher am Brennpunkt der Kamera liegen als der Wert den die tiefenbeschreibenden Real-Pixelfunktion des jeweiligen Projektionspixels liefert. Auch hier lassen sich Voxelapproximationen vergleichbar zu Kapitel 4.3.2.2 verwenden (Abbildung

4. Umsetzung

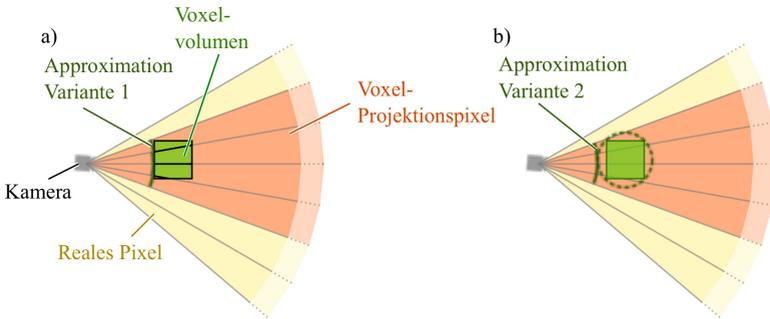


Abbildung 4.12.: Darstellung zweier möglicher Voxelapproximationen, wie sie im Rahmen der Erfassbarkeit mit realen Kameras verwendet werden können.

4.12). Der Unterschied liegt darin, dass nun der Voxel auch dann in der Auswertung berücksichtigt werden muss, wenn nur ein Teil seines Volumens im kontinuierlichen ausschließbaren Raumbereich liegt. Dies wird bewerkstelligt, in dem man nun die Approximation am nächstgelegenen Raumelement des Voxelvolumens für jeden Projektionspixel orientiert und die so ermittelte Distanz der oben eingeführten Datenstruktur der ersten Variante (Abbildung 4.12 a) hinzufügt und in der zweiten Variante (Abbildung 4.12 b) “online” berechnet.

Eine Voxelbewertung bezüglich der Erfassbarkeit – also ob dieser Voxel nun als gesuchtes Objekt erfasst wird – lässt sich mithilfe des Algorithmus 4.5 erreichen. Einerseits wird *konservativ* überprüft, ob der Voxel im ausschließbaren Raumbereich der Umwelt *ohne* gesuchter Objekte enthalten ist (Zeilen 2, 3, 5, 7-10) und andererseits, ob der Voxel *nicht* im voxelbasierten ausschließbaren Raumbereich der Umwelt *mit* gesuchten Objekten enthalten ist (Zeilen 2, 4, 6, 7, 11-13). Es werden für beide dieser Überprüfungen jeweils die auf der Umkugel basierende Voxelapproximation verwendet (Zeilen 3, 4; vgl. Abbildung 4.6 d und Abbildung 4.12 b).

In Algorithmus 4.6 wird diese Voxelbewertung zur Angabe der globalen Erfassbarkeitsfunktion genutzt. Zunächst werden Voxelregionen der gegebenen voxelbasierten geometrischen Rekonstruktion bestimmt (Zeile 3) und diese danach einzeln auf Plausibilität überprüft (Zeilen 4-29). Zur Überprüfung, ob eine Voxelregion plausibel ist, muss die Anzahl der Sensoren gezählt werden, welche diese als gesuchtes Objekt erfassen. Dazu wird eine Zählervariable in Zeile 6 bereitgestellt und im späteren Verlauf hochgezählt. Anfangs wird dann angenommen, dass die zu prüfende Voxelregion nicht-plausibel ist (Zeile 5). Für jeden einzelnen Sensor muss im Fol-

Algorithmus 4.5 Implementierung der Voxelbewertung für die Erfassbarkeit. Eingaben des Algorithmus sind die Voxelposition q , der Brennpunkt f , das Bild I und die zu dieser Kamera zugehörige Datenstruktur D . Ausgabe des Algorithmus ist *true*, falls der Voxel q als gesuchtes Objekt erfasst wird, *false* ansonsten.

```

1: procedure ERFASSBARKEITVOXELAUSWERTUNG( $q, f, I, D$ )
2:    $d \leftarrow \text{VOXELDIAGONALLENGTH}/2$            ▷ Ermittle halbe Länge der Voxel diagonalen
3:    $m_{<} \leftarrow \|\text{VOXELCENTER}(q) - f\|_2 - d$      ▷ Voxelapproximation für  $A^{E \setminus U}$ 
4:    $m_{>} \leftarrow \|\text{VOXELCENTER}(q) - f\|_2 + d$      ▷ Voxelapproximation für  $A^E$ 
5:    $b_{E \setminus U} \leftarrow \text{false}$                  ▷ Bitschalter: In  $A^{E \setminus U}$  enthalten?
6:    $b_E \leftarrow \text{EVALUATABLE}(D, q)$              ▷ Bitschalter: In  $A^E$  enthalten?
7:   for all  $p \in \text{PROJECTIONPIXELS}(D, q)$  do       ▷ Prüfe Voxel in allen Projektionspixeln
8:     if  $m_{<} \leq \text{VALUE}(I, p)$  then             ▷ Liegt Voxel doch in  $A^{E \setminus U}$ ?
9:        $b_{E \setminus U} \leftarrow \text{true}$            ▷ Voxel doch in  $A^{E \setminus U}$  enthalten!
10:    end if
11:    if  $m_{>} \geq \text{VALUE}(I, p)$  then             ▷ Liegt Voxel doch nicht in  $A^E$ ?
12:       $b_E \leftarrow \text{false}$                    ▷ Voxel doch nicht in  $A^E$  enthalten!
13:    end if
14:  end for
15:  return  $b_{E \setminus U} \wedge \text{not } b_E$            ▷ true, falls Voxel als gesuchtes Objekt erfasst; false, sonst
16: end procedure

```

genden überprüft werden, ob er die Voxelregion als gesuchtes Objekt erfasst (Zeilen 7-25). Hier handelt es sich bei den Sensoren um reale Kameras, zu denen als Informationen die Brennpunkte, die Tiefenbilder (gemäß tiefenbeschreibender Real-Pixelfunktion) und die Datenstrukturen (wie oben angegeben) benötigt werden. Um schließlich herauszufinden, ob eine Voxelregion als gesuchtes Objekt erfasst wird, muss jeder einzelne Voxel gemäß Algorithmus 4.5 ausgewertet werden (Zeile 13-20). Dabei kann es sein, dass ein Voxel auf dem Rand der Rekonstruktionszone liegt und die Voxelregion bereits aus diesem Grund plausibel ist (Zeile 9-12). Hat die Zählvariable schon den Wert des Erfassbarkeitsparameters erreicht, so gilt die Voxelregion ebenfalls als plausibel (Zeile 16-18). Ist eine Voxelregion plausibel, so können die Voxel dieser Voxelregion der auszugebenden voxelbasierten geometrischen Rekonstruktion hinzugefügt werden (Zeile 26-28). Sind alle Voxelregionen derart geprüft und deren Voxel gegebenenfalls der auszugebenden voxelbasierten geometrischen Rekonstruktion hinzugefügt worden, so kann diese als Resultat zurückgegeben werden (Zeile 30).

Die anderen Varianten der Erfassbarkeitsfunktion lassen sich nach ähnlichem Prinzip umsetzen, werden aber hier nicht weiter behandelt.

4. Umsetzung

Algorithmus 4.6 Implementierung der globalen Erfassbarkeitsfunktion. Eingaben sind eine voxelbasierte geometrische Rekonstruktion (VGR) R^V , die Rekonstruktionszone Z^V , Brennpunkte $f[]$, Tiefenbilder (gemäß tiefenbeschreibender Real-Pixelfunktionen) $I[]$, Datenstrukturen $D[]$, der Erfassbarkeitsparameter θ und der nachbarschaftsbeschreibende Graph G zur Regionenbildung.

```
1: procedure GLOBALEERFASSBARKEITSFUNKTION( $R^V, Z^V, f[], I[], D[], \theta, G$ )
2:    $R_2^V \leftarrow \text{EMPTYRECONSTRUCTION}()$   $\triangleright$  Initialisiere leere Ausgabe-VGR
3:    $Q \leftarrow \text{CLUSTER}(R^V, G)$   $\triangleright$  Ermittle Voxelregionen von  $R^V$ 
4:   for all  $Q \in \mathbb{Q}$  do  $\triangleright$  Prüfe jede Voxelregion
5:      $p \leftarrow \text{false}$   $\triangleright$  Annahme: Voxelregion nicht-plausibel
6:      $c \leftarrow 0$   $\triangleright$  Anzahl der Sensoren, welche  $Q$  als gesuchtes Objekt erfassen
7:     for all  $(f, I, D) \in (f[], I[], D[])$  do  $\triangleright$  Prüfe in jedem Kamerabild mit Datenstruktur
8:       for all  $q \in Q$  do  $\triangleright$  Prüfe jeden Voxel der Voxelregion  $Q$ 
9:         if  $q \in \text{Rand}(Z^V)$  then  $\triangleright$  Falls  $q$  Randvoxel
10:            $p \leftarrow \text{true}$   $\triangleright$  Voxelregion doch plausibel
11:           BREAK
12:         end if
13:         if  $\text{ERFASSBARKEITVOXELAUSWERTUNG}(q, f, I, D)$  then
14:            $\triangleright$  Falls Voxel als gesuchtes Objekt erfasst
15:              $c \leftarrow c + 1$   $\triangleright$  Erhöhe Anzahl der Sensoren, welche  $Q$  erfassen um 1
16:             if  $c \geq \theta$  then  $\triangleright$  Falls Mindestanzahl von Kameras erreicht
17:                $p \leftarrow \text{true}$   $\triangleright$  Voxelregion doch plausibel
18:             end if
19:             BREAK  $\triangleright$  Kamera wurde gezählt, nächste
20:           end if
21:         end for
22:         if  $p$  then  $\triangleright$  Falls Voxelregion bereits plausibel
23:           BREAK  $\triangleright$  Prüfe in keiner weiteren Kamera
24:         end if
25:       end for
26:       if  $p$  then  $\triangleright$  Falls Voxelregion bereits plausibel
27:          $\text{ADDVOXELS}(R_2^V, Q)$   $\triangleright$  Füge Voxelregionen-Voxel der VGR  $R_2^V$  hinzu
28:       end if
29:     end for
30:     return  $R_2^V$   $\triangleright$  Liefere Ausgabe-VGR zurück
31: end procedure
```

4.3.4. Form bezüglich Raumbereich

Hat man im Volumen der voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$ des Voxelraumes V einen Raumbereich $M^V \subseteq Z^V$ identifiziert, aus welchem Voxelregionen vergrößerter Voxelapproximationen $U^V \subset \mathbb{Z}^3$ der Menge $U \subset \mathbb{R}^n$, welche die gesuchten Objekte darstellt, auf V stets herausragen – insofern also $\varrho^V \setminus M^V \neq \emptyset$ mit $\varrho^V \in$ Menge aller Voxelregionen von U^V gilt – so können alle Voxelregionen einer voxelbasierten geometrischen Rekonstruktion verworfen werden, welche Teilmengen von M^V sind und gleichzeitig keinen Randvoxel von Z^V enthalten (vgl. Kapitel 3.2.4).

Definition 65. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subset \mathbb{R}^3$ die Menge, welche gesuchte Objekte repräsentiert, $U^V \subset \mathbb{Z}^3$ die vergrößerte Voxelapproximation der Menge U auf V und $M^V \subseteq Z^V$ ein Raumbereich für den das Prädikat $\Pi_f^V(\varrho^V, M^V) := \varrho^V \setminus M^V \neq \emptyset$ mit $\varrho^V \in$ Menge aller Voxelregionen von U^V gilt. Sei außerdem R^V eine voxelbasierte geometrische Rekonstruktion von U in Z^V , dann ist

$$p_f^V : 2^{Z^V} \times 2^{Z^3} \times 2^{Z^V} \rightarrow 2^{Z^V}, p_f^V(R^V, Z^V, M^V) := \bigcup_{\substack{\varrho^V \in \text{Menge aller Voxelregionen von } R^V \\ \varrho^V \cap \text{Rand}(Z^V) \neq \emptyset \vee \Pi_f^V(\varrho^V, M^V)}} \varrho^V$$

eine voxelbasierte Plausibilisierungsfunktion und heißt *voxelbasierte Formfunktion*.

Die Definition der voxelbasierten Formfunktion gleicht fast vollständig der Definition 34 im Kontinuierlichen, jedoch statt auf dem zugrundeliegenden Volumen, wird hier lediglich auf den Voxelpositionen gearbeitet – also ideal für eine Implementierung (Algorithmus 4.7).

Typischerweise liegt der Raumbereich, aus welchem Regionen gesuchter Objekte herausragen zunächst *nicht* in Voxelrepräsentation vor. Für den Fall, dass M ein Raumbereich im Kontinuierlichen ist, aus welchem Regionen gesuchter Objekte stets herausragen, so ist die verkleinerte Voxelapproximation als Eingabe für die voxelbasierte Formfunktion zu verwenden. Der Grund liegt darin, dass das Volumen einer Voxelregion, welches aus einer bestimmten Menge im Kontinuierlichen herausragt in jedem Fall auch aus einer Teilmenge, aber nicht aus einer Obermenge herausragen muss.

Im Rahmen einer Implementierung liegt jedoch ein solcher Raumbereich meist z.B. in Dreiecksnetz- oder Kugelmodellrepräsentation vor. Geeignete Transformationen, hin zu einer Voxelrepräsentation lassen sich im Anhangskapitel A nachlesen.

Algorithmus 4.7 Implementierung der Formfunktion. Neben der gegebenen voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$, der voxelbasierten geometrischen Rekonstruktion (VGR) $R^V \subseteq Z^V$ und dem voxelbasierten Raumbereich $M^V \subseteq Z^V$ aus welchem Voxelregionen vergrößerte Voxelapproximationen gesuchter Objekte herausragen wird zudem der Nachbarschaftsbeschreibende Graph G zur Regionenbildung benötigt. Ausgabe des Algorithmus ist die voxelbasierte geometrische Rekonstruktion $R_2^V \subseteq R^V$ gemäß Definition.

```

1: procedure FORMFUNKTION( $R^V, Z^V, G, M^V$ )
2:    $R_2^V \leftarrow$  EMPTYRECONSTRUCTION()           ▷ Initialisiere leere Ausgabe-VGR
3:    $Q \leftarrow$  CLUSTER( $R^V, G$ )                 ▷ Ermittle Voxelregionen von  $R^V$ 
4:   for all  $Q \in Q$  do                           ▷ Prüfe jede Voxelregion
5:      $p \leftarrow$  false                           ▷ Annahme: Voxelregion nicht-plausibel
6:     for all  $q \in Q$  do                           ▷ Prüfe jeden Voxel in Voxelregion
7:       if  $q \in \text{Rand}(Z^V) \vee q \notin M^V$  then   ▷ Falls  $q$  Randvoxel oder ausserhalb  $M^V$ 
8:          $p \leftarrow$  true                           ▷ Voxelregion doch plausibel
9:         BREAK
10:      end if
11:    end for
12:    if  $p$  then                                     ▷ Falls Voxelregion plausibel
13:      ADDVOXELS( $R_2^V, Q$ )                           ▷ Füge Voxelregionen-Voxel der VGR  $R_2^V$  hinzu
14:    end if
15:  end for
16:  return  $R_2^V$                                      ▷ Liefere Ausgabe-VGR zurück
17: end procedure

```

Wie bereits in Kapitel 3.2.4 beschrieben, ist die Hülle um bekannte Gegenstände eine einfache und sinnvolle Weise solche Raumbereiche anzugeben, aus denen Regionen gesuchter Objekte herausragen. Angedeutet wurde, dass die erweiterte Hülle, sofern der bekannte Gegenstand gleichzeitig einen ausschließbaren Raumbereich darstellt, im Rahmen der voxelbasierten Umsetzung wesentlich hilfreicher ist. An dieser Stelle wird direkt ersichtlich warum das so ist: Verwendet man zunächst die Ausschließbarkeitsfunktion und dann die Formfunktion über eine Hülle, so bleiben alle Voxelregionen übrig, bei denen mindestens einer der enthaltenen Voxel auf dem Rand des bekannten Gegenstandes liegt, während durch die Formfunktion über eine erweiterte Hülle genau diese verworfen werden können. Sowohl Ausschließbarkeitsfunktion als auch Formfunktion müssen verkleinerte Voxelapproximationen kontinuierlicher Mengen verwenden, sodass bei der einfachen Hüllenbetrachtung eine "Voxelschicht" zwischen Hülle und bekanntem Gegenstand entsteht, welche das Verwerfen von Voxelregionen verhindert (Abbildung 4.13).

4.3.5. Volumen

Kennt man den Mindestvolumeninhalt $v \in \mathbb{R}^+$ aller Regionen gesuchter Objekte $U \subset \mathbb{R}^3$ innerhalb des Volumens der voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$, so kann man diejenigen Voxelregionen einer voxelbasierten geometrischen Rekonstruktion verwerfen, welche einen geringeren Volumeninhalt als diesen Mindestvolumeninhalt haben. Da die vergrößerte Voxelapproximation der Menge gesuchter Objekte eine Obermenge der Menge gesuchter Objekte darstellt, hat sie folglich einen größeren, mindestens aber den gleichen Volumeninhalt. Daher gilt auch

$$\lambda(\varrho^V) \geq v, \varrho^V \in \text{Menge aller Voxelregionen der vergrößerten Voxelapproximation von } U.$$

Dies entspricht nahezu der Formulierung in Kapitel 3.2.5, mit dem Unterschied, dass man den Umweg über das Volumen von Voxelmengen geht.

Definition 66. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subset \mathbb{R}^3$ die Menge, welche gesuchte Objekte repräsentiert, $U^V \subset \mathbb{Z}^3$ die vergrößerte Voxelapproximation der Menge U auf V und $v \in \mathbb{R}^+$ das Mindestvolumen, sodass das Prädikat $\Pi_v^V(\varrho^V, v) := \lambda(\varrho^V) \geq v$ mit $\varrho^V \in \text{Menge aller Voxelregionen von } U^V$ gilt. Sei außerdem R^V

4. Umsetzung

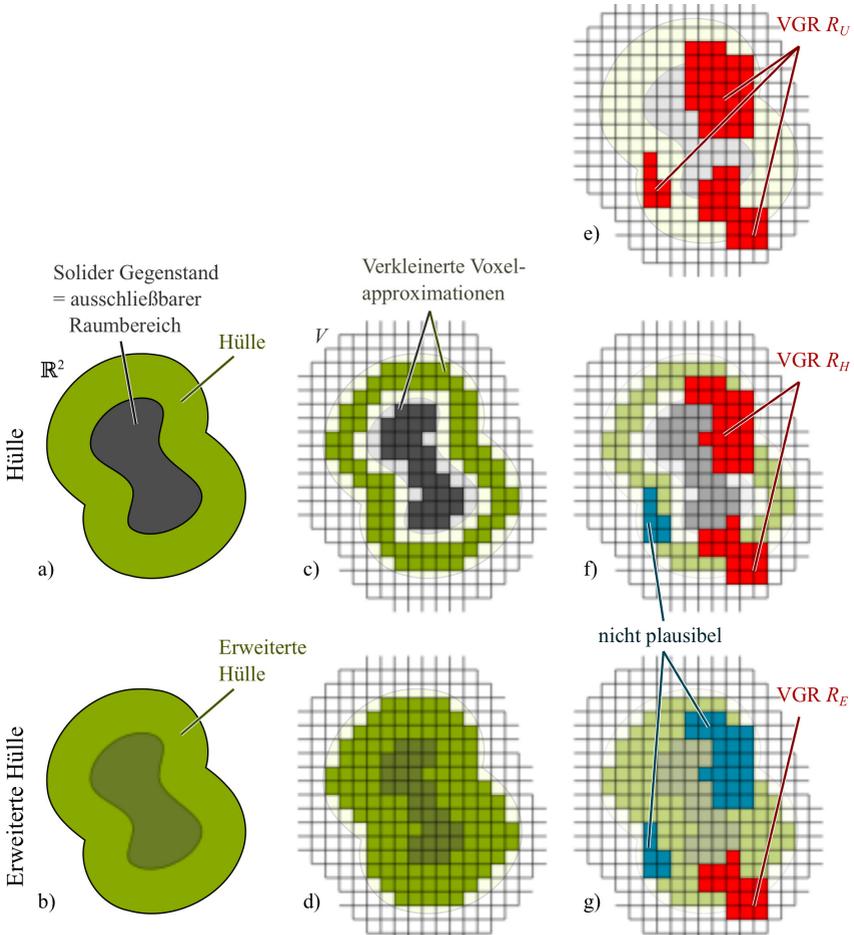


Abbildung 4.13.: Illustration der Unterschiede zwischen der einfachen (a, c) und der erweiterten (b, d) Hüllen in Verbindung mit einem ausschließbaren Raumbereich im Rahmen einer voxelbasierten Umsetzung, und Veranschaulichung der Auswirkungen an einem konkreten Beispiel (e, f, g). Sowohl ausschließbarer Raumbereich als auch Hülle sind in der Voxelraum-Betrachtung verkleinerte Voxelapproximationen (c, d). Zwischen einfacher Hülle und ausschließbarem Raumbereich entsteht eine von keiner der beiden Seiten berücksichtigte Voxelschicht. Im Beispiel ragt bei der einfachen Hülle eine Voxelregion *innen* heraus (f), was im Kontinuierlichen unmöglich ist. Im Fall der erweiterten Hülle bleibt lediglich die Voxelregion bestehen, welche *aussen* herausragt (g).

eine voxelbasierte geometrische Rekonstruktion von U in Z^V , dann ist

$$p_v^V : 2^{Z^V} \times 2^{\mathbb{Z}^3} \times \mathbb{R}^+ \rightarrow 2^{Z^V}, p_v^V(R^V, Z^V, v) := \bigcup_{\substack{\rho^V \in \text{Menge aller Voxelregionen von } R^V \\ \rho^V \cap \text{Rand}(Z^V) \neq \emptyset \vee \Pi_{\rho^V}(\rho^V, v)}} \rho^V$$

eine voxelbasierte Plausibilisierungsfunktion und heißt *voxelbasierte globale Volumenfunktion*.

Die Implementierung der voxelbasierten globalen Volumenfunktion erfolgt via Algorithmus

Algorithmus 4.8 Implementierung der voxelbasierten globalen Volumenfunktion. Neben der gegebenen voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$, der voxelbasierten geometrischen Rekonstruktion (VGR) $R^V \subseteq Z^V$ und dem Mindestvolumen v wird ebenfalls der Nachbarschaftsbeschreibende Graph G zur Regionenbildung benötigt. Ausgabe des Algorithmus ist die voxelbasierte geometrische Rekonstruktion $R_2^V \subseteq R^V$ gemäß Definition.

```

1: procedure GLOBALEVOLUMENFUNKTION( $R^V, Z^V, G, v$ )
2:    $R_2^V \leftarrow \text{EMPTYRECONSTRUCTION}()$  ▷ Initialisiere leere Ausgabe-VGR
3:    $Q \leftarrow \text{CLUSTER}(R^V, G)$  ▷ Ermittle Voxelregionen von  $R^V$ 
4:   for all  $Q \in Q$  do ▷ Prüfe jede Voxelregion
5:      $p \leftarrow \text{false}$  ▷ Annahme: Voxelregion nicht-plausibel
6:      $\lambda_\rho \leftarrow 0$  ▷ Initialisiere Regionenvolumeninhalte
7:     for all  $q \in Q$  do ▷ Prüfe jeden Voxel in Voxelregion
8:        $\lambda_\rho \leftarrow \lambda_\rho + \text{VOXELVOL}$  ▷ Addiere Voxelvolumen zu Regionenvolumeninhalte
9:       if  $q \in \text{Rand}(Z^V) \vee \lambda_\rho > v$  then ▷ Falls  $q$  Randvoxel oder Volumeninhalt  $> v$ 
10:         $p \leftarrow \text{true}$  ▷ Voxelregion doch plausibel
11:        BREAK
12:      end if
13:    end for
14:    if  $p$  then ▷ Falls Voxelregion plausibel
15:       $\text{ADDDOXELS}(R_2^V, Q)$  ▷ Füge Voxelregionen-Voxel der VGR  $R_2^V$  hinzu
16:    end if
17:  end for
18:  return  $R_2^V$  ▷ Liefere Ausgabe-VGR zurück
19: end procedure

```

4.8. Zunächst werden Voxelregionen der gegebenen voxelbasierten geometrischen Rekonstruktion bestimmt (Zeile 3), welche daraufhin im Einzelnen auf Plausibilität überprüft werden

4. Umsetzung

(Zeile 4-17). Diese Überprüfung funktioniert so, dass für jeden enthaltenen Voxel der Volumeninhalt eines solchen zum Gesamtvolumen dieser Voxelregion hinzugefügt wird (Zeile 8) und beim Überschreiten des Mindestvolumeninhalts die Voxelregion als plausibel gewertet (Zeile 10) und der auszugebenden voxelbasierten geometrischen Rekonstruktion hinzugefügt wird (Zeile 15). Wenn sich herausstellt, dass die betrachtete Voxelregion sich mit dem Rand der Rekonstruktionszone überschneidet, wird sie als plausibel gewertet (Zeile 9, 10).

Die voxelbasierte lokale Volumenfunktion ergibt sich wie folgt:

Definition 67. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subset \mathbb{R}^3$ die Menge, welche gesuchte Objekte repräsentiert und $U^V \subset \mathbb{Z}^3$ die vergrößerte Voxelapproximation der Menge U auf V . Sei außerdem $f^V : Z^V \rightarrow \mathbb{R}^+$ eine Funktion, die jedem Voxel in der voxelbasierten Rekonstruktionszone einen minimalen Volumeninhalt zuordnet und $F^V = \{(z, v) \in Z^V \times \mathbb{R}^+ \mid f^V(z) = v\}$ die linkstotale und rechtseindeutige Relation, die diese Funktion beschreibt, sodass das Prädikat $\Pi_V^V(\varrho^V, F) := \lambda(\varrho^V) \geq \inf\{v \mid (z, v) \in F^V \wedge z \in \varrho^V\}$ mit ϱ^V Menge aller Voxelregionen von U^V gilt. Desweiteren sei R^V eine voxelbasierte geometrische Rekonstruktion von U in Z^V , dann ist

$$p_V^V : 2^{Z^V} \times 2^{\mathbb{Z}^3} \times 2^{Z^V \times \mathbb{R}^+} \rightarrow 2^{Z^V}, p_V^V(R^V, Z^V, F^V) := \bigcup_{\substack{\varrho^V \in \text{Menge aller Voxelregionen von } R^V \\ \varrho^V \cap \text{Rand}(Z^V) \neq \emptyset \vee \Pi_V^V(\varrho^V, F^V)}} \varrho^V$$

die *voxelbasierte lokale Volumenfunktion*.

Gegenüber der kontinuierlichen Betrachtung unterscheidet sich die voxelbasierte lokale Volumenfunktion wieder nur durch den Umweg über das Volumen von Voxelmengen. Hat man eine Beschreibung der Relation im Kontinuierlichen, welche jedem Raumelement einen Mindestvolumeninhalt zuordnet, so erhält man die diskrete Version dieser Relation, indem man den zugeordneten *minimalen* Mindestvolumeninhalt aller Raumelemente innerhalb des zugehörigen Voxelvolumens für jenen Voxel verwendet, da gemäß der Festlegung des Prädikates, welches über Plausibilität entscheidet, Voxelregionen beibehalten werden, deren Volumeninhalt größer oder gleich dem *kleinsten* zugeordneten Mindestvolumeninhalt aller von der Voxelregion enthaltenen Raumelemente ist.

Die Implementierung dieser voxelbasierten lokalen Volumenfunktion lässt sich einfach mittels Algorithmus 4.9 umsetzen. Im Vergleich zur Implementierung der globalen Variante kommen

Algorithmus 4.9 Implementierung der voxelbasierten lokalen Volumenfunktion. Neben der gegebenen voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$, der voxelbasierten geometrischen Rekonstruktion (VGR) $R^V \subseteq Z^V$ und dem ortsabhängigen Mindestvolumen F^V wird ebenfalls der Nachbarschaftsbeschreibende Graph G zur Regionenbildung benötigt. Ausgabe des Algorithmus ist die voxelbasierte geometrische Rekonstruktion $R_2^V \subseteq R^V$ gemäß Definition.

```

1: procedure LOKALEVOLUMENFUNKTION( $R^V, Z^V, G, F^V$ )
2:    $R_2^V \leftarrow \text{EMPTYRECONSTRUCTION}()$            ▷ Initialisiere leere Ausgabe-VGR
3:    $Q \leftarrow \text{CLUSTER}(R^V, G)$                  ▷ Ermittle Voxelregionen von  $R^V$ 
4:   for all  $Q \in Q$  do                             ▷ Prüfe jede Voxelregion
5:      $p \leftarrow \text{false}$                            ▷ Annahme: Voxelregion nicht-plausibel
6:      $\lambda_\varrho \leftarrow 0$                        ▷ Initialisiere Regionenvolumeninhalt
7:      $v \leftarrow \infty$                            ▷ Initialisiere Mindestvolumeninhalt
8:     for all  $q \in Q$  do                             ▷ Ermittle Mindestvolumeninhalt anhand enthaltener Voxel
9:       if  $F^V(q) < v$  then                         ▷ Falls zugeordneter Mindestvolumeninhalt kleiner als  $v$ 
10:         $v \leftarrow F^V(q)$                        ▷ Weise  $v$  neuen Mindestvolumeninhalt zu
11:       end if
12:     end for
13:     for all  $q \in Q$  do                             ▷ Prüfe jeden Voxel in Voxelregion
14:        $\lambda_\varrho \leftarrow \lambda_\varrho + \text{VOXELVOL}$    ▷ Addiere Voxelvolumen zu Regionenvolumeninhalt
15:       if  $q \in \text{Rand}(Z^V) \vee \lambda_\varrho > v$  then   ▷ Falls  $q$  Randvoxel oder Volumeninhalt  $> v$ 
16:          $p \leftarrow \text{true}$                        ▷ Voxelregion plausibel
17:         BREAK
18:       end if
19:     end for
20:     if  $p$  then                                     ▷ Falls Voxelregion plausibel
21:        $\text{ADDVOXELS}(R_2^V, Q)$                        ▷ Füge Voxelregionen-Voxel der VGR  $R_2^V$  hinzu
22:     end if
23:   end for
24:   return  $R_2^V$                                      ▷ Liefere Ausgabe-VGR zurück
25: end procedure

```

4. Umsetzung

hier lediglich einige wenige Zeilen (7-12) hinzu, innerhalb derer der anzuwendende Mindestvolumeninhalt mittels einer Minimumbildung bestimmt wird. Dieser so ermittelte Mindestvolumeninhalt wird wie gehabt zur Prüfung der Voxelregion angewendet (Zeilen 13-19).

4.3.6. Distanz

Haben Regionen gesuchter Objekte $U \subset \mathbb{R}^3$ die Eigenschaft eine maximale Distanz $d \in \mathbb{R}_0^+$ zu einer nicht-leeren Menge $M \subset \mathbb{R}^n$ zu besitzen, so sind alle Regionen einer geometrischen Rekonstruktion $R \subseteq Z \subset \mathbb{R}^3$ von U in Z , welche sich nicht mit dem Rand von Z überschneiden und deren Abstand zu M größer ist als d nicht plausibel. In der Voxelbetrachtung ist bei gegebenem Voxelraum V und gegebener voxelbasierter Rekonstruktionszone Z^V die Distanz zwischen der Menge M zur vergrößerten Voxelapproximation von U – welche der voxelbasierten Optimalrekonstruktion entspricht – immer kleiner oder gleich der Distanz zwischen der Menge M und U . Daher ist es zulässig die Distanz d direkt für eine voxelbasierte Plausibilisierungsfunktion zu verwenden:

Definition 68. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subset \mathbb{R}^3$ die Menge, welche gesuchte Objekte repräsentiert, $U^V \subset \mathbb{Z}^3$ die vergrößerte Voxelapproximation der Menge U auf V , $M \subset \mathbb{R}^3$ eine nicht-leere Menge und $d \in \mathbb{R}_0^+$ eine Distanz, sodass das Prädikat $\Pi_d^V(\varrho^V, M, d) := \inf\{\|u - m\|_2 \mid u \in \text{Volumen der Voxelmenge } \varrho^V, m \in M\}$ mit $\varrho^V \in \text{Menge aller Voxelregionen von } U^V$ gilt. Desweiteren sei R^V eine voxelbasierte geometrische Rekonstruktion von U in Z^V , dann ist

$$p_d^V : 2^{Z^V} \times 2^{\mathbb{Z}^3} \times 2^{\mathbb{R}^3} \times \mathbb{R}_0^+ \rightarrow 2^{Z^V}, p_d^V(R^V, Z^V, M, d) := \bigcup_{\substack{\varrho^V \in \text{Menge aller Voxelregionen von } R^V \\ \varrho^V \cap \text{Rand}(Z^V) \neq \emptyset \vee \Pi_d^V(\varrho^V, M, d)}} \varrho^V$$

die *voxelbasierte globale Distanzfunktion*.

Eine Implementierung dieser Funktion ist in Algorithmus 4.10 aufgelistet. Da die voxelbasierte globale Distanzfunktion einzelne Voxelregionen auf Plausibilität prüfen muss, werden diese in Zeile 3 ermittelt und in der nachfolgenden Schleife (Zeilen 4-23) einzeln ausgewertet, was innerhalb der definierten Formel dem " $\varrho^V \in \text{Menge aller Voxelregionen von } R^V$ " entspricht. Die Plausibilität selbst wird im inneren Teil anhand der zur Voxelregion gehörenden einzelnen Voxel geprüft – zunächst ob es sich um einen Randvoxel handelt, danach ob die Voxelregion be-

Algorithmus 4.10 Implementierung der globalen Distanzfunktion. Neben der gegebenen voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$, der voxelbasierten geometrischen Rekonstruktion (VGR) $R^V \subseteq Z^V$, der Menge $M \subset \mathbb{R}^3$ und der Maximaldistanz d zu M wird ebenfalls der Nachbarschaftsbeschreibende Graph G zur Regionenbildung benötigt. Ausgabe des Algorithmus ist die voxelbasierte geometrische Rekonstruktion $R_2^V \subseteq R^V$ gemäß Definition.

```

1: procedure GLOBALEDISTANZFUNKTION( $R^V, Z^V, G, M, d$ )
2:    $R_2^V \leftarrow \text{EMPTYRECONSTRUCTION}()$            ▷ Initialisiere leere Ausgabe-VGR
3:    $Q \leftarrow \text{CLUSTER}(R^V, G)$                  ▷ Ermittle Voxelregionen von  $R^V$ 
4:   for all  $Q \in Q$  do                             ▷ Prüfe jede Voxelregion
5:      $p \leftarrow \text{false}$                            ▷ Annahme: Voxelregion nicht-plausibel
6:      $\delta \leftarrow \infty$                          ▷ Initialisiere Regionendistanz
7:     for all  $q \in Q$  do                             ▷ Prüfe jeden Voxel in Voxelregion
8:       if  $q \in \text{Rand}(Z^V)$  then                   ▷ Falls  $q$  Randvoxel
9:          $p \leftarrow \text{true}$                          ▷ Voxelregion doch plausibel
10:        BREAK
11:      end if
12:      if  $\text{DISTANCE}(q, M) < \delta$  then             ▷ Falls Distanz kleiner als  $\delta$ 
13:         $\delta \leftarrow \text{DISTANCE}(q, M)$          ▷ Weise neue Minimumdistanz  $\delta$  zu
14:        if  $\delta \leq d$  then                         ▷ Falls Regionendistanz kleiner als  $d$ 
15:           $p \leftarrow \text{true}$                          ▷ Voxelregion doch plausibel
16:          BREAK
17:        end if
18:      end if
19:    end for
20:    if  $p$  then                                     ▷ Falls Voxelregion plausibel
21:       $\text{ADDVOXELS}(R_2^V, Q)$                          ▷ Füge Voxelregionen-Voxel der VGR  $R_2^V$  hinzu
22:    end if
23:  end for
24:  return  $R_2^V$                                      ▷ Liefere Ausgabe-VGR zurück
25: end procedure

```

4. Umsetzung

reits die Maximaldistanz unterschritten hat. In beiden Fällen handelt es sich um eine plausible Voxelregion und die Prüfung weiterer Voxel wird abgebrochen. Wenn es sich um eine plausible Voxelregion handelt, werden alle Voxel dieser zur auszugebenden voxelbasierten geometrischen Rekonstruktion hinzugefügt (Zeile 21).

Die Funktion zur Bestimmung der Distanz zwischen Voxel q und Menge M (Zeilen 12 und 13) hängt von der Repräsentation der Menge M ab. Da eine kleiner bestimmte Distanz gegenüber der tatsächlichen Distanz eine konservative Abschätzung darstellt – es werden dadurch tendenziell weniger Voxelregionen verworfen – ist es zulässig, beispielsweise, wenn M durch ein Kugelmodell repräsentiert wird, als Distanz die Länge des Vektors von Voxelmittelpunkt v zu Kugelmittelpunkt c abzüglich des Kugelradius r und halber Länge der Voxeldiagonalen d zu verwenden, was in einer gleich großen oder kleineren Distanz gegenüber der tatsächlichen Distanz D resultiert: $(c - v) - r - \frac{d}{2} \leq D$.

Ist M hingegen durch die konvexe Hülle von Punkten festgelegt, so lässt sich zur Distanzbestimmung der GJK-Algorithmus [Gilb 88] zwischen den Punkten von M und dem Voxelmittelpunkt verwenden, wobei auch hier natürlich wieder die halbe Länge der Voxeldiagonalen abgezogen werden muss. Soll in diesem konkreten Beispiel die Distanz jedoch genau bestimmt werden, so verwendet man statt des Voxelmittelpunktes und dem anschließenden Abzug der halben Länge der Voxeldiagonalen einfach die acht Eckpunkte des Volumens dieses Voxels.

Wie man also sieht hängt die Implementierung der Funktion zur Bestimmung der Distanz zwischen einem Voxel und M direkt mit der Repräsentation von M zusammen und kann im jeweiligen Fall optimiert werden. Ändert sich M nicht, so kann eine Lookup-Tabelle für jeden Voxel “offline” aufgebaut werden, sodass die Funktion zur Ermittlung der Distanz einfach diese für den gerade betrachteten Voxel in der Tabelle nachschlägt. Dies stellt im Rahmen des Voxelansatzes eine “kleine” lokale Optimierung dar und lässt sich bei statischen Gegenständen nutzen.

Die Formel und Implementierung der voxelbasierten *lokalen* Distanzfunktion wird hier nicht explizit aufgeführt, da ihre Anpassung analog zur voxelbasierten lokalen Volumenfunktion verläuft. Ebenso wird die voxelbasierte *richtungsspezifische* globale und lokale Distanzfunktion nicht extra aufgeführt, da der einzige Unterschied in der *gerichteten* Distanzbestimmung liegt. Dies betrifft in der Implementierung der voxelbasierten *globalen* Distanzfunktion lediglich die Zeilen 12 und 13 insofern, als dort beispielsweise *directedDistance* mit einem zusätzlichen Eingabeparameter, nämlich der Richtung, verwendet wird. Dieser zusätzliche Parameter muss

natürlich auch in die Signatur der beiden richtungsspezifischen Distanzfunktionen aufgenommen werden.

4.3.7. Rand

Haben Regionen gesuchter Objekte $U \subset \mathbb{R}^n$ innerhalb einer Rekonstruktionszone $Z \subset \mathbb{R}^n$ die Eigenschaft sich mit dem Rand der Rekonstruktionszone $\text{Rand}(Z)$ zu überschneiden – gilt also $\varrho \cap Z \neq \emptyset \rightarrow \varrho \cap \text{Rand}(Z) \neq \emptyset$ mit $\varrho \in$ Menge aller Regionen von U , so sind alle Regionen einer geometrischen Rekonstruktion $R \subseteq Z$ von U in Z , welche sich nicht mit dem Rand von Z überschneiden nicht plausibel. Dies gilt auch dann, wenn man vergrößerte Voxelapproximationen für Rekonstruktionszone und gesuchte Objekte verwendet, denn die Raumelemente an denen sich eine Region gesuchter Objekte mit dem Rand der Rekonstruktionszone im Kontinuierlichen überschneidet werden sowohl in Randvoxeln der vergrößerten Voxelapproximation der Rekonstruktionszone als auch in den gleichen Voxeln der Voxelregion der vergrößerten Voxelapproximation von U liegen:

Definition 69. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U \subset \mathbb{R}^3$ die Menge, welche gesuchte Objekte repräsentiert und $U^V \subset \mathbb{Z}^3$ die vergrößerte Voxelapproximation der Menge U auf V . Desweiteren sei R^V eine voxelbasierte geometrische Rekonstruktion von U in Z^V und gelte das Prädikat $\Pi_r^V(\varrho^V, Z^V) := \varrho^V \cap Z^V \neq \emptyset \Rightarrow \varrho^V \cap \text{Rand}(Z^V) \neq \emptyset$ mit $\varrho^V \in$ Menge aller Voxelregionen von U^V , dann ist

$$p_r^V : 2^{Z^V} \times 2^{\mathbb{Z}^3} \rightarrow 2^{Z^V}, p_r^V(R^V, Z^V) := \bigcup_{\substack{\varrho^V \in \text{Menge aller Voxelregionen von } R^V \\ \Pi_r^V(\varrho^V, Z^V)}} \varrho^V$$

die *voxelbasierte Randfunktion*.

Die Implementierung (Algorithmus 4.11) wird der Vollständigkeit halber hier angegeben, beinhaltet jedoch nichts Neues gegenüber den Implementierungen vorheriger voxelbasierter Plausibilisierungsfunktionen. Da eine voxelbasierte geometrische Rekonstruktion ohnehin komplett innerhalb der voxelbasierten Rekonstruktionszone liegt, reicht es aus die Voxel der Voxelregionen auf Überschneidung mit Randvoxeln zu testen (Zeile 7).

Algorithmus 4.11 Implementierung der Randfunktion. Neben der gegebenen voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$, und der voxelbasierten geometrischen Rekonstruktion (VGR) $R^V \subseteq Z^V$ wird wie gehabt der Nachbarschaftsbeschreibende Graph G zur Regionenbildung benötigt. Ausgabe des Algorithmus ist die voxelbasierte geometrische Rekonstruktion $R_2^V \subseteq R^V$ gemäß Definition.

```

1: procedure RANDFUNKTION( $R^V, Z^V, G$ )
2:    $R_2^V \leftarrow$  EMPTYRECONSTRUCTION()           ▷ Initialisiere leere Ausgabe-VGR
3:    $\mathbb{Q} \leftarrow$  CLUSTER( $R^V, G$ )                 ▷ Ermittle Voxelregionen von  $R^V$ 
4:   for all  $Q \in \mathbb{Q}$  do                             ▷ Prüfe jede Voxelregion
5:      $p \leftarrow$  false                               ▷ Annahme: Voxelregion nicht-plausibel
6:     for all  $q \in Q$  do                             ▷ Prüfe jeden Voxel in Voxelregion
7:       if  $q \in \text{Rand}(Z^V)$  then                   ▷ Falls  $q$  Randvoxel
8:          $p \leftarrow$  true                             ▷ Voxelregion doch plausibel
9:         BREAK
10:      end if
11:    end for
12:    if  $p$  then                                       ▷ Falls Voxelregion plausibel
13:      ADDVOXELS( $R_2^V, Q$ )                               ▷ Füge Voxelregionen-Voxel der VGR  $R_2^V$  hinzu
14:    end if
15:  end for
16:  return  $R_2^V$                                        ▷ Liefere Ausgabe-VGR zurück
17: end procedure

```

4.3.8. Zeit

Haben Regionen gesuchter Objekte die Eigenschaft zu zwei aufeinander folgenden diskreten Zeitpunkten sich nicht weiter als eine Distanz $d \in \mathbb{R}_0^+$ voneinander zu entfernen, so können alle Regionen einer geometrischen Rekonstruktion R der gesuchten Objekte $U \subset \mathbb{R}^3$ verworfen werden, welche weiter als d von der vorhergehenden Rekonstruktion R_p der gesuchten Objekte $U_p \subset \mathbb{R}^3$ entfernt sind. Dies trifft auch dann zu, wenn man eine vergrößerte Voxelapproximation für die gesuchten Objekte verwendet, da dadurch die Distanz kleiner wird und daher tendenziell weniger Voxelregionen einer voxelbasierten geometrischen Rekonstruktion verworfen werden. Insofern ist eine solche Vorgehensweise konservativ und deswegen zulässig.

Definition 70. Seien $V = (b, t, r)$ ein Voxelraum, $Z^V \subset \mathbb{Z}^3$ eine voxelbasierte Rekonstruktionszone, $U, U_p \subset \mathbb{R}^3$ die Mengen, welche gesuchte Objekte zum Zeitpunkt t bzw. dem vorhergehenden Zeitpunkt t_p repräsentieren und $U^V, U_p^V \subset \mathbb{Z}^3$ die vergrößerte Voxelapproximation der Mengen U bzw. U_p auf V . Des Weiteren seien $R^V, R_p^V \subset Z^V$ voxelbasierte geometrische Rekonstruktionen von U bzw. U_p in Z^V sowie $d \in \mathbb{R}_0^+$ eine Distanz, sodass das Prädikat

$$\Pi_t^V(\varrho^V, \varrho_p^V, d, Z^V) := \inf\{\|r - r_p\|_2 \mid r, r_p \in \text{Volumen der Voxelmenge } \varrho^V \text{ bzw. } \varrho_p^V\},$$

mit $\varrho^V, \varrho_p^V \in$ Menge aller Voxelregionen von U^V bzw. U_p^V gelte. Dann ist

$$p_t^V : 2^{Z^V} \times 2^{Z^V} \times \mathbb{R}_0^+ \times 2^{\mathbb{Z}^3} \rightarrow 2^{Z^V}, p_t^V(R^V, R_p^V, d, Z^V) := \bigcup_{\substack{\varrho^V, \varrho_p^V \in \text{Menge aller Voxelreg. von } R^V \text{ bzw. } R_p^V \\ \inf\{\|r^V - r_p^V\|_2 \mid r^V, r_p^V \in \text{Vol. d. Vm. } \varrho^V \text{ bzw. Rand}(Z^V)\} \leq d \\ \vee \Pi_t^V(\varrho^V, \varrho_p^V, d, Z^V)}} \varrho^V$$

die *voxelbasierte Temporalfunktion*.

Zur Implementierung benötigt man, wie bereits in der Implementierung der voxelbasierten Distanzfunktion, die Berechnung einer Distanz – in diesem Fall jedoch zwischen zwei Voxelmengen. Auch hier ist, wie zu Beginn erwähnt, die Bestimmung einer kürzeren Distanz gegenüber der tatsächlichen, eine konservative und daher zulässige Vorgehensweise. In der Implementierung (Algorithmus 4.12) könnte also die Funktion zur Berechnung der Distanz Umkugeln der einzelnen Voxel in den Voxelmengen verwenden und die Distanz zwischen diesen Umkugeln berechnen.

4. Umsetzung

Algorithmus 4.12 Implementierung der Temporalfunktion. Neben der gegebenen voxelbasierten Rekonstruktionszone $Z^V \subset \mathbb{Z}^3$ und der beiden voxelbasierten geometrischen Rekonstruktionen (VGR) $R^V, R_p^V \subseteq Z^V$ wird die Distanz d und wie gehabt der Nachbarschaftsbeschreibende Graph G zur Regionenbildung benötigt. Ausgabe des Algorithmus ist die voxelbasierte geometrische Rekonstruktion $R_2^V \subseteq R^V$ gemäß Definition.

```

1: procedure TEMPORALFUNKTION( $R^V, R_p^V, d, Z^V, G$ )
2:    $R_2^V \leftarrow$  EMPTYRECONSTRUCTION()           ▷ Initialisiere leere Ausgabe-VGR
3:    $Q \leftarrow$  CLUSTER( $R^V, G$ )                 ▷ Ermittle Voxelregionen von  $R^V$ 
4:   for all  $Q \in \mathbb{Q}$  do
5:      $p \leftarrow$  false                         ▷ Annahme: Voxelregion nicht-plausibel
6:     for all  $q \in Q$  do                         ▷ Prüfe jede Voxelregion
7:       if DISTANCE( $q, \text{Rand}(Z^V)$ )  $\leq d$  then   ▷ Falls Distanz zu Rand  $\leq d$ 
8:          $p \leftarrow$  true                       ▷ Voxelregion doch plausibel
9:         BREAK
10:      end if
11:      if DISTANCE( $q, R_p^V$ )  $\leq d$  then           ▷ Falls Distanz  $\leq d$ 
12:         $p \leftarrow$  true                         ▷ Voxelregion doch plausibel
13:        BREAK
14:      end if
15:    end for
16:    if  $p$  then                                   ▷ Falls Voxelregion plausibel
17:      ADDVOXELS( $R_2^V, Q$ )                         ▷ Füge Voxelregionen-Voxel der VGR  $R_2^V$  hinzu
18:    end if
19:  end for
20:  return  $R_2^V$                                    ▷ Liefere Ausgabe-VGR zurück
21: end procedure

```

4.4. Experiment

In diesem Kapitel werden anhand einer beispielhaften Sequenz (Abbildung 4.14) der konkrete Einsatz der implementierten Plausibilisierungsfunktionen gezeigt und die errechneten geometrischen Rekonstruktionen bezüglich ihrer Güte quantitativ untersucht. Die Sequenz besteht aus 2452 betrachteten aufeinanderfolgenden Zeitpunkten, die jeweils 200 ms auseinanderliegen und hier der Einfachheit halber von 0 bis 2451 durchnummeriert sind. Außerdem wird das Resultat der in der Literatur zu findenden Methode der Occlusion Masks [Guan 06, Ladi 08a] gegenübergestellt.

4.4.1. Aufbau, Wissen und Sensoren

Die Umwelt besteht aus einem Roboter mit Steuerung, einem Tisch, einem Förderband, einem Regal, einem Boden, einer Decke und drei Wänden. Sie ist geometrisch bekannt und modelliert (Abbildung 4.15 a).

Neben den bekannten Gegenständen ist zusätzlich eine virtuelle Wand im Eingangsbereich modelliert, die den Zugang zum betrachteten Raumbereich einschränkt (Abbildung 4.15 a). Dies ist zulässig, da die gesuchten Objekte (hier Personen) tatsächlich nur in dem verbliebenen offenen Teil eintreten können, zumal der Rest einerseits durch – hier nicht weiter modellierte – Gegenstände teilweise versperrt ist und die Person andererseits auch nicht an der Decke hinein fliegen kann. Durch die virtuelle Wand wird die Güte der letztendlich resultierenden geometrischen Rekonstruktion verbessert, da hierdurch Regionen verworfen werden können, die die meisten Plausibilisierungsfunktionen vom Typ R nicht verwerfen dürften, wenn sich diese mit dem Rand der Rekonstruktionszone überschneiden würden. Die Anwendung der Ausschließbarkeitsfunktion über die modellierte Umwelt, also insbesondere auch der virtuellen Wand im Eingangsbereich, verhindert ein solches Überschneiden von Regionen mit dem Rand der Rekonstruktionszone.

Desweiteren werden als Sensoren sieben Farbkameras für Ausschließbarkeitsfunktionen und die allgemeine globale Erfassbarkeitsfunktion eingesetzt.

Die Regionen gesuchter Objekte

- können sich nicht innerhalb solider Gegenstände der Umwelt aufhalten, sodass eine (voxelbasierte) Ausschließbarkeitsfunktion über das Volumen der soliden Gegenstände eingesetzt werden kann (siehe Definition 58, Algorithmus 4.3);

4. Umsetzung

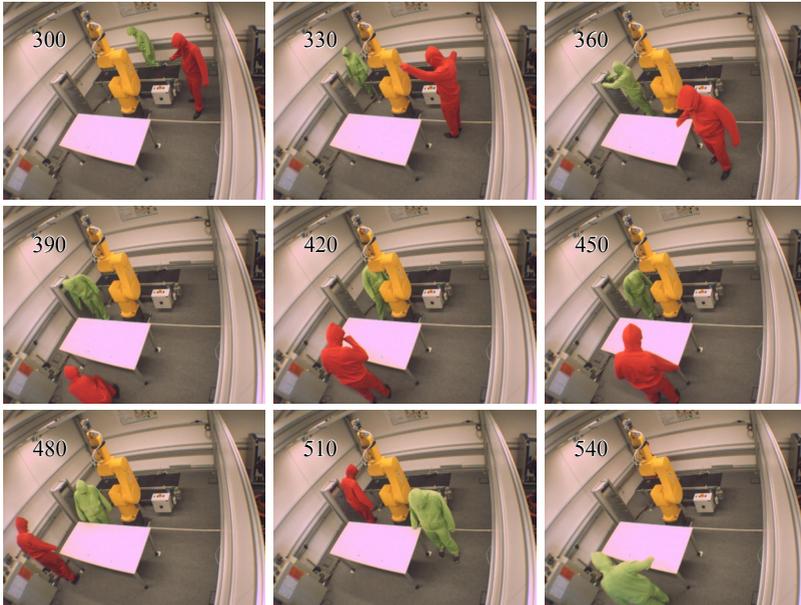


Abbildung 4.14.: Bilder mit eingetragener Zeitpunktnummer der aufgezeichneten und untersuchten Beispielsequenz aus Sicht der dritten Farbkamera. Regionen gesuchter Objekte bestehen aus den Farben rot und grün, haben einen Mindestvolumeninhalt von 65 Liter, ragen aus Hüllen mit einem Radius von 40 cm um bekannte Gegenstände heraus, befinden sich nicht weiter als 35 cm vom Fußboden entfernt, werden in mindestens drei Sensoren als gesuchte Objekte erfasst, können sich nicht in soliden Gegenständen aufhalten und überschneiden sich mit ihren Regionen im dreidimensionalen Raum zum unmittelbar vorhergehenden betrachteten Zeitpunkt.

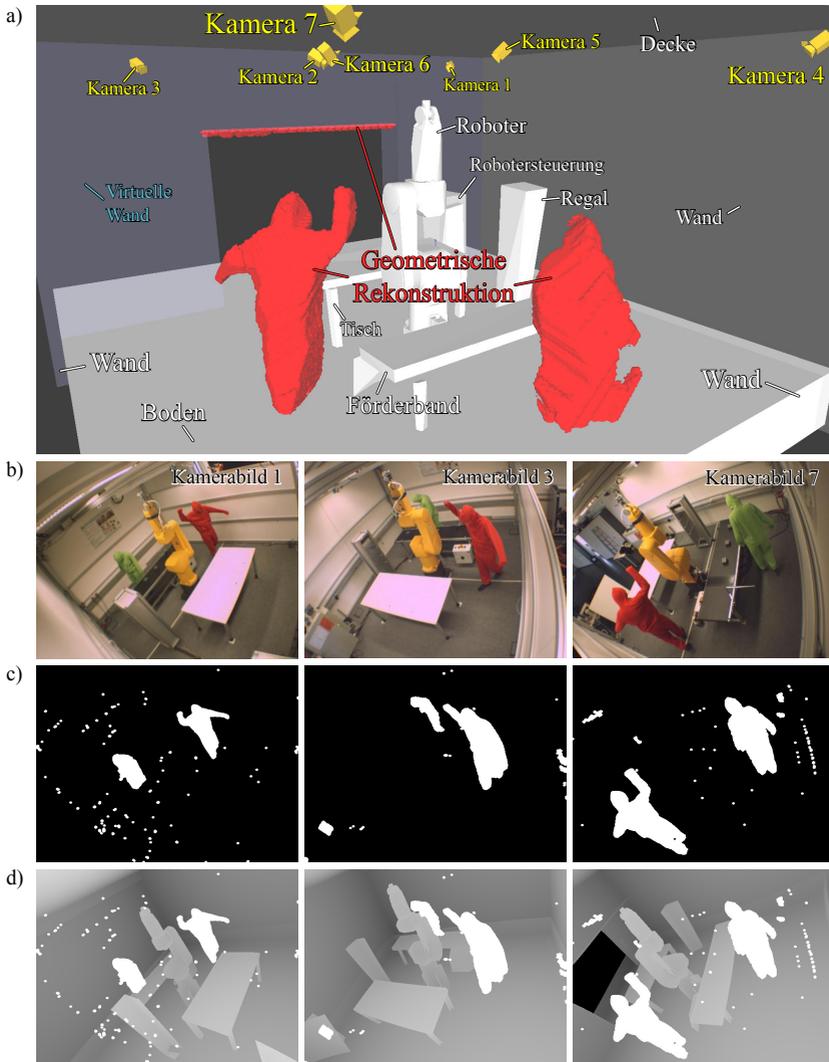


Abbildung 4.15.: a) Darstellung einer berechneten geometrischen Rekonstruktion (rot) innerhalb einer modellierten Umwelt zum Zeitpunkt mit der Nummer 307. b) Zur Plausibilisierung wurden unter anderem sieben Farbkameras verwendet. c) Es wurde eine farbbaasierte Segmentierung durchgeführt. d) Mit der Umweltmodellierung und der Segmentierung kann eine erweiterte tiefenbeschreibende Pixelfunktion bereitgestellt werden. Die Distanz ist hier über die Helligkeitswerte kodiert. Je heller ein Bildpunkt ist, desto geringer ist die Distanz. 141

4. Umsetzung

- bestehen aus den Farben *rot* und *grün*, sodass (voxelbasierte) Ausschließbarkeitsfunktionen über Farbkameras mit einer farbbasierten Segmentierung genutzt werden können. Abbildung 4.15 c zeigt die farbbasiert segmentierten Bilder von drei der sieben Farbkameras und die erweiterten tiefenbeschreibenden (Real-)Pixelfunktionen (Abbildung 4.15 d) – siehe auch Definition 19, Definition 62 und Algorithmus 4.4;
- haben einen Mindestvolumeninhalt von 65 l, sodass eine entsprechend parametrisierte (voxelbasierte) globale Volumenfunktion eingesetzt werden kann (siehe Definition 66, Algorithmus 4.8);
- haben eine geringere Distanz als 35 cm zum Fußboden, sodass die (voxelbasierte) richtungsspezifische globale Distanzfunktion entsprechend parametrisiert eingesetzt werden kann (siehe Definition 68, Algorithmus 4.10);
- werden von mindestens drei Sensoren als solche erfasst, sodass eine entsprechend parametrisierte (voxelbasierte) allgemeine globale Erfassbarkeitsfunktion verwendet werden kann (siehe Algorithmus 4.6);
- ragen aus Hüllen mit einem Radius von 40 cm um die modellierte Umwelt heraus, sodass die (voxelbasierte) Formfunktion über Hüllen entsprechend parametrisiert eingesetzt werden kann (siehe Definition 65, Algorithmus 4.7);
- überschneiden sich mit ihrer Region zum vorhergehenden betrachteten Zeitpunkt, der 200 ms zurückliegt, sodass eine entsprechend parametrisierte (voxelbasierte) Temporalfunktion verwendet werden kann (siehe Definition 70, Algorithmus 4.12).

Die Rekonstruktionszone umfasst den gesamten modellierten Raum und verläuft dort durch (virtuelle) Wände und Decken. Lediglich im verbliebenen, eingeschränkten Eingangsbereich können sich also geometrische Rekonstruktionen nach Anwendung der Ausschließbarkeitsfunktion über das Volumen solider Gegenstände mit dem Rand der Rekonstruktionszone noch überschneiden.

Zur Berechnung der geometrischen Rekonstruktionen wird eine Voxelkantenlänge von 15 mm und eine Auflösung von 560×480 Pixel in allen sieben Farbkameras gewählt.

4.4.2. Auswertung

In Abbildung 4.15 a und Abbildung 4.16 a, b sind resultierende geometrische Rekonstruktionen

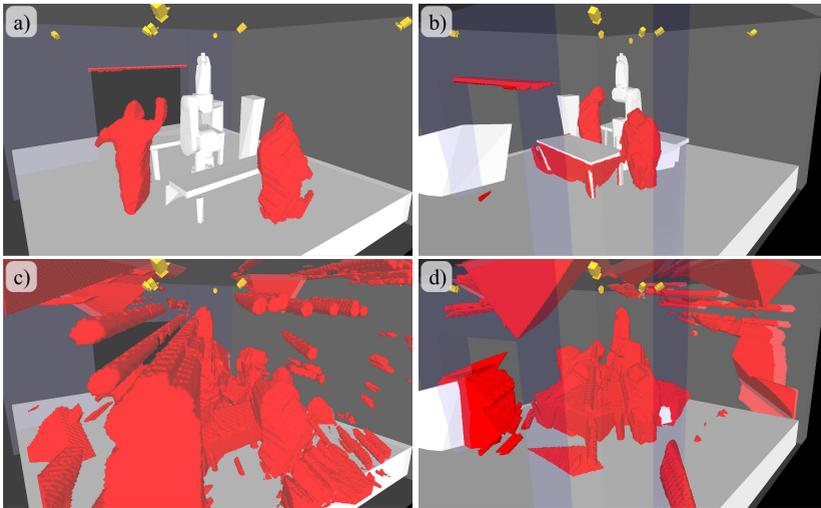


Abbildung 4.16.: a, b) Darstellung der berechneten geometrischen Rekonstruktionen innerhalb der modellierten Umwelt zu den Zeitpunkten mit den Nummern 307 und 509. c, d) Darstellung der Rekonstruktion gemäß der Methode mittels Occlusion Masks zu den selben Zeitpunkten.

innerhalb der Rekonstruktionszone in der modellierten Umwelt nach Anwendung aller zur Verfügung stehenden Plausibilisierungsfunktionen zu sehen. Zum Zeitpunkt mit der Nummer 307 (Abbildung 4.15 a, Abbildung 4.16 a) erhält man eine geometrische Rekonstruktion mit besonders hoher Güte: Abgesehen von einigen Voxeln im Eingangsbereich sind lediglich die gesuchten Objekte – also die beiden Personen – in der geometrischen Rekonstruktion enthalten. Etwas anders sieht es zum Zeitpunkt mit der Nummer 509 (Abbildung 4.16 b) aus; hier sind zusätzliche Voxel unterhalb des Tisches in der geometrischen Rekonstruktion enthalten, was sich wie folgt erklären lässt: Bei der Standardrekonstruktion geht man zunächst davon aus, dass gesuchte Objekte sich überall innerhalb der Rekonstruktionszone aufhalten können. Danach werden Plau-

4. Umsetzung

sibilisierungsfunktionen vom Typ E angewendet – hier also die Ausschließbarkeitsfunktionen über die soliden Gegenstände und der sieben Farbkameras. Anschließend werden nur noch die Plausibilisierungsfunktionen vom Typ R angewendet, also solche, die auf ganzen Regionen arbeiten. Zum Zeitpunkt mit der Nummer 509 ist es so, dass weder die Ausschließbarkeitsfunktion über die soliden Gegenstände, noch die über die sieben Farbkameras zwei separate Regionen für den von keinem Sensor eingesehenen Bereich unterhalb des Tisches und den daneben stehenden Personen erzeugen, stattdessen aber eine gesamte Region. Und diese gesamte Region muss selbstverständlich plausibel sein, da hier tatsächlich ein gesuchtes Objekt enthalten ist. Entstehen jedoch zwei getrennte Regionen, beispielsweise wenn sich die Person ein Stück vom Tisch entfernt, so besteht natürlich die Chance, dass eine der Plausibilisierungsfunktionen vom Typ R die Region unterhalb des Tisches verwerfen kann. Die hier aufgetretene Situation könnte entschärft werden, indem zusätzlich Sensoren angebracht würden, mit deren Hilfe der große verdeckte Bereich unter dem Tisch durch Anwendung von weiteren Ausschließbarkeitsfunktionen verkleinert oder gar komplett verworfen werden könnte. Alternativ könnte der Tisch auch durch einen soliden Quader ersetzt werden, sodass an der Stelle gar keine komplett verdeckten Bereiche entstehen könnten. Die Voxel würden in diesem Fall also mittels der Ausschließbarkeitsfunktion über dem Volumen solider Gegenstände verworfen werden.

Im Folgenden wird betrachtet, wieviele Voxel nach der Anwendung der Ausschließbarkeitsfunktionen über die sieben Kameras zu jedem Zeitpunkt der Sequenz übrig bleiben (Abbildung 4.17). Oben im Diagramm sieht man die Anzahl der Voxel der Standardrekonstruktion (rot), die konstant über die gesamte Sequenz bleibt, was daran liegt, dass zu jedem Zeitpunkt der Sequenz die gleiche Rekonstruktionszone verwendet wird. Etwas unterhalb, im Bereich von 15 bis 17 Millionen Voxel, befinden sich drei Kurven, welche beschreiben, wieviele Voxel nach der Anwendung der Ausschließbarkeitsfunktionen auf die Standardrekonstruktion separat über die Kameras 2, 5 und 6 verbleiben. Im Bereich von 10 bis 13 Millionen Voxel befinden sich die vier Kurven, welche separat von den Kameras 1, 3, 4 und 7 resultieren. Betrachtet man die Kameraplatzierung in Abbildung 4.15 a, so erkennt man, dass diejenigen vier Kameras, bei denen eine geringere Anzahl von Voxel übrig bleibt, am äußeren Rand des betrachteten Raumbereichs angebracht sind und diejenigen drei Kameras, bei denen eine höhere Anzahl von Voxel übrig bleibt, in der Mitte des betrachteten Raumbereichs angebracht sind. Die vier äußeren Kameras “sehen” folglich mehr Voxel und können demnach auch mehr Voxel verwerfen. Die unterste Kurve beschreibt das Resultat nach der Anwendung aller Ausschließbarkeitsfunktionen

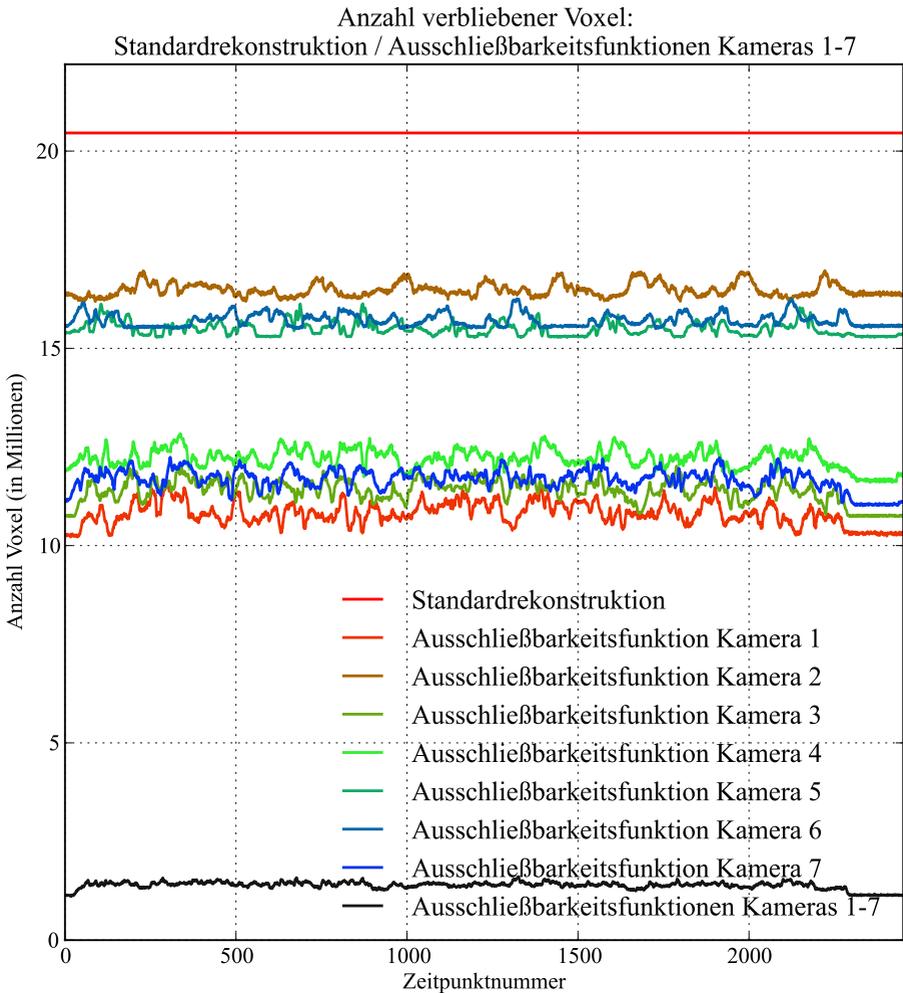


Abbildung 4.17.: Darstellung der Anzahl verbliebener Voxel nach Anwendung der Ausschließbarkeitsfunktionen auf die Standardrekonstruktion separat über die sieben Kameras sowie zusammengefasst.

4. Umsetzung

auf die Standardrekonstruktion über die Kameras und stellt damit eine Zusammenfassung dar. Es verbleiben demnach durchschnittlich etwa 6,79% der Voxel gegenüber der Standardrekonstruktion. Die Schwankungen im Verlauf der Kurven begründen sich durch die unterschiedlichen Aufenthaltsorte der gesuchten Objekte im Verlauf der Sequenz.

Neben den Ausschließbarkeitsfunktionen über die Kameras wird die Ausschließbarkeitsfunktion über das Volumen der soliden Gegenstände der modellierten Umwelt verwendet. In Abbildung 4.18 ist zunächst wieder die Anzahl an Voxeln der Standardrekonstruktion (rot) zu finden. Grün dargestellt ist die Kurve der verbleibenden Voxel nach Anwendung der Ausschließbarkeitsfunktion auf die Standardrekonstruktion über das Volumen der soliden Gegenstände. Die Kurve ist ebenso wie die der Standardrekonstruktion konstant, da keine bekannten soliden Gegenstände im Verlauf der Sequenz die Rekonstruktionszone betreten oder verlassen. In blau ist wieder die Zusammenfassung der Kameras 1 bis 7 dargestellt. Fasst man nun alle Ausschließbarkeitsfunktionen zusammen – also die über Kameras und die über das Volumen der soliden Gegenstände – so resultiert die dunkelgraue Kurve ganz unten im Diagramm. Man kann feststellen, dass der Abstand zwischen der blauen und dunkelgrauen Kurve geringer ist, als der zwischen roter und grüner Kurve. Dies ist begründet durch die Verwendung der erweiterten tiefenbeschreibenden (Real-)Pixelfunktionen (Definition 62): Die Ausschließbarkeitsfunktionen über die Kameras dürfen auch gewisse Voxel verwerfen, die innerhalb solider Gegenstände liegen, wodurch die Überlappung ausschließbarer Raumbereiche erreicht wird und damit dem Problem der Nicht-Verwerfbarkeit von "halben" Voxeln entgegen getreten wird. Insgesamt verbleiben nach Anwendung aller Ausschließbarkeitsfunktionen auf die Standardrekonstruktion durchschnittlich 4,12% der Voxel gegenüber der Standardrekonstruktion.

An diesem Punkt wurden alle zur Verfügung stehenden Plausibilisierungsfunktionen vom Typ E eingesetzt. Zum besseren Überblick sind die Mittelwerte der diskutierten Kurven nochmals in einem Diagramm zusammengefasst (Abbildung 4.19).

Als nächstes gilt es auf deren Resultat die Plausibilisierungsfunktionen vom Typ R anzuwenden. In Abbildung 4.20 wird zunächst als Referenzkurve das Resultat der Anwendung aller Plausibilisierungsfunktionen vom Typ E aus dem vorhergehenden Diagramm (Abbildung 4.18) übernommen. Auf dieses Resultat werden zunächst einzeln betrachtet die globale Volumenfunktion (Definition 37, Definition 66, Algorithmus 4.8), die richtungsspezifische globale Distanzfunktion (Definition 41, Definition 68, Algorithmus 4.10), die Formfunktion über Hüllen (Definition 34, Definition 35, Definition 36, Definition 65, Algorithmus 4.7) und die allgemeine

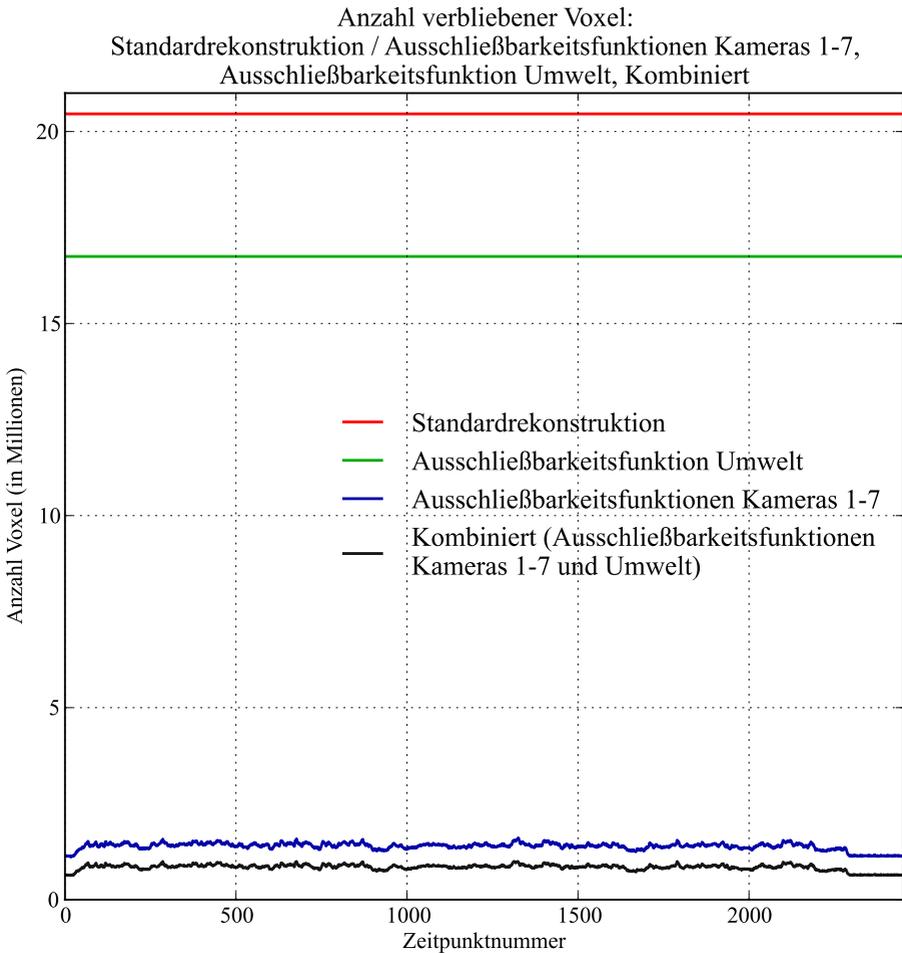


Abbildung 4.18.: Darstellung der Anzahl verbliebener Voxel nach Anwendung der Ausschließbarkeitsfunktion auf die Standardrekonstruktion über das Volumen der soliden Gegenstände der modellierten Umwelt sowie über alle Kameras und schließlich der Zusammenfassung dieser beiden.

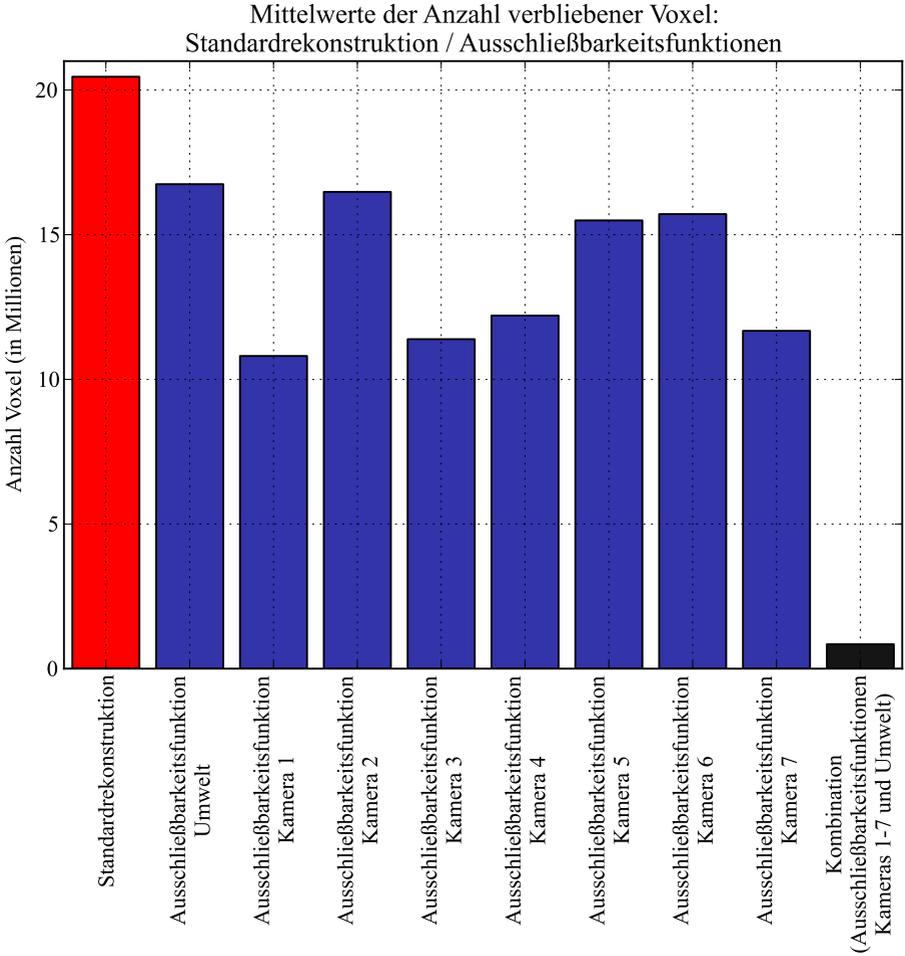


Abbildung 4.19.: Vergleich der Anzahl an Voxeln der Standardrekonstruktion (rot) mit der resultierenden Anzahl an Voxeln nach separater Anwendung sowohl der Ausschließbarkeitsfunktionen über die Kameras 1-7 als auch über das Volumen der soliden Gegenstände im Mittel über die Zeitpunkte (blau) und der resultierenden Anzahl an Voxeln nach Anwendung aller Ausschließbarkeitsfunktionen über die Kameras 1-7 zuzüglich der über das Volumen der soliden Gegenstände im Mittel über die Zeitpunkte (dunkelgrau).

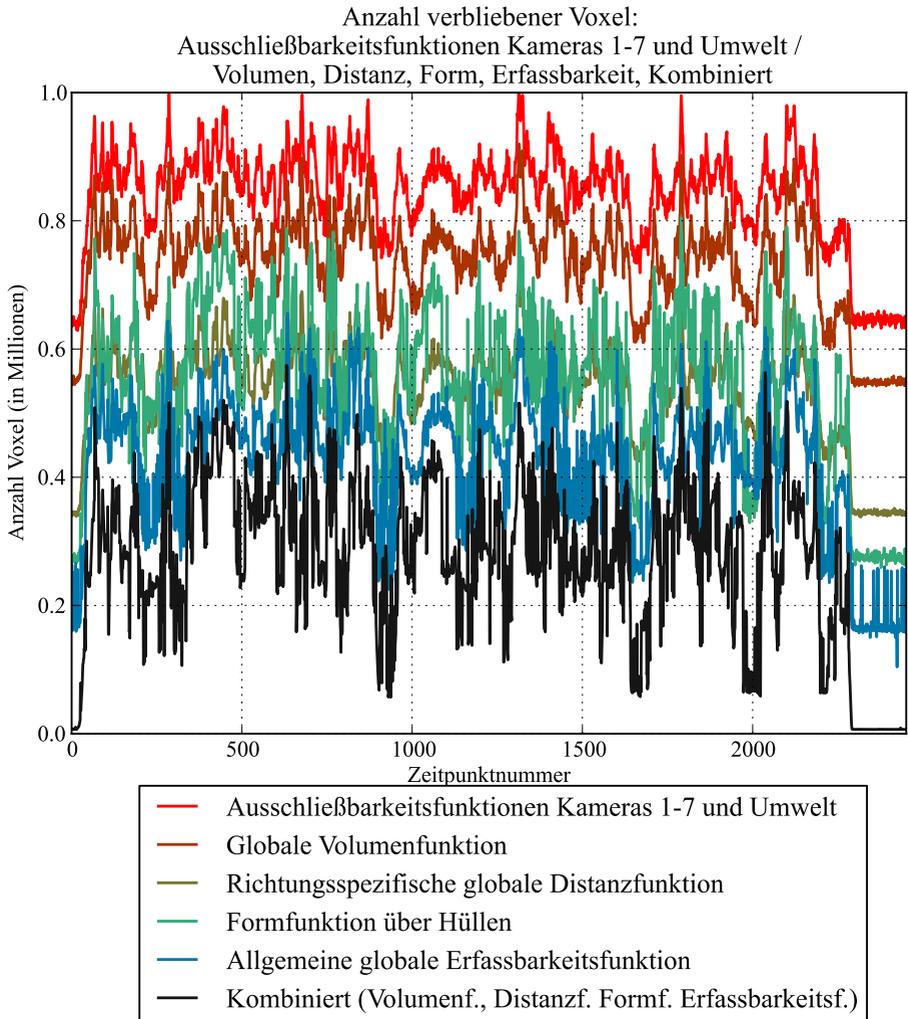


Abbildung 4.20.: Darstellung der Anzahl verbliebener Voxel nach Anwendung unterschiedlicher Plausibilisierungsfunktionen vom Typ R auf die geometrische Rekonstruktion nach Anwendung aller verfügbaren Plausibilisierungsfunktionen vom Typ E (rot).

4. Umsetzung

globale Erfassbarkeitsfunktion (Definition 30, Algorithmus 4.6) wie oben beschrieben angewendet. Im Diagramm lassen sich die daraufhin jeweils verbliebenen Voxel über alle Zeitpunkte der Sequenz ablesen. Die Zusammenfassung dieser vier Plausibilisierungsfunktionen vom Typ R resultiert in der dunkelgrauen Kurve, wobei erkennbar ist, dass die Zusammenfassung zu einem besseren Ergebnis gegenüber allen einzeln betrachteten Plausibilisierungsfunktionen vom Typ R führt, insofern, als meist eine geringere Anzahl von Voxeln übrig bleibt.

Die Temporalfunktion ist ebenfalls vom Typ R, benötigt als Eingabe allerdings die vorhergehende geometrische Rekonstruktion. Da die vorhergehende geometrische Rekonstruktion sinnvollerweise sämtliche Plausibilisierungen enthalten sollte, wird hier auch nur der Unterschied zwischen der Anwendung aller anderen Plausibilisierungsfunktionen vom Typ R gemäß des vorhergehenden Diagramms (Abbildung 4.20) und der Anwendung aller Plausibilisierungsfunktionen vom Typ R inklusive der Temporalfunktion betrachtet (Abbildung 4.21). Man erkennt, dass die Temporalfunktion in verschiedenen Situationen das Ergebnis weiter deutlich verbessert. Solche Situationen zeichnen sich dadurch aus, dass eine Region durch keine der anderen Plausibilisierungsfunktionen vom Typ R verworfen werden kann, wohl aber durch die Temporalfunktion, da in diese betrachtete Region zwischenzeitlich kein gesuchtes Objekt hineinschlüpfen konnte. Es verbleiben schließlich durchschnittlich über alle Zeitpunkte insgesamt nur noch 1,22% der Voxel gegenüber der Standardrekonstruktion.

Damit wurden nun alle verfügbaren Plausibilisierungsfunktionen angewendet. Die resultierende Kurve (grün) aus Abbildung 4.21 beschreibt daher die Anzahl an Voxeln der bestmöglichen geometrischen Rekonstruktion eines jeden einzelnen Zeitpunktes unter Berücksichtigung sämtlicher gegebener Information. Die Optimalrekonstruktion besteht aus den gesuchten Objekten innerhalb der Rekonstruktionszone. Da hier bekannt ist, dass es sich bei den gesuchten Objekten um zwei Personen handelt und deren Volumeninhalt zusammen geschätzte 180l umfasst und außerdem ein Voxel einen Volumeninhalt von $(15\text{ mm})^3$ besitzt, muss die Optimalrekonstruktion aus etwa 53334 Voxeln bestehen. Um die bestmögliche geometrische Rekonstruktion im Verlauf der Sequenz besser einschätzen zu können, ist dieser Wert als konstante Kurve im Diagramm in Abbildung 4.22 eingetragen (dunkelgrau). Die bestmögliche geometrische Rekonstruktion kann diese konstante Kurve in der aufgezeichneten Sequenz dann unterschreiten, wenn sich zu einem Zeitpunkt nicht beide Personen in der Rekonstruktionszone aufhalten – dies ist am Anfang und am Ende der Sequenz zu erkennen. Genaugenommen beschreibt in diesem Fall die konstante Kurve auch nicht mehr die Anzahl der Voxel der Optimalrekonstruktion.

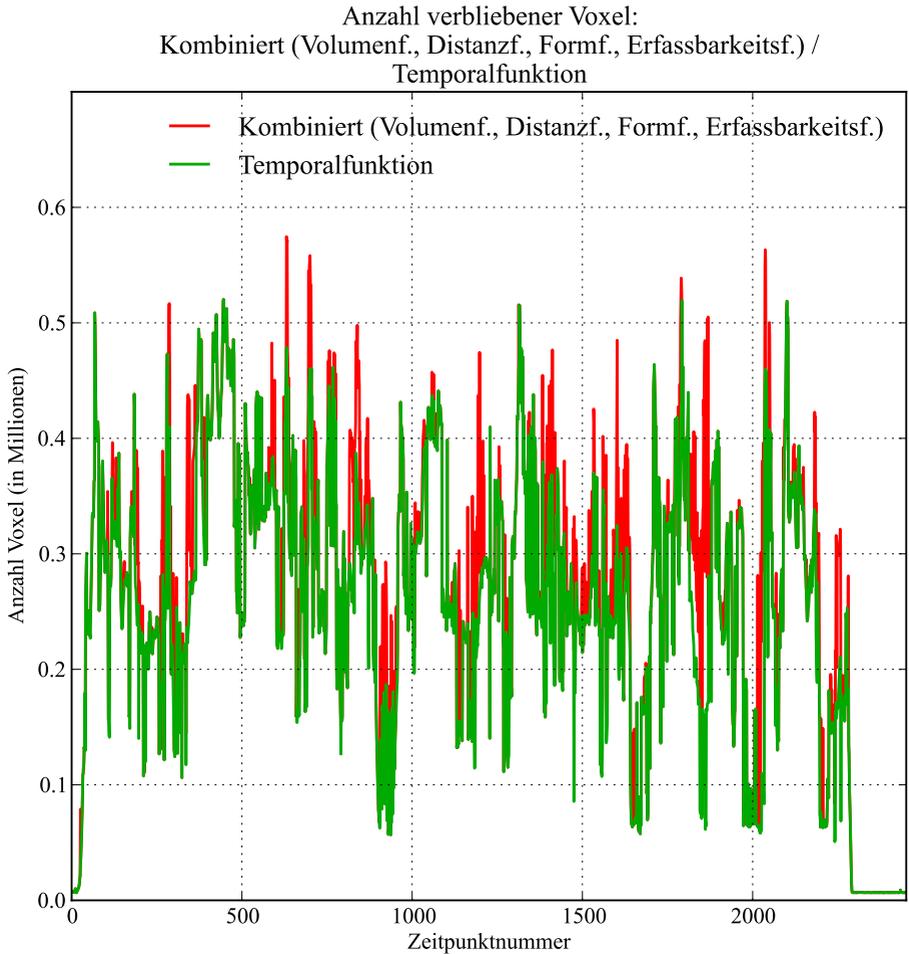


Abbildung 4.21.: Darstellung der Anzahl verbliebener Voxel nach Anwendung der Temporalfunktion (grün) auf die geometrische Rekonstruktion nach Anwendung aller anderen verfügbaren Plausibilisierungsfunktionen (rot).

Vergleich: Occlusion-Masks-basierte Rekonstruktion, Wissens- und sensorbasierte geometrische Rekonstruktion, Optimalrekonstruktion

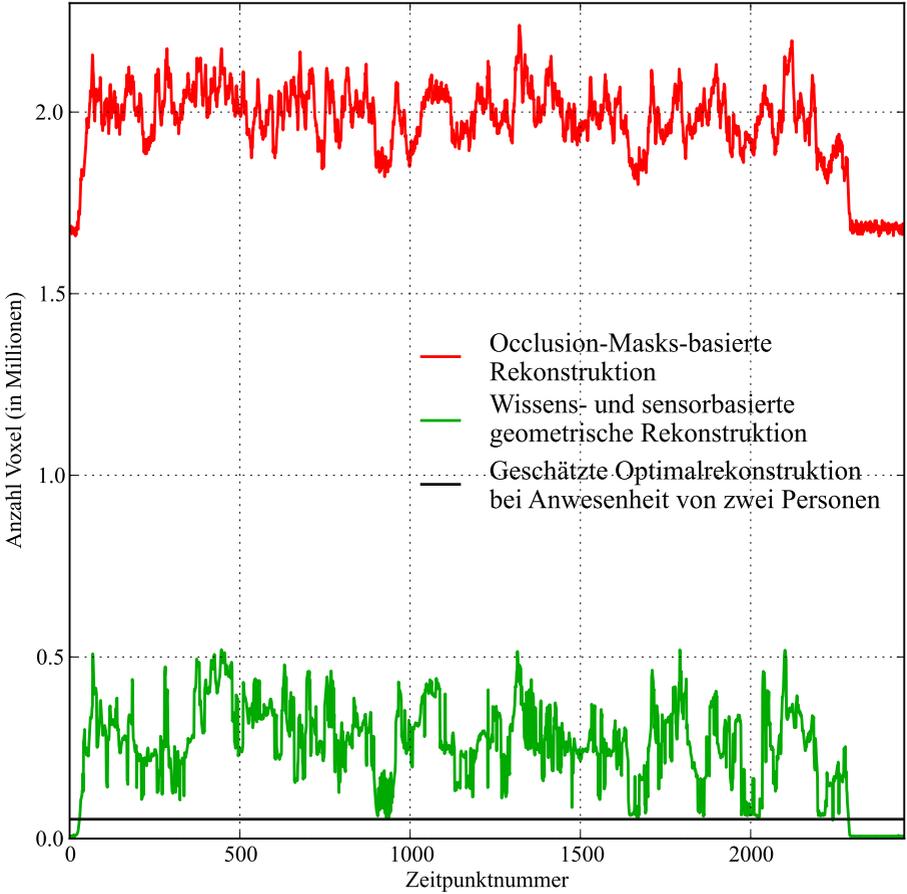


Abbildung 4.22.: Vergleich der Anzahl verbliebener Voxel über die gesamte Sequenz unter Verwendung der Occlusion Masks (rot), der wissens- und sensorbasierten geometrischen Rekonstruktion (grün) sowie der geschätzten Optimalrekonstruktion (dunkelgrau) für zwei Personen à 90l geteilt durch das Voxelvolumen von $(15\text{ mm})^3 = 3375\text{ mm}^3$, demnach also $53333, \bar{3}$ Voxel. Zu bemerken sei, dass diese Linie hier unterschritten wird, wenn sich die beiden Personen *nicht* vollständig innerhalb der Rekonstruktionszone befinden.

Zur Übersicht sind in Abbildung 4.23 wieder die Mittelwerte der Kurven in Bezug auf die Plausibilisierungsfunktionen vom Typ R zusammengefasst.

4.4.3. Vergleich mit der Methode der Occlusion Masks

Bei der Methode der Occlusion Masks [Guan 06, Ladi 08a] werden potentiell sicht-verdeckende Gegenstände genauso behandelt wie die als gesuchte Objekte klassifizierten Bereiche der segmentierten Kamerabilder. In Abbildung 4.24 (a) sieht man drei Kamerabilder zum Zeitpunkt mit der Nummer 307, in (b) deren Segmentierung, in (c) die jeweils zur Perspektive der Kamera zugehörige Occlusion Mask und in (d) die Zusammenfassung der Segmentierungen und der Occlusion Masks. Die resultierende Rekonstruktion ist in Abbildung 4.16 c zu sehen. Eine weitere resultierende Rekonstruktion zum Zeitpunkt mit der Nummer 509 ist in Abbildung 4.16 d zu sehen. Im Vergleich zur wissens- und sensorbasierten geometrischen Rekonstruktion wirkt die Rekonstruktion mittels Occlusion Masks viel größer und damit auch deutlich schlechter. Bestätigt wird dieser Eindruck durch das Diagramm in Abbildung 4.22: Im Durchschnitt besteht die Rekonstruktion mittels Occlusion Masks etwa aus dem 7,87-fachen an Voxeln gegenüber der wissens- und sensorbasierten geometrischen Rekonstruktion. Sie enthält 9,62% Voxel im Mittel gegenüber der Standardrekonstruktion.

4.4.4. Zusammenfassung und Schlussfolgerungen

In diesem Kapitel wurde die wissens- und sensorbasierte geometrische Rekonstruktion anhand einer Beispielsequenz vorgeführt und analysiert. Es wurden insgesamt sieben unterschiedliche Plausibilisierungsfunktionen eingesetzt – Ausschließbarkeitsfunktionen über Kameras und über das Volumen solider Gegenstände, die globale Volumenfunktion, die richtungsspezifische globale Distanzfunktion, die allgemeine globale Erfassbarkeitsfunktion, die Formfunktion über Hüllen und die Temporalfunktion. Mittels der Plausibilisierungsfunktionen vom Typ E konnte eine Reduktion der Voxel auf etwa 4,12% im Durchschnitt über alle Zeitpunkte der Sequenz gegenüber der Standardrekonstruktion erreicht werden. Die anschließende Anwendung der Plausibilisierungsfunktionen vom Typ R resultiert bei etwa nur noch 1,22% der Voxel im Durchschnitt über alle Zeitpunkte der Sequenz gegenüber der Standardrekonstruktion. Im direkten Vergleich besteht die wissens- und sensorbasierte geometrische Rekonstruktion nur aus rund 12,7% der Voxel im Mittel über alle Zeitpunkte der Sequenz gegenüber der Rekonstruktionsmethode mit-

Mittelwerte der Anzahl verbliebener Voxel:
 Kombination (Ausschließbarkeitsfunktionen Kameras 1-7 und Umwelt) / Volumen-, Distanz-, Form-, Erfassbarkeits-, Zeitfkt.

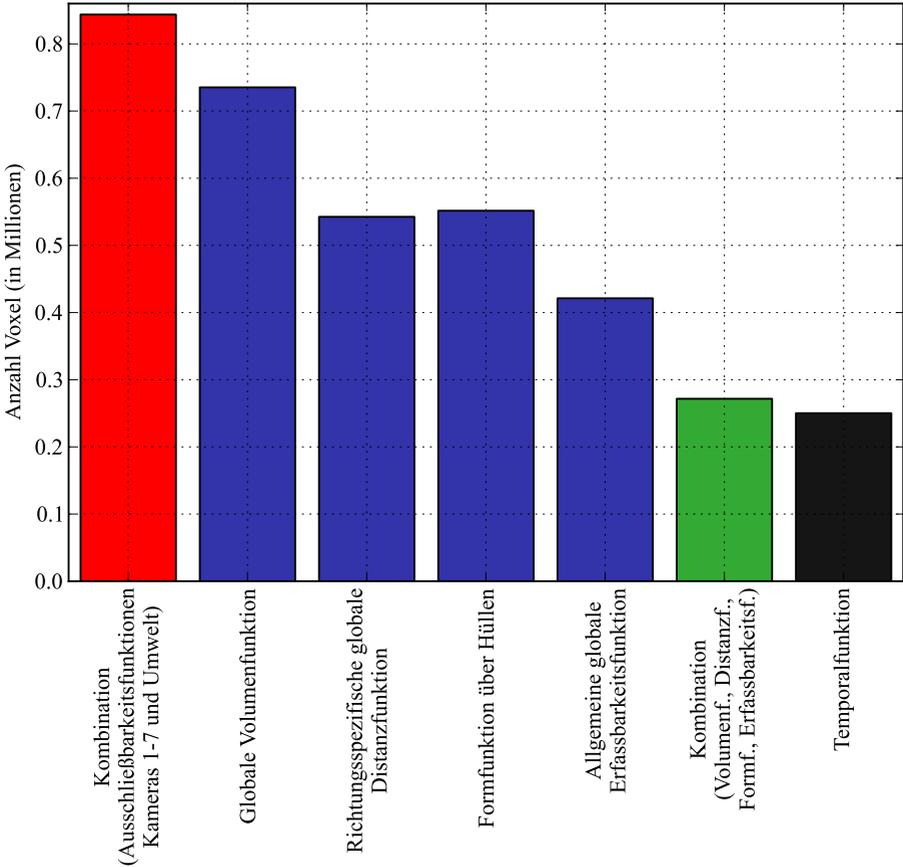


Abbildung 4.23.: Vergleich der Anzahl verbliebener Voxel nach Anwendung aller Plausibilisierungsfunktionen vom Typ E (rot) mit der resultierenden Anzahl an Voxel nach separater Anwendung der Volumenfunktion, der Distanzfunktion, der Formfunktion sowie der Erfassbarkeitsfunktion im Mittel (blau) und der resultierenden Anzahl an Voxel nach Kombination dieser vier Plausibilisierungsfunktionen im Mittel (grün). Die resultierende Anzahl an Voxel nach Anwendung der Temporalfunktion auf die Kombination der vier Plausibilisierungsfunktionen ist dunkelgrau dargestellt.

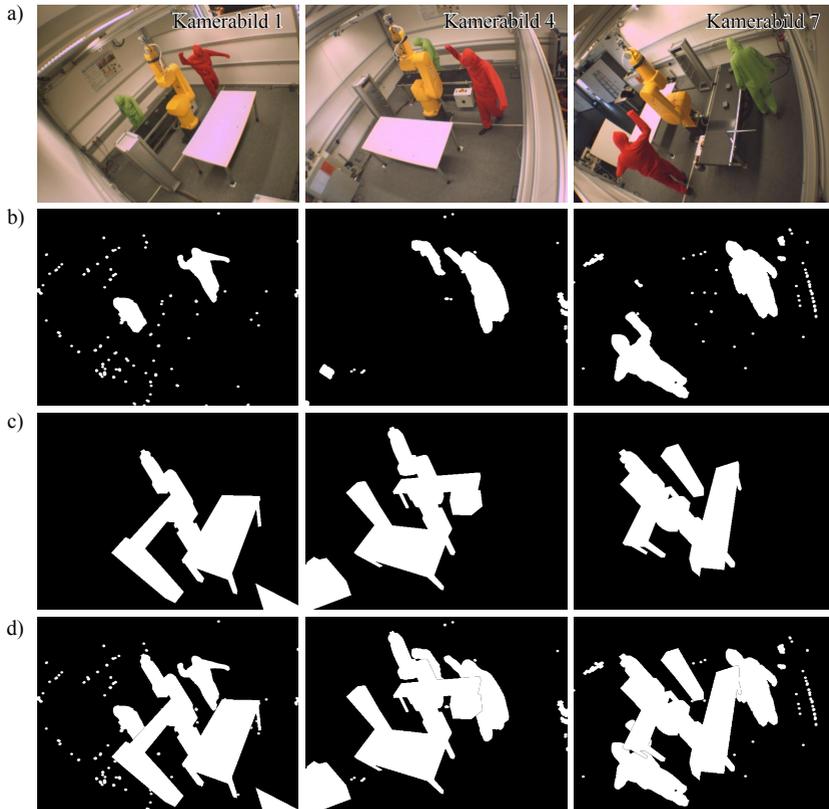


Abbildung 4.24.: a) Bilder der Kameras 1, 4 und 7 zum Zeitpunkt mit der Nummer 307, b) deren Segmentierung; c) Occlusion Mask der jeweiligen Perspektive der Kameras 1, 4 und 7; d) Zusammenfassung der Segmentierungen und der Occlusion Masks.

tels Occlusion Masks und ist daher deutlich besser.

4.5. Zusammenfassung und Schlussfolgerungen

Dieses Kapitel hat gezeigt, dass es problemlos möglich ist, die im Kontinuierlichen eingeführte wissens- und sensorbasierte geometrische Rekonstruktion (Kapitel 3) voxelbasiert umzusetzen. Es ist aber auch deutlich geworden, dass, obwohl das meiste direkt aus dem Kontinuierlichen übernommen werden kann, eine approximierende Repräsentation, wie es die Voxelraumbetrachtung ist, einige Tücken bereithält: Es muss stets darauf geachtet werden, dass die approximierende Repräsentation konservativ gehandhabt wird, also das Resultat einer Plausibilisierungsfunktion tatsächlich eine geometrische Rekonstruktion darstellt und nicht fälschlicherweise Raumbereiche verwirft, in welchen sich gesuchte Objekte aufhalten könnten.

Wählt man beispielsweise im Rahmen der Formfunktion in einer approximierenden Repräsentation eine teilweise größere Hülle als sie im Kontinuierlichen tatsächlich ist, so kann es passieren, dass gesuchte Objekte aus der Hülle im Kontinuierlichen zwar korrekt herausragen, aber nicht in der approximierenden Repräsentation und folglich fälschlicherweise “verschluckt” werden. Daher muss das Volumen der Hülle in einer approximierenden Repräsentation eine Teilmenge der tatsächlichen Hülle im Kontinuierlichen sein – im Rahmen der voxelbasierten Umsetzung wurde dies durch die Verwendung der verkleinerten Voxelapproximation erreicht.

Eine weitere Schwierigkeit ist, dass nur für *gesamte* Voxel Aussagen gemacht werden können: Wird ein Voxel von der Oberfläche eines Gegenstandes geschnitten, so bereitet dies Schwierigkeiten beispielsweise im Rahmen ausschließbarer Räume: Könnte doch im Kontinuierlichen die eine Hälfte des Voxelvolumens durch den soliden Gegenstand weggeschnitten werden und die andere Hälfte durch die Auswertung eines Sensors, so ist das in der diskreten Darstellung nicht möglich. Abhilfe lässt sich – wie gezeigt – durch Überlappung von ausschließbaren Räumen schaffen. Ein vergleichbares Problem konnte im Fall der Formfunktion über die “einfachen” Hüllen beobachtet werden. Eine vergleichbare Lösung nutzt die Überlappung des Hüllenvolumens mit einem darin enthaltenen ausschließbaren Raumbereich.

Auch sensorseitige Diskretisierungen, wie sie im diskutierten Fall von Kameras präsent sind, bedürfen einer genauen Analyse im Rahmen ausschließbarer Raumbereiche und der Erfassbarkeit.

Ergänzend muss hinzugefügt werden, dass in der voxelbasierten Implementierung kleinere

lokale zeitliche Optimierungen eingesetzt wurden. Die drei wesentlichen Optimierungen sind:

1. Plausibilisierungsfunktionen vom Typ R arbeiten ausschließlich auf Regionen. Jede solche voxelbasierte Plausibilisierungsfunktion muss auf gleiche Weise eine gegebene geometrische Rekonstruktion in Regionen umwandeln (clustern), um dann die Regionen bezüglich Plausibilität auszuwerten. Ebenso müssen viele dieser voxelbasierten Plausibilisierungsfunktionen eine Region auf Überschneidung mit dem Rand der Rekonstruktionszone prüfen. Beide Schritte wurden in der Implementierung aus den Plausibilisierungsfunktionen ausgelagert, sodass diese pro Rekonstruktionsschritt nur einmal berechnet werden müssen (vgl. Definition 6).
2. Sowohl die ausschließbaren Raumbereiche über Farbkameras als auch die Erfassbarkeit benötigen Tiefenkarten der statischen und dynamischen Umwelt. Die Berechnung jener Karten wurde ausgelagert, sodass sie nur einmal berechnet werden müssen und für beide Plausibilisierungen genutzt werden können.
3. Wo es möglich war wurde die OpenMP-Erweiterung [Open] des Compilers zur Parallelisierung von for-Schleifen auf den verfügbaren CPU-Kernen genutzt. Außerdem wurde der Code mit der Optimierungsstufe 3 (`g++ -O3`) kompiliert.

Grundsätzliche und tiefer greifende Optimierungen sind Thema des Ausblicks (Kapitel 7.6).

Die Analyse einer konkreten Beispielsequenz hat belegt, dass die wissens- und sensorbasierte geometrische Rekonstruktion praktisch anwendbar ist und die resultierende geometrische Rekonstruktion deutlich besser im Vergleich zur Rekonstruktionsmethode mittels Occlusion Masks ist.

5. Anwendung

Die wissens- und sensorbasierte geometrische Rekonstruktion wird während des Entstehens dieser Dissertation in Zusammenarbeit mit dem Fahrzeughersteller BMW in München in einem konkreten Anwendungsfall der Mensch/Roboter-Koexistenz getestet. Ziel ist, einen mittels Roboter vollautomatisierten Prozessschritt in der Fahrzeugherstellung dem Werker zugänglich zu machen, um den Raum- und Zeitbedarf insofern zu verringern, als auf dem gleichen Abschnitt auf dem Produktionsband gleichzeitig weitere Aufgaben durch den Werker erledigt werden können. Es handelt sich bei dem Prozessschritt um eine mit trennenden Schutzeinrichtungen versehene Roboterzelle, in welcher ein Roboter mittels einer am Tool Center Point angebrachten Walze den Innenteil gegen den Außenteil einer Fahrzeugtür presst, auf den zuvor eine Kleberaube aufgetragen wurde. Dieser Vorgang stellt die Abdichtung der Fahrzeugtür dar. Ein SafetyEye [Pilz] als Vergleichssystem konnte aufgrund der großen zu überwachenden Fläche und der relativ niedrigen Decke von etwa 3,5 m Höhe nicht eingesetzt werden. Dieses Kapitel beschreibt die Aufgabenstellung (Kapitel 5.1), deren Umsetzung (Kapitel 5.2), eine detaillierte Beschreibung der verwendeten Plausibilisierungen (Kapitel 5.3) sowie der Ergebnisse (Kapitel 5.4) und endet mit den Schlussfolgerungen (Kapitel 5.5). Die verwendeten Symbole und Bezeichnungen können in Anhang B nachgeschlagen werden.

5.1. Aufgabenstellung

Aufgabe ist, die trennenden Schutzeinrichtungen der Roboterzelle durch die wissens- und sensorbasierte geometrische Rekonstruktion zu ersetzen (Abbildung 5.1), um eine Mensch/Roboter-Koexistenz zu ermöglichen, indem die Robotergeschwindigkeit in Abhängigkeit zur Distanz zum nächsten potentiellen gesuchten Objekt – also dem Werker – angepasst wird. Im Falle einer Annäherung zwischen Werker und Roboter wird dessen Geschwindigkeit bis hin zum Stillstand reduziert. Wächst die Distanz hingegen, so bewegt sich der Roboter wieder schneller.

5. Anwendung



Abbildung 5.1.: Foto der Roboterzelle, in welcher mittels der *wissens- und sensorbasierten geometrischen Rekonstruktion* die Mensch/Roboter-Koexistenz erprobt werden soll. Zu sehen sind ein Roboter, eine Autotür, ein Dummy sowie Regale, ein Tisch und Stühle. Rechts im Hintergrund befindet sich der Visualisierungsmo-
nitor. (Foto: BMW, 2011)

Neben der Distanzberechnung sollen Signale ausgegeben werden, sobald eine konfigurierbare Minimaldistanz zwischen Werker und Roboter unterschritten wird oder die Änderung der Distanz zwischen zwei zeitlich aufeinanderfolgenden Rekonstruktionen einen festlegbaren Wert überschreitet. Außerdem soll ein Signal ausgegeben werden, sobald ein Werker eine konfigurierbare statische Zone innerhalb des überwachten Bereiches betritt.

5.2. Umsetzung

Der Aufbau der Szenerie des Abdichtungsprozesses befindet sich in einer Umgebung von $6\text{ m} \times 6\text{ m} \times 3,5\text{ m}$ Ausmaß und umfasst einen Industrieroboter (ABB IRB 4600), eine Fahrzeugtür (BMW) in einer Halterung, einen Tisch, zwei Regale und ein Zuführregal (Abbildung 5.2). Die Gegenstände sind neben dem Zweck zur Visualisierung auch für Plausibilisierungen angepasst modelliert – z.B. expandiert oder geschrumpft und gegebenenfalls in entsprechende Kugelmodelle transformiert.

Die wissens- und sensorbasierte geometrische Rekonstruktion ist mittels ausschließbarer Raumbereiche über Farbkameras und über das Volumen statischer bekannter Gegenstände, Formfunktion über Hüllen bekannter Gegenstände, globalem Volumen und zeitlicher Plausibilität realisiert, was im nachfolgenden Kapitel detailliert beschrieben wird. Die Bilder von acht mittels [Svob 05] kalibrierten Kameras (SVS-VISTEC eco204CVGE Farbe, 1024×768 bei $\leq 47\text{ Hz}$) werden auf Vorrechnern (Siemens Simatic IPC647C, Core I3-330E, 2,13 GHz, 2 GB RAM) in Hintergrund und Vordergrund segmentiert (Codebook-basiertes Background-Subtraction nach [Brad 08] und morphologische Operatoren) und an den Zentralrechner (Siemens Simatic IPC547C, Core 2 Quad Q9400, 2,66 GHz, 8 GB RAM) weitergeleitet. Dort wird die wissens- und sensorbasierte geometrische Rekonstruktion gemäß der oben genannten Plausibilisierungen berechnet. Zwischen den rekonstruierten potentiellen gesuchten Objekten und dem Roboter in Form eines vereinfachten Kugelmodells (Abbildung 5.3) wird die Distanz berechnet und der Robotersteuerung (ABB) mitgeteilt. Das Kugelmodell besteht aus 20 Kugeln und hat gegenüber den beweglichen Gliedern des exakten Modells eine minimale positive Abweichung von 0 mm und eine maximale positive Abweichung von 350 mm. Die berechnete Distanz ist damit eine konservative Abschätzung, weil sie kleiner oder gleich der tatsächlichen Distanz ist. Da der Aufwand zur Distanzberechnung linear mit der Anzahl der Kugeln des Robotermodells in der aktuellen Implementierung steigt, ist eine geringe Anzahl von Kugeln vorteilhaft.

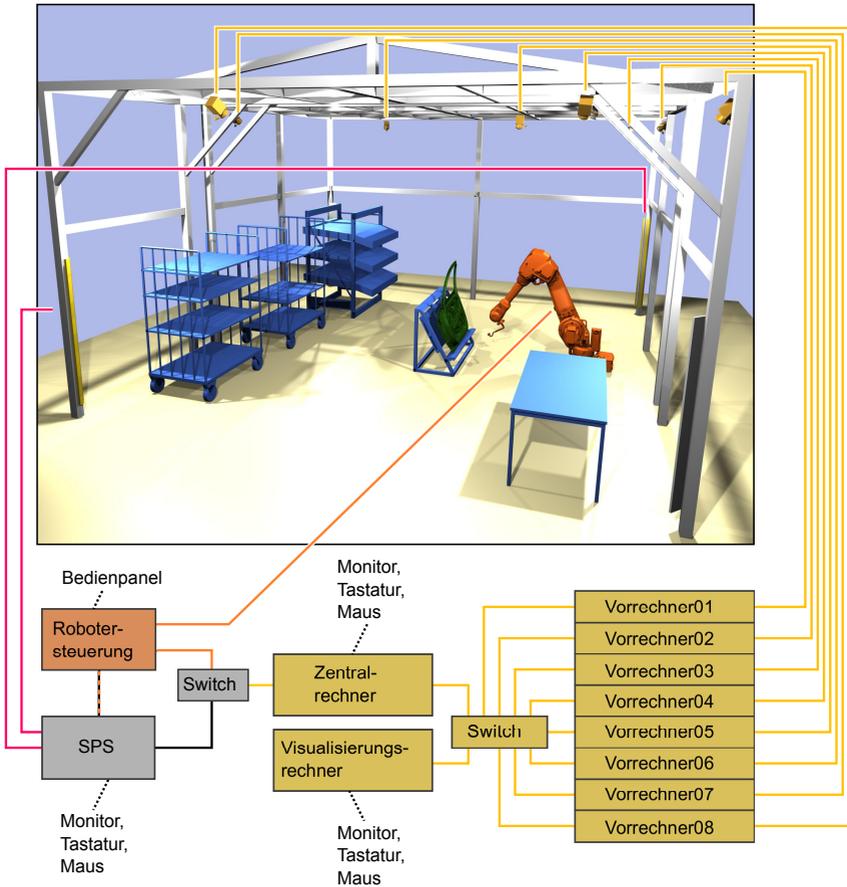


Abbildung 5.2.: Aufbau des roboterunterstützten Abdichtungsprozesses im Kontext der Mensch/Roboter-Koexistenz: Die Bilder der Farbkameras werden nach einer Vorverarbeitung auf den Vorrechnern an den Zentralrechner weitergeleitet. Der Zentralrechner berechnet die geometrische Rekonstruktion und die kürzeste Distanz zu dem Robotermodell. Die berechneten Daten werden an den Visualisierungsrechner zur Darstellung weitergeleitet. Die Distanz wird der Robotersteuerung mitgeteilt. Umgekehrt erhält der Zentralrechner von der Robotersteuerung die aktuellen Gelenkwinkel des Roboters. Über die SPS werden zusätzliche Lichtgitter betrieben, da die Umsetzung der geometrischen Rekonstruktion nicht sicher implementiert ist. Weiterhin teilt die SPS die aktuell angewählte Betriebsart dem Zentralrechner mit und erhält von diesem die gewünschten Signale.

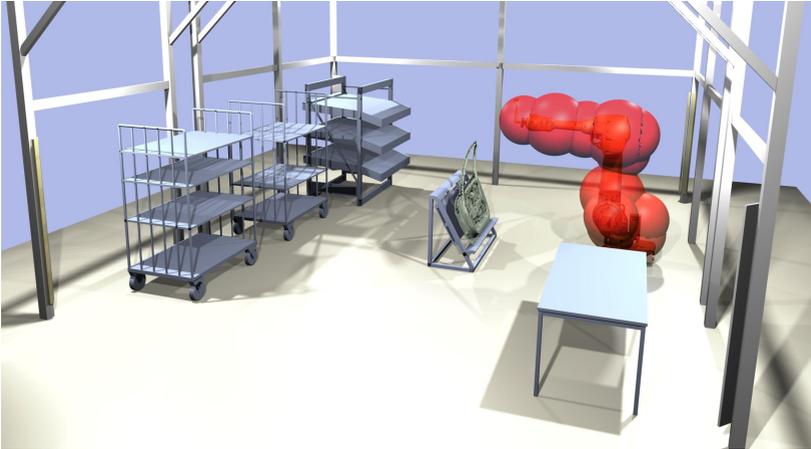


Abbildung 5.3.: Darstellung des Roboter-Kugelmodells (transparent rot) zur Distanzberechnung. Die Distanz wird zwischen diesem Kugelmodell und den geometrisch rekonstruierten gesuchten Objekten berechnet.

Der Zentralrechner erhält von der Robotersteuerung alle 20 ms die aktuellen Gelenkwinkel des Roboters über das RRI (Robot-Reference-Interface, ABB). Die Voxel potentieller gesuchter Objekte, die Distanz und die aktuell von der SPS (Siemens) vorgegebene Betriebsart wird dem Visualisierungsrechner (Siemens Simatic IPC547C, Core 2 Quad Q9400, 2,66 GHz, 8 GB RAM) zur Darstellung mitgeteilt. Die SPS gibt eine der Betriebsarten *HAND*, *AUTOMATIK*, *MMK*, *MMK TEST* und *REFERENZ* vor, wobei MMK für Mensch/Maschine-Koexistenz steht. In den Betriebsarten *MMK* und *MMK TEST* wird das System zur wissens- und sensorbasierten Rekonstruktion mit anschließender Distanzberechnung aktiviert und die in der Aufgabenstellung gewünschten Signale erzeugt und an die SPS weitergeleitet. In der Betriebsart *REFERENZ*, welche angewählt werden muss, bevor zum ersten Mal eine MMK-Betriebsart angewählt wird, startet der Zentralrechner auf den Vorrechnern das Einlernen des Hintergrundmodells. In dieser Betriebsart verfährt der Roboter alle einprogrammierten Bewegungen in verlangsamer Geschwindigkeit, sodass dieser später in den segmentierten Bildern stets als Hintergrund klassifiziert wird. Durch die verlangsamte Geschwindigkeit wird erreicht, dass möglichst alle Bildbereiche, die der Roboter bei seiner Bewegung überstreicht auch durch die Kameras erfasst und schließlich im Hintergrundmodell berücksichtigt werden können. Die Kommunikation

5. Anwendung

erfolgt über Gigabit-Netzwerk-Hardware. Zur Vermeidung einer Netzwerküberlastung existiert für jede Kamera/Vorrechner-Verbindung sowie für die Kommunikation zwischen Vorrechnern, Zentralrechner und Visualisierungsrechner und für die Kommunikation zwischen Zentralrechner, SPS und Robotersteuerung jeweils ein separates Netzwerk. Da die derzeitige Umsetzung der wissens- und sensorbasierten geometrischen Rekonstruktion nicht sicher bezüglich geltender Normen ist, wird die sichere Ausführung des Prozesses durch weitere Maßnahmen gewährleistet: Für Tests, bei denen sich Mensch- und Roboterpfade überschneiden, wird lediglich ein Dummy eingesetzt. Für Tests ohne Überschneidung der Pfade hält der eintretende Mensch einen Totmannschalter. Außerdem sind zusätzlich Lichtgitter (Fa. Sick) aktiv, welche ein unkontrolliertes Hineinlaufen überwachen und gegebenenfalls ein Notaus in der Robotersteuerung auslösen. Weitere Sicherheitsmaßnahmen, wie z.B. Begrenzung der Achsbereiche des Roboters werden ebenfalls eingesetzt.

Die Konfigurierbarkeit der wissens- und sensorbasierten Rekonstruktion und Distanzberechnung über eine Konfigurationsdatei umfasst unter anderem die Angabe der Kamerakalibrierungsdateien, verschiedener Kameraparameter (z.B. Bildgrößen, Bildwiederholrate, Belichtungszeit, Verstärkungsregelung), Background-Subtraction-Parameter (u.a. Anzahl von Iterationen auszuführender Open- und Close-Operationen), Voxelraum-Grenzen und Auflösung, Überwachungsbereiche, zu verwendende Kugelmodelle für elementbasierte und regionenbasierte Exklusion und Distanzberechnung, globales Minimalvolumen zu rekonstruierender Objekte, Minimaldistanz für Signalausgabe, minimale Abstandsänderung zwischen zwei aufeinanderfolgenden Rekonstruktionen für Nothalt-Signal, Minimal- und Maximaldistanz zur Geschwindigkeitsregelung des Roboters, Kommunikationsports für SPS, Robotersteuerung und Visualisierungskomponente, etc.

Die Echtzeit-Visualisierung beinhaltet eine simulierte Umgebung mit automatischer Perspektivänderung auf die Szene (virtueller Kameraflug). Der Kameraflug wird abhängig von Betriebsart und aktueller Distanz angepasst, sodass der Betrachter vor dem Bildschirm möglichst das jeweilige Interessante zu sehen bekommt. Neben dem was in Abbildung 5.2 gezeigt ist, enthält die Szene die aktuell belegten Voxel der wissens- und sensorbasierten geometrischen Rekonstruktion, die berechnete Distanz (mit Verbindung von Start- und Endpunkt), das Lichtgitter und die aktivierte Betriebsart.

5.3. Plausibilisierungen

Zur Plausibilisierung werden ausschließbare Raumbereiche über Farbkameras und das Volumen statischer bekannter Gegenstände, Formfunktion über Hüllen der Volumina statischer und dynamischer bekannter Gegenstände, globales Volumen und zeitlicher Aspekt eingesetzt (vgl. Tabelle 3.2). Zudem ist festgelegt, dass der Mensch weder in das Bounding-Box-Volumen der Regale, noch in das des Tisches eindringt. Außerdem wird angenommen, dass sich die Voxel des geometrisch rekonstruierten Menschen mit denen der zeitlich direkt darauffolgend berechneten geometrischen Rekonstruktion des Menschen überschneiden. Weiterhin wird garantiert, dass der Überwachungsbereich beim Starten einer der MMK-Betriebsarten frei von gesuchten Objekten ist. Der Bereich hinter dem Roboter – in Abbildung 5.2 rechts des Industrieroboters – wird als nicht zugänglich deklariert. Wäre dies nicht so, dann würde die berechnete Distanz nicht größer sein können als vom Roboter zu diesem Bereich, da dort keine Rekonstruktion vorgenommen wird und somit ein Mensch angenommen werden muss. Ebenfalls wird ein Eindringen in den überwachten Bereich von oben ausgeschlossen. Die Plausibilisierungsfunktionen lassen sich daher wie folgt angeben:

Als ausschließbare Raumbereiche über Farbkameras werden die Bounding-Box-Volumina der Regale und des Tisches sowie einem Modell der Fahrzeughür in ihrer Halterung und einem Kugelmodell des Industrieroboters eingesetzt (Abbildung 5.4). Die Verwendung der Bounding-Box-Volumina stellt in diesem Fall eine zulässige, weil konservative Betrachtung dar – tatsächlich können die Kameras nämlich weiter als nur bis zur Bounding-Box “sehen”. Auf Basis dieser Modellierung lässt sich berechnen, bis zu welcher Sichttiefe pro Pixel einer Farbkamera Aussagen über die Abwesenheit eines gesuchten Objektes gemacht werden können. Besonders wichtig ist die genaue Modellierung des Roboters, da an diesem möglichst kein Rekonstruktionsartefakt entstehen darf. Das Robotermodell besteht aus 1918 Kugeln und weicht maximal um 30 mm positiv gegenüber den beweglichen Gliedern des exakten Modells ab.

Zum ausschließbaren Raumbereich über Volumina statischer bekannter Gegenstände werden ebenfalls die o.g. Bounding-Box-Volumina und ein Modell der Fahrzeughür mit Halterung verwendet (Abbildung 5.5). Aufgrund der dünnen Roboterglieder lohnt eine solche Plausibilisierung für den Roboter nicht: Sie würde Rechenzeit kosten, ohne dass viele Voxel eliminiert werden könnten. Der obere dünne Teil der Autotür kann ebenfalls vernachlässigt werden, da die Voxel zu groß sind, als dass sie verworfen werden können. Außerdem wird eine virtuelle Wand

5. Anwendung

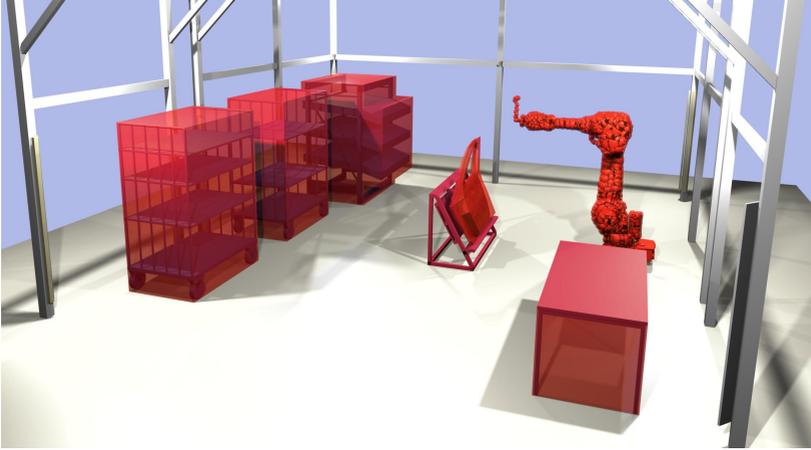


Abbildung 5.4.: Darstellung der Objekt-Modellierung (transparent rot) für die ausschließbaren Raumbereiche über Farbkameras. Alle Elemente der Raumbereiche, welche von den Kameras als *frei* gesehen werden, können eliminiert werden.

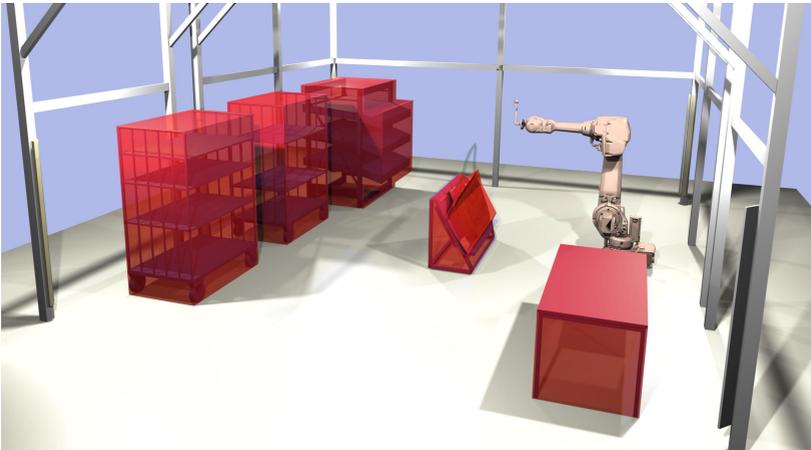


Abbildung 5.5.: Darstellung der Objekt-Modellierung (transparent rot) für die ausschließbaren Raumbereiche über die Volumina statischer bekannter Gegenstände. Alle Voxel innerhalb der modellierten Raumbereiche sind nicht plausibel, da ein Eindringen des Menschen in diese Bereiche ausgeschlossen wurde. Insofern können alle darin liegenden Voxel eliminiert werden.

hinter dem Industrieroboter und eine virtuelle Decke modelliert und in der Ausschließbarkeitsfunktion verwendet, was zulässig ist, da diese Bereiche als nicht zugänglich deklariert wurden. Dadurch werden Rekonstruktionsartefakte unterdrückt, die über den andernfalls “offenen” Rand in die Überwachungszone eindringen könnten und sich in der unmittelbaren Nähe des Roboters als störend auswirken würden. Auch in dieser Plausibilisierung ist die Verwendung der Bounding-Box-Volumina der Regale und des Tisches aufgrund der Festlegung, dass der Mensch in diese nicht eindringt, zulässig.

Für die Formfunktion über Hüllen statischer und dynamischer bekannter Gegenstände werden Hüllen um die o.g. Bounding-Box-Volumina, um das Modell von Fahrzeugtür und Halterung modelliert und ein expandiertes Kugelmodell für den Industrieroboter erzeugt (Abbildung 5.6).

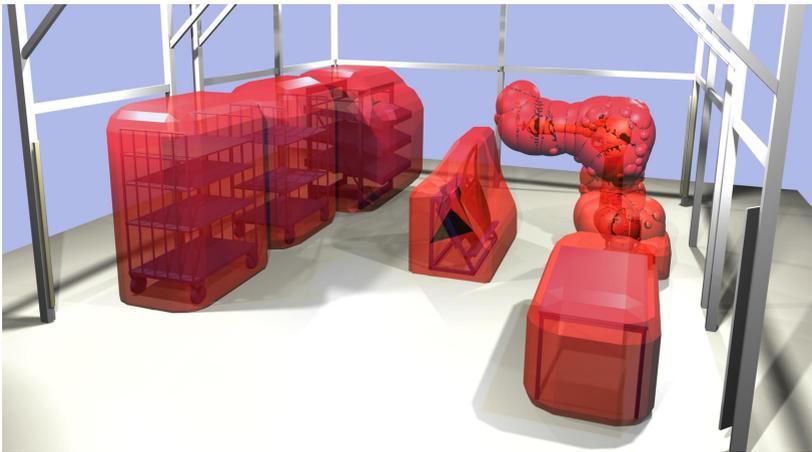


Abbildung 5.6.: Darstellung der Objekt-Modellierung (transparent rot) für die Formfunktion über Hüllen statischer und dynamischer bekannter Gegenstände. Würde ein Mensch sich noch so dicht an einem der Gegenstände befinden, so würde immer noch ein Teil von ihm aus den modellierten Hüllen ragen. Geometrisch rekonstruierte Voxelregionen, die nicht aus diesen Hüllen herausragen, können also kein Mensch sein und damit verworfen werden.

Das Kugelmodell des Roboters besteht aus 360 Kugeln und hat gegenüber den beweglichen

5. Anwendung

	<i>Voxelraumauflösung pro Dimension</i>	<i>Voxel- kantenlänge</i>	<i>Bildauflösung pro Dimension</i>	<i>Rekonstruktionen pro Sekunde</i>
<i>Konfiguration 1</i>	(126, 90, 66)	4,52 cm	(300, 175)	~ 3
<i>Konfiguration 2</i>	(63, 45, 33)	9,05 cm	(200, 150)	~ 10

Tabelle 5.1.: Vergleich der der beiden getesteten Konfigurationen. Es werden etwa drei geometrische Rekonstruktionen pro Sekunde in der hohen und etwa zehn geometrische Rekonstruktionen pro Sekunde in der niedrigen Auflösung erreicht.

Gliedern des exakten Modells eine minimale positive Abweichung von 200 mm und eine maximale positive Abweichung von 300 mm.

Aufgrund der Annahme, dass sich die Voxel zweier aufeinanderfolgender geometrischen Rekonstruktionen eines Menschen überschneiden, lässt sich die zeitliche Plausibilisierung mit Radius Null anwenden. Außerdem wird ein Minimalvolumen von 5000000 mm^3 , was in etwa einem Menschen mit einer Masse von 50 kg entspricht, zur Plausibilisierung verwendet.

5.4. Ergebnisse

Die Ergebnisse beschränken sich hier auf die grobe Betrachtung der Anzahl der geometrischen Rekonstruktionen pro Sekunde auf dem Zentralrechner für zwei unterschiedliche Konfigurationen und der Beschreibung von Beobachtungen und deren Erklärungen. Dies dient zur Einschätzung der Leistungsfähigkeit der derzeitigen Implementierung in einer realen Anwendung. Die Rekonstruktionsdauer hängt von vielen Faktoren ab, wie beispielsweise der Anzahl, der Beschaffenheit und der eingesetzten Modelle der statischen und dynamischen bekannten Gegenstände, der Bild- und Voxelraumauflösung, der Anzahl an Kameras, deren Auswertungsreihenfolge und deren Platzierungen sowie der Anzahl an Menschen und deren Positionen innerhalb und außerhalb des überwachten Bereiches. An dieser Stelle werden lediglich zwei unterschiedliche Bild- und Voxelraumaufösungen mit einem einzigen Menschen im überwachten Bereich bei ansonsten gleichem, wie oben beschriebenen Aufbau getestet (Tabelle 5.1).

Die Konfigurationen unterscheiden sich um den Faktor zwei in jeder Dimension der Voxelraumauflösung und um den Faktor drei Halbe in jeder Dimension der Bildauflösung. Eine Rekonstruktionsdauer von etwas über 300 ms in der ersten Konfiguration lässt Wünsche nach einer schnelleren Ausführung und damit schnelleren Reaktion des Systems offen, während eine

Voxelkantenlänge von etwas über 9 cm Wünsche nach einer höheren Auflösung offen lässt, insbesondere im Hinblick darauf, dass einige regionenbasierte Plausibilisierungen bei höherer Auflösung Regionen besser als Artefakte identifizieren können –ragt beispielsweise ein Voxel mit 9 cm Kantenlänge noch aus einer Hülle bei der Formfunktion über Hüllen heraus, so tut er das möglicherweise bei einer Kantenlänge von 2 cm *nicht*. Dies kann eine Rolle in der folgenden Beobachtung spielen:

In einigen Fällen konnte beobachtet werden, dass wenn der Mensch den Roboter berührt, die durch die ausschließbaren Raumbereiche über Kameras resultierende visuelle Hülle um den Roboter in Form von Voxeln nicht mehr von den restlichen Plausibilisierungsfunktionen eliminiert werden kann – selbst dann nicht, wenn sich der Mensch wieder entfernt. Dieses Verhalten lässt sich dadurch erklären, dass die visuelle Hülle aufgrund einer ungünstigen Roboterstellung bezüglich der Kameraplatzierung aus der Hülle der Formfunktion über Hüllen herausragt und zudem größer als das globale Minimalvolumen ist und somit nicht mehr als Artefakt erkannt und eliminiert werden kann. Infolgedessen ist es notwendig, erneut in eine MMK Betriebsart zu wechseln, welchem ein Reset der rekonstruierten Objektbereiche einhergeht und damit auch den Bereich um den Roboter wieder freigibt. Dies ist zulässig, da der Überwachungsbereich beim Starten einer MMK-Betriebsart frei von gesuchten Objekten sein muss. Eine alternative Lösung ist, den Roboter bei Minimaldistanz sehr langsam (und damit zulässig) weiter bewegen zu lassen, in der Hoffnung eine günstige Roboterstellung bezüglich der Kameraplatzierung zu erreichen, in der die visuelle Hülle in Form von Voxeln wieder komplett in der Hülle der Formfunktion über Hüllen liegt und damit als Artefakt identifiziert und eliminiert werden kann. Eine Entschärfung dieser Problematik lässt sich durch zusätzliche, geschickt platzierte Sensorik (z.B. weitere Kameras oder Näherungssensoren am Roboter) erreichen. Ebenso spielt die Genauigkeit der realen Roboterplatzierung gegenüber Kamerakalibrierung eine Rolle: Befindet sich der Roboter etwas neben der durch die Kalibrierung angenommenen Position, so wird, falls der Roboter nicht einwandfrei als Hintergrund detektiert wird, eher die Hülle der Formfunktion über Hüllen durchstoßen.

5.5. Schlussfolgerungen

Die wissens- und sensorbasierte geometrische Rekonstruktion wurde erfolgreich zur Distanzberechnung im konkreten Anwendungsfall der Mensch/Roboter-Koexistenz eingesetzt (Abbildung

5. Anwendung

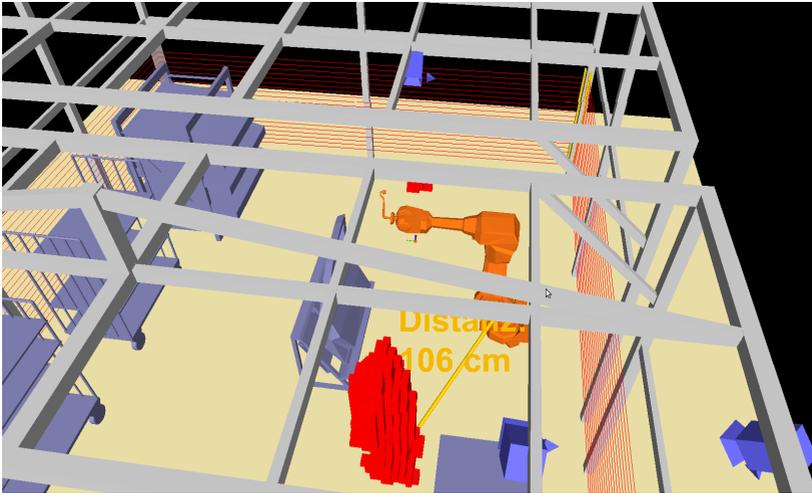


Abbildung 5.7.: Screenshot der Visualisierung. Die aktuelle Distanz zwischen Werker und Roboter beträgt – unter der Annahme einer exakten Kamerakalibrierung und einer korrekten Detektion im Kamerabild – mindestens 106 cm.

5.7). Je nach Bild- und Voxelraumaufösung erfolgen die Rekonstruktionen und Bestimmungen der Distanz zwischen drei und zehn Hertz. Die beschriebenen zusätzlichen Sicherheitsmaßnahmen sind in diesem Rahmen notwendig, aber für einen Einsatz in der Produktion natürlich inakzeptabel. Es besteht daher für einen solchen Anwendungsfall der dringende Bedarf an einer sicheren und normenkonformen Umsetzung der wissens- und sensorbasierten geometrischen Rekonstruktion. Eine schnellere Berechnung bei gleichzeitig höherer Ortsauflösung ist wünschenswert.

6. Schlussfolgerungen

Mit der wissens- und sensorbasierten geometrischen Rekonstruktion wurde erstmals ein formales Rahmenwerk geschaffen und präsentiert, welches die automatisierte Ermittlung und geometrische Beschreibung von potentiellen Aufenthaltsorten gesuchter Objekte innerhalb eines definierten Raumes auf Basis von Wissen und Sensoren umfassend ermöglicht. Unter *Wissen* werden hierbei Informationen bezüglich des Raumes, beispielsweise ein geometrisches Umweltmodell und Informationen bezüglich der gesuchten Objekte, beispielsweise Mindestvolumina, Maximaldistanzen, Geschwindigkeiten etc. verstanden. Die Auswertung unterschiedlichster Sensoren liefert Informationen bezüglich An- bzw. Abwesenheit gesuchter Objekte.

Mittels eines systematischen Ausschlussverfahrens werden all jene Informationen schrittweise genutzt, um Raumbereiche zu verwerfen, in welchen sich garantiert kein gesuchtes Objekt aufhalten kann. Diese schrittweise Verbesserung des Resultats ist über sogenannte Plausibilisierungsfunktionen realisiert, die auf der Ausnutzung jeweils einer speziellen Teilinformation beruhen. Die gezielte Anwendung aller in einem konkreten Fall zur Verfügung stehenden Plausibilisierungsfunktionen resultiert in einer geometrischen Beschreibung potentieller Aufenthaltsorte gesuchter Objekte, deren Güte sich nicht weiter erhöhen lässt.

Die wissens- und sensorbasierte geometrische Rekonstruktion erlaubt damit die einheitliche Integration sowohl existierender Methoden und Systeme – denkbar ist die Einbindung einer Zonenüberwachung mittels des SafetyEye der Firma Pilz [Pilz] in Form des diskutierten Anwesenheitssensors – oder aber auch neuer, im Rahmen dieser Arbeit präsentierter Methoden, wie beispielsweise dem sensorbasierten ausschließbaren Raumbereich in der Ausschließbarkeitsfunktion oder der Formfunktion auf Basis von Hüllen.

Neben der Schaffung eines formalen Rahmenwerkes belegen die vielen eingeführten praxisnahen Plausibilisierungsfunktionen sowie deren Implementierung und Anwendung in einem Beispiel in der Industrie auch die Praxistauglichkeit der wissens- und sensorbasierten geometrischen Rekonstruktion.

6. Schlussfolgerungen

Die voxelbasierte Umsetzung der zunächst im Kontinuierlichen eingeführten wissens- und sensorbasierten geometrischen Rekonstruktion hat gezeigt, was grundsätzlich bei der Verwendung einer approximierenden Repräsentation zu beachten ist, sodass beispielsweise eine geometrische Rekonstruktion auch tatsächlich eine solche ist und nicht fälschlicherweise Aufenthaltsorte gesuchter Objekte verworfen werden.

Der erfolgreiche praktische Einsatz des entwickelten voxelbasierten Systems hat – trotz seiner passablen zeitlichen Performanz durch kleinere lokale zeitliche Optimierungen – Wünsche nach einer sicheren, schnelleren und höher auflösenden Verarbeitung geweckt, was jedoch nicht im Fokus dieser Arbeit lag. Im folgenden Kapitel werden diese und weitere Aspekte im Rahmen eines Ausblicks skizziert und kurz diskutiert.

7. Ausblick

Mit dieser Arbeit besteht die Möglichkeit, basierend auf Informationen, Aufenthaltsorte gesuchter Objekte in einem Raum zu ermitteln und geometrisch zu beschreiben. Die Umsetzung (Kapitel 4) und Anwendung (Kapitel 5) der wissens- und sensorbasierten geometrischen Rekonstruktion in der Praxis hat gezeigt, dass der Einrichtprozess wesentlich vereinfacht werden muss und eine sichere und schnellere Ausführung bei gleichzeitig höherer Raumaufösung der Voxel-darstellung wünschenswert ist. Die bereits sehr attraktive abstandsabhängige Geschwindigkeitsregelung eines Roboters im Beispiel der Mensch/Roboter-Koexistenz könnte außerdem durch wesentlich bessere und komplexere Reaktionen hin zu einer echten Kooperation von Mensch und Roboter ausgebaut werden.

Insofern lässt sich der Bedarf an weiteren Arbeiten den drei Bereichen *Eingabe*, *Ausführung* und *Ausgabe* zuordnen (Abbildung 7.1). Themen, welche die Eingabe betreffen, sind *Intuitive Umweltmodellierung* (Kapitel 7.1), *Optimale Sensorplatzierung, -kalibrierung* und *-registrierung* (Kapitel 7.2) sowie *Automatisierte Parameterbestimmung* (Kapitel 7.3). Im Rahmen der Ausführung sind die Themen *Evaluation weiterer Sensortypen* (Kapitel 7.4), *Sicherheit* (Kapitel 7.5), *Ausführungsgeschwindigkeit* und *Genauigkeit* (Kapitel 7.6) angesiedelt. Die *Online-Bahnplanung eines Roboters* (Kapitel 7.8), das *Tracking*, die *Gestenerkennung* (Kapitel 7.7) und die *Mensch/Roboter-Kooperation* (Kapitel 7.8) basierend auf der resultierenden geometrischen Rekonstruktion gesuchter Objekte sind Themen, welche dem Ausgabebereich zugeordnet werden können. Im Folgenden werden einige Ideen zu diesen Themen skizziert und Vorschläge kurz diskutiert.

7.1. Intuitive Umweltmodellierung

Ausschließbare Raumbereiche sowohl über Volumina solider Gegenstände als auch über die Auswertung von Sensoren benötigen ebenso wie Hüllenbetrachtungen im Rahmen der Form-

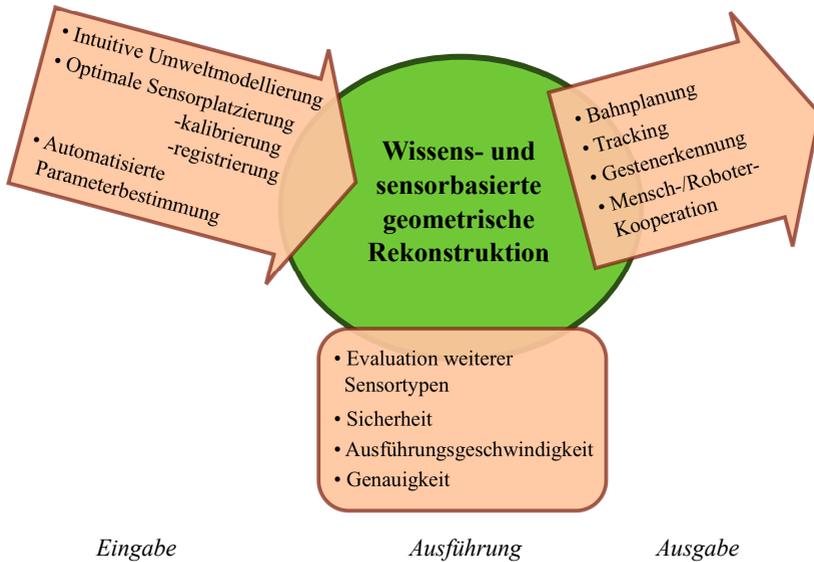


Abbildung 7.1.: Illustration angrenzender Themen, welche den drei Bereichen *Eingabe*, *Ausführung* und *Ausgabe* zugeordnet werden können. Themen im Bereich der *Eingabe* ermöglichen einen komfortableren Einrichtungsbetrieb für die wissens- und sensorbasierte geometrische Rekonstruktion. Eine sichere, schnellere und gleichzeitig genauere Verarbeitung stellen eine Verbesserung der existierenden Implementierung dar und spielen im Rahmen der *Ausführung* eine Rolle. Die ermittelte geometrische Rekonstruktion in der *Ausgabe* kann für wesentlich mehr als nur zur abstandabhängigen Geschwindigkeitsregelung eines Roboters genutzt werden.

funktion ein Geometrie-Modell der Umwelt. Sollen beispielsweise Menschen in einer Roboterzelle geometrisch rekonstruiert werden, in welcher sich neben dem Roboter nur wenige weitere Gegenstände, wie ein Förderband, ein Tisch und ein Regal befinden, so gestaltet sich bereits die Erzeugung eines solch einfachen Modells als sehr zeitaufwändig und zudem ungenau. Ebenso ist die Überprüfung eines solchen Modells nicht trivial.

Es werden daher Methoden benötigt, welche eine intuitive geometrische Modellierung der Umwelt gestatten. Eine Herangehensweise könnte dabei die Methode nach dem Prinzip *Structure-From-Motion* [Dell 00] darstellen, bei welcher der Einrichter jeden Teil des Raumes mit einer Farb- oder alternativ mit einer Tiefenkamera "ausleuchten" muss und dabei gleichzeitig das bereits errechnete und zur Verfügung stehende Modell visualisiert bekommt. Ein Problem könnte jedoch die sukzessive Erzeugung des Modells und der damit einhergehende kumulative Fehler darstellen. Eine aktuelle Publikation auf diesem Gebiet stellt [Izad 11] dar, in welcher der Kinect-Sensor von Microsoft verwendet wird. Die Registrierung des so erzeugten Geometrie-Modells in einem gemeinsamen Arbeitsraum mit bereits anderweitig gegebenen Modellen könnte möglicherweise genauso wie eine Überprüfung der Geometrie-Modelle durch Projektion und Anpassung in einem Kamerabild ("Augmented Reality") durchgeführt werden.

Eine anderer Ansatz basiert auf der Verwendung eines bereits kalibrierten Kameranetzwerkes, in welchem der Benutzer Gegenstände aus Sicht der Kameras mit ähnlichen Mitteln wie bei einer CAD-Software aber auf Basis aufgenommener Bilder modelliert. Hier sind verschiedene Ansätze denkbar; einer davon, welcher visuelle Hüllen einzelner Gegenstände generiert, wurde in der Bachelorarbeit [Baum 09] implementiert und getestet. Als Ausgangsmodell ist dies sicherlich sinnvoll, benötigt jedoch eine Nachbearbeitung real existierender Hohlräume zur Erzeugung eines exakten geometrischen Modells.

7.2. Optimale Sensorplatzierung, -kalibrierung und -registrierung

Beim Aufbau der Anwendung (Kapitel 5) stellte sich die Frage, wo man die verfügbaren acht Kameras denn am sinnvollsten aufhängt, sodass die Anwendung möglichst optimal funktioniert – also das Computersystem zur wissens- und sensorbasierten geometrischen Rekonstruktion möglichst immer den tatsächlichen Abstand zwischen Mensch und Roboter ermitteln kann. Die

7. Ausblick

Frage wurde auf Erfahrungswerten ähnlicher Aufbauten beruhend durch Abschätzung so gut wie möglich beantwortet.

Die Lösung eines solchen Problems sollte natürlich automatisch und ohne Fachwissen für ein Netzwerk an Sensoren – auch unterschiedlichen Typs – computergestützt ermittelt werden können. Hierfür sind typischerweise anwendungsabhängig *unterschiedliche* Optimierungskriterien von Interesse: Zum einen könnte ein möglichst kleiner Gesamtverdeckungsvolumeninhalt bei gegebener Umwelt und gegebener Kameraanzahl gewünscht sein, zum anderen eine möglichst genaue Abstandbestimmung bei gegebener Umwelt und gegebener Kameraanzahl. Ein Ansatz zur Optimierung nach dem zweiten Kriterium (Abstandskriterium) wurde bereits untersucht [Hane 12].

7.3. Automatisierte Parameterbestimmung

Neben der Umweltmodellierung, welche auch als Parameter verstanden werden könnte, gibt es aber auch weitere Parameter, welche vor allem Eigenschaften gesuchter Objekte beschreiben, beispielsweise Volumeninhalt, Hüllendicke und Kameraanzahl in der Erfassbarkeit. Der Parameter *Kameraanzahl* in der Erfassbarkeit lässt sich wie in Kapitel 3.2.3 beschrieben nur schwerlich abschätzen. Daher wäre es sehr hilfreich, wenn die Möglichkeit zur automatischen und konservativen Abschätzung dieses Parameters bestünde. Mit konservativ ist gemeint, dass bei Anwendung des Parameters nicht fälschlicherweise Regionen, welche gesuchte Objekte enthalten können, verworfen werden.

7.4. Evaluation weiterer Sensortypen

In der Anwendung (Kapitel 5) wurden lediglich Farbkameras als Sensoren verwendet. Dabei war der Roboter zeitweise komplett in einer nur schwerlich eliminierbaren Hülle gefangen – nämlich dann, wenn die angebrachten Kameras nicht mehr in der Lage waren, die Hülle als menschen-leer zu identifizieren. In Anbetracht dessen könnte unter anderem die Nutzung einer Haut aus Näherungssensoren direkt am Roboter zur Vermeidung solcher Situationen beitragen. Folglich sollte diese Möglichkeit, verbunden mit dem Einsatz alternativer Sensoren, genutzt und getestet werden.

7.5. Sicherheit

Wie in Kapitel 5 beschrieben, ersetzt die Implementierung in der vorliegenden Form kein Sicherheitssystem. Dies erfordert neben einer redundanten Auslegung großer Teile des Systems auch Untersuchungen bezüglich Detektierbarkeit gesuchter Objekte in den verwendeten Sensoren. Vergleichbare Betrachtungen wurden bei der Entwicklung des SafetyEye der Firma Pilz [Pilz] angestellt. Zu beachten sind im Zuge dessen auch die derzeit sich im Wandel befindenden Normen hierzu.

7.6. Ausführungsgeschwindigkeit und Genauigkeit

Erstrebenswert ist die Erhöhung der Ausführungsgeschwindigkeit zur Berechnung der wissens- und sensorbasierten geometrischen Rekonstruktion bei gleichzeitiger Erhöhung der Raumauflösung. Im Folgenden werden einige Ansatzmöglichkeiten insbesondere zur Optimierung der Ausführungsgeschwindigkeit aufgezeigt, welche es zu überprüfen gilt. Weil die Optimierung der Ausführungsgeschwindigkeit und eine höhere Raumauflösung sich in direkter Weise gegenseitig beeinflussen, ist die Untersuchung zur Erhöhung der Raumauflösung ohnehin mit der Optimierung der Ausführungsgeschwindigkeit einhergehend. Drei wesentliche Ansatzpunkte stellen hierbei die *parallele Verarbeitung*, die *Auswahl geeigneter Datenstrukturen* und *anwendungsspezifische Optimierungen* dar. Durch eine *geschickte Verknüpfung* der Plausibilisierungsfunktionen lässt sich die Ausführungsgeschwindigkeit weiter optimieren.

Optimierung durch parallele Verarbeitung

Je nach gewählter Datenstruktur lassen sich Operationen, wie sie im Rahmen der wissens- und sensorbasierten geometrischen Rekonstruktion benötigt werden, mehr oder weniger gut parallel verarbeiten. Die parallele Verarbeitung kann beispielsweise auf der Recheneinheit des Computers (CPU) und/oder auf der Recheneinheit einer Grafikkarte (GPU) stattfinden. Sie kann entweder *innerhalb* einer Plausibilisierungsfunktion oder auch *über mehrere* Plausibilisierungsfunktionen hinweg greifen. Im ersten Fall liefert die Plausibilisierungsfunktion durch eine interne parallele Verarbeitung schneller ein Ergebnis. Im zweiten Fall werden mehrere Plausibilisierungsfunktionen parallel ausgeführt. Beispielsweise könnten Plausibilisierungsfunktionen vom Typ R die Plausibilität der Regionen gleichzeitig auswerten. Ein Voxel- oder

Octree-basierter Ansatz ist für beide Herangehensweisen geeignet. Beispielsweise wurde in [Ladi 08a, Schi 09b] eine reine Multikamera-Rekonstruktion mittels Voxeln und alternativ Octrees auf der GPU realisiert. In beiden Referenzen werden Verdeckungen nur zu einem gewissen Grad (“Occlusion Masks”) bei der Auswertung der Kamerainformation berücksichtigt.

Optimierung durch Auswahl geeigneter Datenstrukturen

Eine geeignete Datenstruktur muss in jedem Fall robust bezüglich der benötigten Operationen der wissens- und sensorbasierten geometrischen Rekonstruktion sein. Hierzu scheinen Ansätze, welche auf polygonalen Oberflächennetzen operieren, weniger geeignet zu sein, da beispielsweise Mengenoperationen gewisse Schwierigkeiten bereiten [Bier 01, Wang 11]. Gegenüber dem einfachen Voxelansatz, wie in Kapitel 4 beschrieben, ist es denkbar, lediglich Oberflächenvoxel zu verwenden, also das Innere von rekonstruierten Raumbereichen implizit darzustellen. Lassen sich Plausibilisierungsfunktionen auf die Form anpassen, dass sie tatsächlich nur die Oberflächenvoxel statt zusätzlich aller inneren Voxel verarbeiten, so wird typischerweise der zeitliche Aufwand von $O(n^3)$ auf $O(n^2)$ reduziert, wobei n hier die Anzahl der Voxel einer jeden Dimension eines kubischen Bereiches in einem Voxelraum darstellt. Ein solcher Oberflächen-Voxel-Ansatz wurde in [Kuhn 09a, Kuhn 09b] beschrieben. Die Verwendung eines Octrees verspricht eine weitere zeitliche Aufwandsreduktion im Hinblick auf einige der benötigten Operationen. Eine weitere Datenstruktur, welche auf dem Verschnitt von Pixelvolumina mehrerer Kameras beruht und ebenfalls hierarchisch gestaltet werden kann, stellen die sogenannten *Conezel* [Casa 06] dar. Eine Anpassung dieser Datenstruktur an eine statische Umgebung im überwachten Raumbereich wurde in [Kuhn 10] vorgeschlagen.

Man kann vermuten, dass die *unterschiedlichen benötigten Operationen* auf *unterschiedlichen Datenstrukturen* optimal hinsichtlich des zeitlichen Aspektes umgesetzt werden können. Dies legt nahe, dass möglicherweise sogar die Anpassung von Datenstrukturen entsprechend der anzuwendenden Plausibilisierungsfunktion *während* eines Rekonstruktionsprozesses eine höhere Ausführungsgeschwindigkeit zur Folge hat. Es gilt hier, bereits die parallele Verarbeitung der Operationen in die Bewertung mit einzubeziehen und zu berücksichtigen. Ebenso gilt es, die Kosten zur Konvertierung von Datenstrukturen zu beachten.

Optimierung durch Berücksichtigung der Anwendung

Möchte man lediglich den kürzesten Abstand zwischen einem bekannten Gegenstand und der geometrischen Rekonstruktion ermitteln, so ist es ausreichend, wenn die geometrische Rekonstruktion nur in der Nähe dieses bekannten Gegenstandes möglichst exakt ausgewertet wird, um darauf dann den Abstand zu berechnen. Hierzu scheint eine hierarchische Datenstruktur, wie sie der Octree darstellt, besonders geeignet zu sein: Nur derjenige Bereich einer geometrischen Rekonstruktion wird immer weiter verfeinert, welcher sich am nächsten am bekannten Gegenstand, zu welchem die Distanz gemessen werden soll, befindet. Nach jeder Verfeinerung wird er auf Plausibilität geprüft. Ist er nach einigen Iterationen der Verfeinerung nicht mehr plausibel, so wird der nächste Bereich schrittweise verfeinert und so weiter. Dies kann beispielsweise bis zu einer gewissen Tiefe (= Genauigkeit) geschehen oder aber bis ein Ergebnis benötigt wird (“Anytime-Fähigkeit”).

Eine Anytime-Fähigkeit lässt sich durch die Verwendung eines Octrees zwar ohnehin erreichen, jedoch eignet sich die punktuelle Verfeinerung, wie sie hier beschrieben wurde, nicht in jeder Anwendung.

Geschickte Verknüpfung von Plausibilisierungsfunktionen

Eine geschickte Verknüpfung von Plausibilisierungsfunktionen berücksichtigt die Optimierung der gesamten Ausführungsgeschwindigkeit unter Beachtung des Erfolgs einer Plausibilisierungsfunktion, der Ausführungsgeschwindigkeit der Plausibilisierungsfunktion selbst und der Dauer zur eventuellen Konvertierung in die benötigte Datenstruktur. Dabei ist eine Plausibilisierungsfunktion desto erfolgreicher, je mehr Raumbereiche, gemessen am Volumen, verworfen werden können. So ist denkbar, dass mit einer Untersuchung dieser Aspekte eine anwendungsabhängige, zeitoptimale Plausibilisierungspermutation bestimmt werden kann.

7.7. Tracking und Gestenerkennung

Tracking und Gesten- bzw. auch Intentionserkennung auf der geometrischen Rekonstruktion scheinen besonders attraktiv zu sein, da diese die hinsichtlich gegebener Information optimale Eingrenzung gesuchter Objekte darstellt. Verdeckungen müssen nicht explizit berücksichtigt werden, da sie bereits implizit in den dreidimensionalen Daten enthalten sind.

7.8. Bahnplanung und Mensch/Roboter-Kooperation

Im Kontext der Robotik kann die wissens- und sensorbasierte geometrische Rekonstruktion in weitaus komplexerer Art und Weise genutzt werden, als lediglich in einer abstandsabhängigen Geschwindigkeitsregulierung. Denkbar ist eine fortwährende “Online”-Bahnplanung im dynamischen Umfeld, welches durch die wissens- und sensorbasierte geometrische Rekonstruktion ermittelt wird. Ein solcher Ansatz, welcher auf Abständen zwischen dem Roboter und gesuchten Objekten beruht, ist in einer Dissertation [Geck 11] am selben Lehrstuhl parallel zur vorliegenden Arbeit entstanden. Das Vorgängersystem [Eber 03], welches bildbasiert eine Kollisionsdetektion für zukünftige Roboterpositionen errechnet, könnte ebenfalls direkt die wissens- und sensorbasierte geometrischen Rekonstruktion nutzen. Damit ist bereits eine umfangreiche Mensch/Roboter-Koexistenz möglich. In Verbindung mit einer Gestenerkennung und weiteren Kommunikationswegen sowie geeigneter Reaktionen seitens des Roboters wird der Weg hin zu der weitaus komplexeren Mensch/Roboter-*Kooperation* geebnet.

Literaturverzeichnis

- [2D3D 12] 2D3DProc. “Dynamisches 3D-Sehen 2D/3D Datenverarbeitung und -fusion auf Basis der PMD-Technologie”. <http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=22933533> [Stand: 14.07.2012, 15:05 Uhr], Start: 2005; Ende: 2012.
- [AAL 08] AAL. “Ambient Assisted Living”. <http://www.aal-deutschland.de/> [Stand: 14.07.2012, 15:12 Uhr], Seit: 2008.
- [Akti 11] “Aktive daten- und modellbasierte Sensorpositionierung zur 3D-Vermessung”. <http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=27973300> [Stand: 14.07.2012, 15:15 Uhr], Start: 2006; Ende: 2011.
- [Albu 07] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. “The DLR Lightweight Robot – Design and Control Concepts for Robots in Human Environments”. *Industrial Robot: An International Journal*, Vol. 34, pp. 376–385, 2007.
- [Baer 92] A.-J. Baerfeldt. “Cooperation between man and robot: interface and safety”. In: *Robot and Human Communication, 1992. Proceedings., IEEE International Workshop on*, pp. 183–187, Tokyo, Japan, September 1992.
- [Bart 11] S. Bartscher. “Mensch-Roboter-Kooperation in der Produktion: Chancen und Risiken”. *ROBOTWORLD*, Vol. 1./2., pp. 8–9, April 2011.
- [Baum 09] J. Baumgartl. “Modellierung von Objekten im Dreidimensionalen mittels eines kalibrierten Kameranetzwerks”. Bachelorarbeit, Universität Bayreuth, Lehrstuhl für Angewandte Informatik III, Februar 2009.

- [Baum 74] B. G. Baumgart. *Geometric modeling for computer vision*. PhD thesis, Stanford, CA, USA, 1974. AAI7506806.
- [Bier 01] H. Biermann, D. Kristjansson, and D. Zorin. "Approximate Boolean Operations on Free-form Solids". In: *SIGGRAPH '01 Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, August 2001.
- [Bort 11] D. Bortot and K. Bengler. "Experimentelle Entwicklung ergonomisch gestalteter Mensch-Roboter-Kooperationen". *Ergonomie aktuell*, http://www.lfe.mw.tum.de/_media/de/institute/zeitung12.pdf [Stand: 14.07.2012, 15:18 Uhr], Vol. 012, pp. 9–11, 2011.
- [Brad 08] G. Bradski and A. Kaehler. *Learning OpenCV*. O'Reilly Media, September 2008.
- [Brec 05] A. Breckweg. "Assistenzroboter für die Produktion: Stand der Technik - Herausforderungen - Potenziale". In: *Workshop für OTS Systeme in der Robotik*, November 2005.
- [Busi] "Business dressed male figure design". http://artist-3d.com/free_3d_models/dnm/model_disp.php?uid=570 [Stand: 17.07.2012, 16:13 Uhr].
- [Cail 04] F. Caillette and T. Howard. "Real-Time Markerless Human Body Tracking with Multi-View 3-D Voxel Reconstruction". In: *In Proc. BMVC*, pp. 597–606, 2004.
- [Casa 06] J. R. Casas and J. Salvador. "Image-based multi-view scene analysis using 'conexels'". In: *Proceedings of the HCSNet workshop on Use of vision in human-computer interaction - Volume 56*, pp. 19–28, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2006.
- [Dell 00] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun. "Structure from motion without correspondence". In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, Juni 2000.
- [Deut 09] B. I. für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung. "BG/BGIA-Empfehlungen für die Gefährdungsbeurteilung nach Maschinenrichtlinie". http://publikationen.dguv.de/dguv/pdf/10002/bg_bgia_emp_u001d.pdf [Stand: 14.07.2012, 15:19 Uhr], 2009.

- [Dyer 01] C. R. Dyer. “Volumetric Scene Reconstruction from Multiple Views”. In: L. S. Davis, Ed., *Foundations of Image Understanding*, pp. 469–489, Springer US, 2001.
- [Eber 03] D. Ebert. *Bildbasierte Erzeugung kollisionsfreier Transferbewegungen für Industrieroboter*. PhD thesis, Universität Kaiserslautern, Oktober 2003.
- [Fedd 94] J. T. Feddema and J. L. Novak. “Whole arm obstacle avoidance for teleoperated robots”. In: *IEEE International Conference on Robotics and Automation*, pp. 3303 – 3309, San Diego, CA , USA, Mai 1994.
- [Fisc 09a] M. Fischer and D. Henrich. “3D Collision Detection for Industrial Robots and Unknown Obstacles Using Multiple Depth Images”. In: T. Kröger and F. M. Wahl, Eds., *Advances in Robotics Research*, pp. 111–122, Springer Berlin Heidelberg, 2009.
- [Fisc 09b] M. Fischer and D. Henrich. “Surveillance of robots using multiple colour or depth cameras with distributed processing”. In: *Distributed Smart Cameras, 2009. ICD-SC 2009. Third ACM/IEEE International Conference on*, pp. 1 –8, 30 2009-sept. 2 2009.
- [Foro 08] H. Foroughi, A. Rezvanian, and A. Pazirae. “Robust Fall Detection Using Human Shape and Multi-class Support Vector Machine”. *Computer Vision, Graphics & Image Processing, Indian Conference on*, Vol. 0, pp. 413–420, 2008.
- [Fran 09] J.-S. Franco and E. Boyer. “Efficient Polyhedral Modeling from Silhouettes”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, pp. 414–427, 2009.
- [Frau 11] Fraunhofer. “14. IFF-Wissenschaftstage, Fachtagung Sichere Mensch-Roboter-Interaktion”. <http://www.wissenschaftstage.iff.fraunhofer.de/programm/20110630-robotik-mensch-roboter-interaktion.html> [Stand: 23.10.2011, 17:03 Uhr], Juni 2011.
- [Furu 06] Y. Furukawa and J. Ponce. “Carved Visual Hulls for Image-Based Modeling”. In: A. Leonardis, H. Bischof, and A. Pinz, Eds., *Computer Vision - ECCV 2006*, pp. 564–577, Springer Berlin / Heidelberg, 2006.

- [Gand 03] D. Gandhi and E. Cervera. “Sensor covering of a robot arm for collision avoidance”. In: *IEEE International Conference on Systems, Man and Cybernetics*, pp. 4951 – 4955, Oktober 2003.
- [Geck 11] T. Gecks. *Sensorbasierte, echtzeitfähige Online-Bahnplanung für die Mensch-Roboter-Koexistenz*. PhD thesis, Universität Bayreuth, März 2011.
- [Gilb 88] E. Gilbert, D. Johnson, and S. Keerthi. “A fast procedure for computing the distance between complex objects in three-dimensional space”. *Robotics and Automation, IEEE Journal of*, Vol. 4, No. 2, pp. 193 –203, April 1988.
- [Guan 06] L. Guan, S. Sinha, J.-S. Franco, and M. Pollefeys. “Visual Hull Construction in the Presence of Partial Occlusion”. In: *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’06)*, pp. 413–420, IEEE Computer Society, Washington, DC, USA, 2006.
- [Guan 08] L. Guan, J.-S. Franco, and M. Pollefeys. “3D Object Reconstruction with Heterogeneous Sensor Data”. In: *International Symposium on 3D Data Processing, Visualization and Transmission*, Atlanta, États-Unis, 2008.
- [Hane 12] M. Hänel, L. Grüne, D. Henrich, S. Kuhn, and J. Pannek. “Optimal Camera Placement to Measure Distances Regarding Static and Dynamic Obstacles”. *International Journal of Sensor Networks*, Vol. 12, Number 1/2012, pp. 25–36, 2012.
- [Heil 03] P. Heiligensetzer. *Sichere Mensch-Roboter Kooperation durch Fusion haptischer und kapazitiver Sensorik*. PhD thesis, Universität Karlsruhe, 2003.
- [Hein 03] J. Heinzmann and A. Zelinsky. “Quantitative Safety Guarantees for Physical Human-Robot Interaction”. *The International Journal of Robotics Research*, Vol. 22, No. 7-8, pp. 479–504, 2003.
- [Henr 06] D. Henrich and S. Kuhn. “Modeling Intuitive behavior for safe human/robot coexistence cooperation”. In: *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 3929–3934, Orlando, FL, Mai 2006.

- [Henr 08] D. Henrich, M. Fischer, T. Gecks, and S. Kuhn. “Sichere Mensch/Roboter-Koexistenz und Kooperation”. In: *ROBOTIK 2008*, München, Germany, Juni 2008.
- [Hoch 11] “Hochgenaue und echtzeitnahe 3D Rekonstruktion aus Stereobildpaaren und Stereobildfolgen”. <http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=17445064> [Stand: 14.07.2012, 15:23 Uhr], Start: 2006; Ende: 2011.
- [Hofm 11] M. Hofmann, M. Kaiser, H. Aliakbarpour, and G. Rigoll. “Fusion of multi-modal sensors in a voxel occupancy grid for tracking and behaviour analysis”. In: *12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Delft, The Netherlands, April 2011.
- [Hoov 99] A. Hoover and B. D. Olsen. “Path planning for mobile robots using a video camera network”. In: *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*, pp. 890–895, Atlanta, GA, USA, September 1999.
- [Horn 06] A. Hornung and L. Kobbelt. “Hierarchical Volumetric Multi-view Stereo Reconstruction of Manifold Surfaces based on Dual Graph Embedding”. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Vol. 1, pp. 503–510, 2006.
- [HUMO 11] HUMOREV. “Human Modeling, Reconstruction and Recognition in Video”. http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_LANG=EN&PJ_RCN=8372000&pid=77&q=3940AECCF0C385D26FE7974D9AE7C1B0&type=adv [Stand: 14.07.2012, 15:24 Uhr], Start: 2007; Ende: 2011.
- [Izad 11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. “KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera”. In: *24th Symposium on User Interface Software and Technology (UIST)*, Santa Barbara, California, Oktober 2011.

- [Jian 87] B. C. Jiang and C. A. G. Jr. “A cause-and-effect analysis of robot accidents”. *Journal of Occupational Accidents*, Vol. 9, No. 1, pp. 27 – 45, 1987.
- [Karl 00] B. Karlsson, N. Karlsson, and P. Wide. “A dynamic safety system based on sensor fusion”. *Journal of Intelligent Manufacturing*, Vol. 11, pp. 475–483, 2000.
- [Kerp 03] O. Kerpa, K. Weiss, and H. Wörn. “Development of a flexible tactile sensor system for a humanoid robot”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1 – 6, Oktober 2003.
- [Kim 09] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun. “Multi-view Image and ToF Sensor Fusion for Dense 3D Reconstruction”. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 1542–1549, Kyoto, September 2009.
- [Kuhn 06] S. Kuhn, T. Gecks, and D. Henrich. “Velocity control for safe robot guidance based on fused vision and force/torque data”. In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 485 – 492, Heidelberg, Germany, September 2006.
- [Kuhn 07] S. Kuhn and D. Henrich. “Fast vision-based minimum distance determination between known and unknown objects”. In: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2186 – 2191, San Diego, CA, Oktober 2007.
- [Kuhn 09a] S. Kuhn and D. Henrich. “Multi-View Reconstruction of Unknown Objects in the Presence of Known Occlusions”. Tech. Rep., Universität Bayreuth, <http://opus.uni-bayreuth.de/volltexte/2009/576/> [Stand: 14.07.2012, 15:26 Uhr], Juli 2009.
- [Kuhn 09b] S. Kuhn and D. Henrich. “Multi-View Reconstruction of Unknown Objects within a Known Environment”. In: *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I*, pp. 784 – 795, Springer-Verlag, Berlin, Heidelberg, 2009.

- [Kuhn 10] S. Kuhn and D. Henrich. “Multi-View Reconstruction in-between Known Environments”. Tech. Rep., Universität Bayreuth, <http://opus.ub.uni-bayreuth.de/volltexte/2010/661/> [Stand: 14.07.2012, 15:27 Uhr], März 2010.
- [Kutu 00] K. N. Kutulakos and S. M. Seitz. “A Theory of Shape by Space Carving”. *International Journal of Computer Vision*, Vol. 38, pp. 199–218, 2000. 10.1023/A:1008191222954.
- [Ladi 08a] A. Ladikos, S. Benhimane, and N. Navab. “Efficient Visual Hull Computation for Real-Time 3D Reconstruction using CUDA”. In: *Proceedings of the 2008 Conference on Computer Vision and Pattern Recognition Workshops*, Juni 2008.
- [Ladi 08b] A. Ladikos, S. Benhimane, and N. Navab. “Real-Time 3D Reconstruction for Collision Avoidance in Interventional Environments”. In: D. Metaxas, L. Axel, G. Fichtinger, and G. Szekely, Eds., *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2008*, pp. 526–534, Springer Berlin / Heidelberg, 2008.
- [Laze 07] S. Lazebnik, Y. Furukawa, and J. Ponce. “Projective Visual Hulls”. *International Journal of Computer Vision*, Vol. 74, pp. 137–165, 2007. 10.1007/s11263-006-0008-x.
- [Lee 03] J.-H. Lee and H. Hashimoto. “Controlling mobile robots in distributed intelligent sensor network”. In: *Industrial Electronics, IEEE Transactions on*, pp. 890–902, IEEE Industrial Electronics Society, Oktober 2003.
- [Lens 10] T. Lens, J. Kunz, O. von Stryk, C. Trommer, and A. Karguth. “BioRob-Arm: A Quickly Deployable and Intrinsically Safe, Light-Weight Robot Arm for Service Robotics Applications”. In: *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–6, Juni 2010.
- [Lim 00] H.-O. Lim and K. Tanie. “Human Safety Mechanisms of Human-Friendly Robots: Passive Viscoelastic Trunk and Passively Movable Base”. *The International Journal of Robotics Research*, Vol. 19, pp. 307–335, 2000.

- [Lume 93] J. V. Lumelsky. "Real-Time Collision Avoidance in Teleoperated Whole-Sensitive Robot Arm Manipulators". *IEEE Trans. on Syst. Man. Cybern.*, Vol. 23, pp. 194–203, 1993.
- [Matu 01] W. Matusik, C. Buehler, and L. McMillan. "Polyhedral Visual Hulls for Real-Time Rendering". In: *In Proceedings of Twelfth Eurographics Workshop on Rendering*, pp. 115–125, 2001.
- [Mont 04] A. Montenegro, M. Gattass, P. Carvalho, and L. Velho. "Adaptive space carving". In: *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pp. 199 – 206, sept. 2004.
- [Mura 00] A. Murata. "Ergonomics and cognitive engineering for robot-human cooperation". In: *9th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, pp. 206 – 211, Osaka , Japan, September 2000.
- [Open] OpenMP. "OpenMP Specifications". <http://www.openmp.org/mp-documents/OpenMP3.1.pdf> [Stand: 14.07.2012, 15:28 Uhr].
- [Otte 09] H. J. Ottersbach and M. Umbreit. "Arbeitssicherheit bei Arbeitsplätzen mit kollaborierenden Robotern.". May 2009.
- [Pilz] Pilz. "SafetyEye". Webseite <http://www.pilz.de> [Stand 16.09.2011, 16:23 Uhr].
- [Pilz 11] Pilz. *SafetyEye Bedienungsanleitung - Nr. 21 743-DE-11*. http://eshop.pilz.com/downloads/open/PSENse_Operat_Man_21743-DE-11.pdf [Stand 16.09.2011, 16:28 Uhr], 2011.
- [PMDL 12] PMDLumi. "Dynamisches 3D-Sehen Echtzeit-Akquisition bildbasierter 3D Modelle zur Objekterkennung". <http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=22933574> [Stand: 14.07.2012, 15:30 Uhr], Start: 2005; Ende: 2012.
- [REAL 16] REALITYCG. "Computer Graphics of the Real World - Realistic Rendering, Modelling, and Editing of Dynamic, Complex Natural Scenes". http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_LANG=EN&PJ_

- RCN=11675327&pid=78&q=CC45093A8AA29B9A1993664C5ED9A631&type=adv
[Stand: 14.07.2012, 15:17 Uhr], Start: 2011; Ende: 2016.
- [REAL 96] REALISE. “Reconstruction of Reality from Image Sequences”. http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_LANG=EN&PJ_RCN=628672&pid=14&q=CC45093A8AA29B9A1993664C5ED9A631&type=adv [Stand: 14.07.2012, 15:31 Uhr], Start: 1994; Ende 1996.
- [Reko 09] “Rekonstruktion von 3D-Modellen aus zweidimensionalen Laserscans unterschiedlicher Ausrichtung”. <http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=140186972> [Stand: 14.07.2012, 15:31 Uhr], Seit: 2009.
- [RESO 98] RESOLV. “Reconstruction using Scanned Laser and Video”. http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_LANG=EN&PJ_RCN=1427355&pid=16&q=CC45093A8AA29B9A1993664C5ED9A631&type=adv [Stand: 14.07.2012, 15:32 Uhr], Start: 1995; Ende: 1998.
- [Schi 09a] R. Schiavi, A. Bicchi, and F. Flacco. “Integration of active and passive compliance control for safe human-robot coexistence”. In: *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pp. 2471–2475, IEEE Press, Piscataway, NJ, USA, 2009.
- [Schi 09b] A. Schick and R. Stiefelhagen. “Real-Time GPU-Based Voxel Carving with Systematic Occlusion Handling”. In: J. Denzler, G. Notni, and H. Süsse, Eds., *Pattern Recognition*, pp. 372–381, Springer Berlin / Heidelberg, 2009.
- [Seit 06] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. “A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms”. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Vol. 1, pp. 519–528, 2006.
- [Seit 99] S. M. Seitz and C. R. Dyer. “Photorealistic Scene Reconstruction by Voxel Coloring”. *International Journal of Computer Vision*, Vol. 35, pp. 151–173, 1999. 10.1023/A:1008176507526.

- [Sens 12a] SensFloor. “Verbundprojekt: Sensitiver Bodenbelag zur Unterstützung selbständigen Lebens im Alter -Sensfloor-; Teilvorhaben: Sensfloor für AAL”. <http://foerderportal.bund.de/foekat/jsp/SucheAction.do?actionMode=view&fkz=16SV3936> [Stand: 14.07.2012, 15:33 Uhr], Start: 2009; Ende: 2012.
- [sens 12b] sens@home. “Verbundprojekt: Sensorbetreutes Wohnen -sens@home-; Teilvorhaben: Entwicklung eines Multisensorsystems zur Verhaltensanalyse”. <http://foerderportal.bund.de/foekat/jsp/SucheAction.do?actionMode=view&fkz=16SV3907> [Stand: 14.07.2012, 15:34 Uhr], Start: 2009; Ende: 2012.
- [SFB4 05] SFB453. “Gestaltung und Evaluation der Mensch-Roboter-Schnittstelle”. <http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=5442612> [Stand: 14.07.2012, 15:56 Uhr], Seit: 2005.
- [SFB4 10] SFB453. “Wirklichkeitsnahe Telepräsenz und Teleaktion”. <http://www.sfb453.de/> [Stand: 14.07.2012, 15:49 Uhr], Start: 1999; Ende: 2010.
- [SFB5 12] SFB588. “Humanoide Roboter: Lernende und kooperierende multimodale Roboter”. <http://www.sfb588.uni-karlsruhe.de/> [Stand: 14.07.2012, 15:35 Uhr], Start: 2001; Ende: 2012.
- [SFB6] SFB673. “Adaptive alignment in human-robot-cooperation”. <http://www.sfb673.org/projects/C4> [Stand: 14.07.2012, 15:57Uhr].
- [SFB6 06] SFB673. “Ausrichtung in der Kommunikation”. <http://www.sfb673.org/> [Stand: 14.07.2012, 15:55 Uhr], Seit: 2006.
- [SIME 05] SIMERO-2. “Sicherheitsstrategien für die Mensch/Roboter-Kooperation und -Koexistenz”. <http://www.ai3.uni-bayreuth.de/projects/simero/> [Stand: 14.07.2012, 15:37 Uhr], Seit: 2005.
- [Slab 01] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. “A Survey of Methods for Volumetric Scene Reconstruction from Photographs”. In: *International Workshop on Volume Graphics*, pp. 81–100, 2001.

- [Som 05] F. Som. "Sichere Steuerungstechnik für den OTS-Einsatz von Robotern". In: *4. Workshop für OTS-Systeme in der Robotik (Sichere Mensch-Roboter-Interaktion ohne trennende Schutzsysteme)*, November 2005.
- [Spin 02] J. Spingler and S. Thiemermann. "Direkte Mensch-Roboter Kooperation in der flexiblen Montagezelle". In: *Robotik 2002*, pp. 191–195, VDI-Berichte Nr. 1679, 2002.
- [Stau] Stäubli. "CAD-Daten RX130 b". Stäubli, <http://www.staubli.com/de/robotik/kundendienst/cad-daten/> [Stand: 17.10.2011, 15:43 Uhr].
- [Svob 05] T. Svoboda, D. Martinec, and T. Pajdla. "A Convenient Multi-Camera Self-Calibration for Virtual Environments". *PRESENCE: Teleoperators and Virtual Environments*, Vol. 14, No. 4, pp. 407–422, August 2005.
- [Szel 93] R. Szeliski. "Rapid Octree Construction from Image Sequences". *CVGIP: Image Understanding*, Vol. 58, No. 1, pp. 23 – 32, 1993.
- [Thie 02] S. Thiemermann. "team@work - Direkte Mensch-Roboter Kooperation". In: *OTS-Systeme in der Robotik, Reihe BKM Berichte*, pp. 4.1–4.5, Herbert Utz Verlag, München, Germany, 2002.
- [Thie 05] S. Thiemermann. *Direkte Mensch-Roboter-Kooperation in der Kleinteilmontage mit einem SCARA-Roboter*. PhD thesis, Universität Stuttgart, 2005.
- [Towa 08] "Towards a General Vision System for Humanoid Robots". <http://gepris.dfg.de/gepris/OCTOPUS/?module=gepris&task=showDetail&context=projekt&id=94328069> [Stand: 14.07.2012, 15:39 Uhr], Seit: 2008.
- [TRR8 03] TRR8. "Raumkognition - Schliessen, Handeln, Interagieren". <http://www.sfbtr8.spatial-cognition.de/> [Stand: 14.07.2012, 15:41 Uhr], Seit: 2003.
- [TRR8 12] TRR8. "Social Learning for Cognitive Robots". <http://srl.informatik.uni-freiburg.de/> [Stand: 14.07.2012, 16:01 Uhr], Start: 2008; Ende: 2012.

- [Tung 09] T. Tung, S. Nobuhara, and T. Matsuyama. “Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo”. In: *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1709–1716, 29 2009-oct. 2 2009.
- [Viet 08] M. Vieth, O. Zilken, R. Herpers, and D. Reinert. “Vision Based Hand and Finger Detection at Machines with Manual Operation”. In: *ROBOTIK 2008*, München, Germany, 2008.
- [VISI 03] VISIRE. “Virtual Image-processing System for Intelligent Reconstruction of 3D Environments”. http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_LANG=EN&PJ_RCN=5077933&pid=51&q=CC45093A8AA29B9A1993664C5ED9A631&type=adv [Stand: 14.07.2012, 15:41 Uhr], Start: 2000; Ende: 2003.
- [Vitr 11] Vitracom. “Sensorbetreutes Wohnen”. <http://www.vitracom.de/de/f-a-e/senshome.html> [Stand: 14.07.2012, 15:43 Uhr], 2011.
- [Wang 11] C. C. Wang. “Approximate Boolean Operations on Large Polyhedral Solids with Partial Mesh Reconstruction”. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 17, pp. 836–849, 2011.
- [Wikia] Wikipedia. “Industrieroboter”. Wikipedia, <http://de.wikipedia.org/wiki/Industrieroboter> [Stand 23.09.2011, 14:13 Uhr].
- [Wikib] Wikipedia. “STL-Schnittstelle”. Wikipedia, <http://de.wikipedia.org/wiki/STL-Schnittstelle> [Stand 16.10.2011, 15:19 Uhr].
- [Wink 07] B. Winkler. “Safe Space Sharing Human-Robot Cooperation Using a 3D Time-of-Flight Camera”. In: *Proceedings of ISR 2007 International Robots & Vision Show*, 2007.
- [Wink 09] A. Winkler and J. Suchý. “Intuitive Collision Avoidance of Robots Using Charge Generated Virtual Force Fields”. In: T. Kröger and F. M. Wahl, Eds., *Advances in Robotics Research*, pp. 77–87, Springer Berlin Heidelberg, 2009.

- [Worn 09] H. Wörn, J. Graf, K. Safronov, and I. Mamaev. “3D-Umwelterfassung in der Produktion mit neuer 3D-Kamera”. *Allgemeine Vermessungs-Nachrichten (AVN)*, 2009.
- [Yama 96] Y. Yamada, K. Suita, K. Imai, H. Ikeda, and N. Sugimoto. “A failure-to-safety robot system for human-robot coexistence”. *Robotics and Autonomous Systems*, Vol. 18, No. 1-2, pp. 283 – 291, 1996.
- [Zett 02] H. Zettl. “Verwirklichte OTS-Systeme und Gedanken zur Weiterentwicklung”. In: *OTS-Systeme in der Robotik, Reihe BKM Berichte*, pp. 9.1–9.13, Herbert Utz Verlag, München, Juni 2002.
- [Zura 01] J. Zurada, A. L. Wright, and J. H. Graham. “A neuro-fuzzy approach for robot system safety”. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 31, pp. 49 – 64, Februar 2001.

A. Dreiecks-, Kugel- und Voxelmodelle

Objekte in der realen Welt lassen sich auf vielfältige Weise zur Be- und Verarbeitung mit dem Computer geometrisch modellieren. Üblicherweise bringt jede Art der Modellierung verschiedene Vor- und Nachteile mit sich, weswegen eine Konvertierung zwischen solchen Modellen sinnvoll oder gar notwendig sein kann. In diesem Kapitel wird sich auf drei wesentliche solcher Modelle beschränkt: *Dreiecks-*, *Kugel-* und *Voxelmodelle* (Abbildung A.1). Obgleich Objekte meist mit spezieller CAD-Software modelliert werden und in speziellen CAD-Formaten vorliegen, ist diese Software typischerweise auch in der Lage die Modelle in Form von Dreiecksoberflächen mittels der Angabe eines maximalen Fehlers zu exportieren, beispielsweise im STL-Format [Wikib]. Bei hinreichend kleinem Maximalfehler kann man diese Modelle als (nahezu) exakt bezeichnen. Da im Rahmen dieser Arbeit letztendlich Voxelmodelle benötigt werden, stellt sich also die Frage, wie sich solche Dreiecksmodelle in Voxelmodelle konvertieren lassen. Wie in Kapitel 4 eingeführt, sind insbesondere die verkleinerte bzw. vergrößerte Voxelapproximation (Definition 47) von Interesse. Dreiecksmodelle und deren Konvertierung zu den beiden Voxelapproximationen werden in Kapitel A.1 behandelt. Auch wenn eine Optimierung bezüglich der Berechnungszeit im Rahmen dieser Arbeit eine untergeordnete Rolle spielt, so kosten diese Konvertierungen in der aktuellen Implementierung zuviel Zeit, als dass sie in der vorgestellten Anwendung (Kapitel 5) sinnvoll online einsetzbar wären. Um dennoch vertretbare Berechnungszeiten zu erlangen, bieten sich Kugelmodelle für Online-Konvertierungen in Voxelmodelle an. Diese sind – je nach Beschaffenheit – recht zügig in Voxelmodelle konvertierbar. Kugelmodelle und ihre Konvertierung zu vergrößerten Voxelapproximationen sind Thema des Kapitels A.2. Da Kugelmodelle typischerweise erst einmal nicht zur Verfügung stehen, bleibt die Frage, wie man aus den üblichen Dreiecksflächen die Kugelmodelle erzeugen kann. Mit dieser Frage beschäftigt sich Kapitel A.3. Auch die Beschaffenheit der Kugelmodelle, also beispielsweise Anzahl der Kugeln und Überlappungsgrad der einzelnen Kugeln gilt es zu betrachten. Für Hüllenbetrachtungen (vgl. Kapitel 4.3.4) sind vergrößerte Modelle vorausgesetzt.

A. Dreiecks-, Kugel- und Voxelmodelle

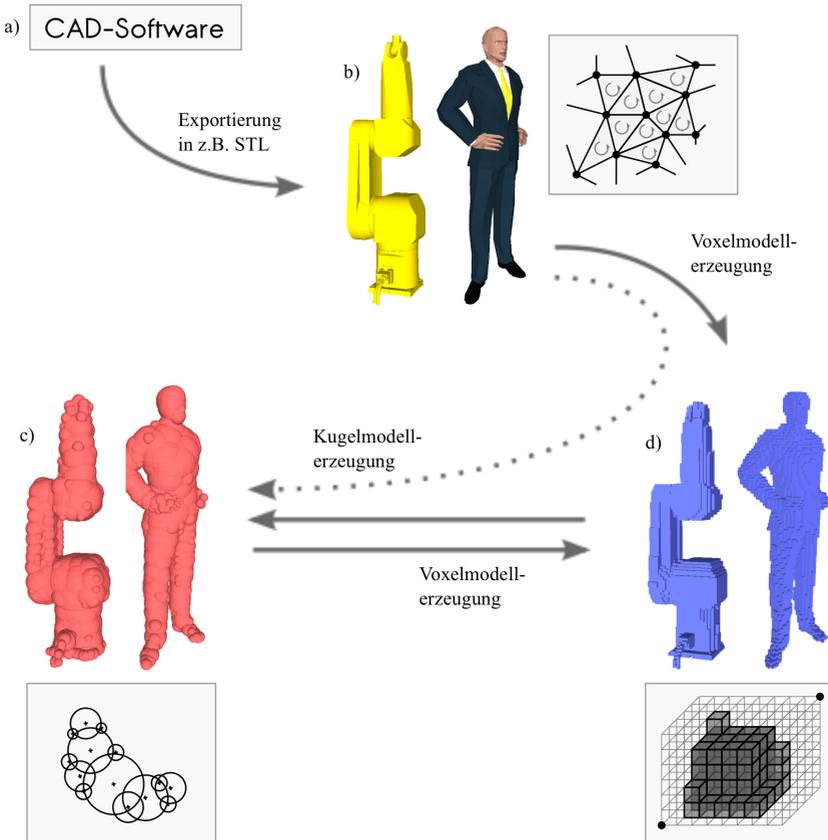


Abbildung A.1.: Illustration der verwendeten geometrischen Darstellungen und Konvertierungen (durchgezogene Linien): Dreieckmodelle (b), Kugelmodelle (c), Voxelmodelle (d). Modelle in CAD-Formaten lassen sich von der jeweiligen CAD-Software (a) als STL-Modell exportieren. Die Konvertierung eines Dreieckmodells in ein Kugelmodell wird indirekt über ein Voxelmodell erreicht (gestrichelte Linie). (Der Abbildung zugrunde liegende Modelldaten: Robotermodell [Stau], Menschmodell [Busi])

Überlegungen und Ansatzmöglichkeiten werden in Kapitel A.4 beschrieben. In Kapitel A.5 wird anhand eines Beispiels gezeigt, wie man ausgehend von CAD-Daten über die Exportierung in ein Dreiecksmodell schließlich mit dem Umweg über geschrumpfte Voxelmodelle zu einem Kugelmodell gelangt, welches über seine vergrößerte Voxelapproximation zu einer verkleinerten Voxelapproximation bzw. Teilmenge dieser gegenüber des ursprünglichen Dreiecksmodell konvertiert wird. Die eingeführten Symbole und Bezeichnungen sind in Anhang B zur Übersicht alphabetisch aufgelistet und können dort nachgeschlagen werden.

A.1. Dreiecksmodelle

Dreiecksmodelle sind einfach durch eine Menge von Dreiecken definiert und diese wiederum jeweils durch drei Eckpunkte im dreidimensionalen Raum:

Definition 71. Ein Tripel $D = (v_1, v_2, v_3)$, mit $v_1, v_2, v_3 \in \mathbb{R}^3$ heißt *Dreieck*. Eine Menge von Dreiecken $\Delta = \{D_1, \dots, D_n\}$, mit $n \in \mathbb{N}^+$ *Dreiecksmodell*.

Unter Berücksichtigung der Reihenfolge der Punkte eines Dreiecks, lässt sich eine Normale mittels des Kreuzproduktes bestimmen:

Definition 72. Sei $D = (v_1, v_2, v_3)$ ein Dreieck, so heißt $n_D = (v_2 - v_1) \times (v_3 - v_1)$ *Normale des Dreiecks*.

Die Fläche eines Dreiecks ist eine Menge von Raumelementen und lässt sich wie folgt definieren:

Definition 73. Sei $D = (v_1, v_2, v_3)$ ein Dreieck, so heißt eine Menge $M_D = \{x \in \mathbb{R}^3 \mid \exists s, r \in \mathbb{R}_0^+ : s + r \leq 1 \wedge x = v_1 + s \cdot (v_2 - v_1) + r \cdot (v_3 - v_1)\}$ die *Fläche eines Dreiecks*.

Das zu einem Punkt entlang eines Strahls am nächstliegenden Dreieck eines Dreiecksmodells wird wie folgt definiert:

Definition 74. Sei $[p_1 p_2]$, mit $p_1, p_2 \in \mathbb{R}^3$ und $p_1 \neq p_2$ ein Strahl und $\Delta = \{D_1, \dots, D_n\}$, mit $n \in \mathbb{N}^+$ ein Dreiecksmodell, so heißt ein Dreieck $S \in \Delta$ *sichtbares Dreieck bezüglich* $[p_1 p_2]$, falls gilt

$$\begin{aligned} & \inf\{\|p_1 - s\|_2 \mid s \in [p_1 p_2] \cap \text{Fläche des Dreiecks } S\} \\ & \leq \inf\{\|p_1 - d\|_2 \mid d \in \bigcup_{D \in \Delta \setminus S} [p_1, p_2] \cap \text{Fläche des Dreiecks } D\} \end{aligned}$$

Gemäß der Definition liegt der Schnittpunkt des Strahls mit dem sichtbaren Dreieck (linker Teil der Formel) also näher oder gleich nah gegenüber den Schnittpunkten des Strahls mit allen im Dreiecksmodell enthaltenen Dreiecken (rechter Teil der Formel) vom Startpunkt des Strahls entfernt. Alle von einem Punkt aus sichtbaren Dreiecke ergeben sich, indem man alle "Richtungen" berücksichtigt:

Definition 75. Seien $p_1 \in \mathbb{R}^3$ ein Punkt und $\Delta = \{D_1, \dots, D_n\}$, mit $n \in \mathbb{N}^+$ ein Dreiecksmodell, so heißt eine Menge $\Theta = \{D \in \Delta \mid \exists p_2 \in \mathbb{R}^3 \text{ mit } p_2 \neq p_1 : D \text{ sichtbares Dreieck bezüglich } [p_1 p_2]\}$ Menge aller sichtbaren Dreiecke bezüglich p_1 .

Mithilfe der Normalen eines Dreiecks lässt sich ermitteln, ob von einem Raumpunkt dessen Vorder- oder Rückseite gesehen wird:

Definition 76. Seien $p \in \mathbb{R}^3$ ein Raumpunkt und $D = (v_1, v_2, v_3)$ ein Dreieck. Sei weiterhin die Normale des Dreiecks n , so ist die Vorderseite von D p zugewandt, falls $(v_1 - p) \cdot n < 0$ und die Rückseite von D p zugewandt, falls $(v_1 - p) \cdot n > 0$.

Mittels des Skalarproduktes der Normalen und der Sichtrichtung lässt sich ermitteln, ob nun Vorder- oder Rückseite einem Raumpunkt zugewandt sind. In dem Fall, dass die Normale des Dreiecks senkrecht auf die Sichtrichtung steht, ist natürlich keine der Seiten dem Raumpunkt zugewandt.

Mit diesen Definitionen lässt sich nun ermitteln, ob ein Dreiecksmodell geschlossen ist, also ein Volumen im Raum korrekt, d.h. widerspruchsfrei eingrenzt:

Definition 77. Sei $\Delta = \{D_1, \dots, D_n\}$, mit $n \in \mathbb{N}^+$ ein Dreiecksmodell, so heißt dieses geschlossen, wenn gilt

$$\nexists x \in \mathbb{R}^3 : \exists D_1, D_2 \in \text{Menge aller sichtbaren Dreiecke bezüglich } x : \\ \text{Vorderseite von } D_1 x \text{ zugewandt} \wedge \text{Rückseite von } D_2 x \text{ zugewandt}$$

Ein Dreiecksmodell ist somit dann geschlossen, wenn von jedem Raumpunkt aus, entweder nur Vorderseiten oder keine Seite der sichtbaren Dreiecke diesem Raumpunkt zugewandt sind oder nur Rückseiten oder keine Seite der sichtbaren Dreiecke diesem zugewandt sind. Ein offenes Dreiecksmodell ist insofern dadurch gekennzeichnet, dass es Raumpunkte gibt, von denen man sowohl Vorderseiten als auch Rückseiten der sichtbaren Dreiecke sehen kann.

Für den Fall, dass ein Dreiecksmodell geschlossen ist, lässt sich ein Inneres, ein Äußeres und ein Rand definieren:

Definition 78. Sei $\Delta = \{D_1, \dots, D_n\}$, mit $n \in \mathbb{N}^+$ ein geschlossenes Dreiecksmodell, so liegt ein Raumpunkt $x \in \mathbb{R}^3$ *innen*, falls

$\exists D \in$ Menge aller sichtbaren Dreiecke bezüglich x : Rückseite von Dx zugewandt;

aussen, falls

$\exists D \in$ Menge aller sichtbaren Dreiecke bezüglich x : Vorderseite von Dx zugewandt;

auf dem Rand, falls x weder innen noch aussen liegt.

Im Folgenden wird immer von geschlossenen Dreiecksmodellen ausgegangen.

Objekte werden oftmals als komplett solide modelliert oder aber auch mit Hohlräumen (Abbildung A.2 a, b), weil die Außenhaut tatsächlich nur aus einem dünnen Blech besteht oder man Gewicht im Inneren einsparen möchte. Im Rahmen dieser Arbeit ist es oft wünschenswert, beide Varianten als komplett solide zu betrachten, da der Hohlraum (beispielsweise innerhalb eines Roboterarmes oder einer Metallstrebe) durch ein gesuchtes Objekt ohnehin nicht zugänglich ist. Für beide Varianten wird im Folgenden betrachtet, ob und wie sich daraus verkleinerte und vergrößerte Voxelapproximationen erzeugen lassen (Definition 47).

A.1.1. Erzeugung einer vergrößerten bzw. verkleinerten Voxelapproximation

Das Volumen eines Voxels liegt entweder komplett im Inneren oder komplett im Äußeren des Dreiecksmodells oder aber es überschneidet sich mit dem Rand des Dreiecksmodells. Voxel, deren Volumen sich mit dem Rand des Dreiecksmodells überschneiden, lassen sich mittels einfacher geometrischer Zusammenhänge über sogenannte Triangle-Voxelization-Algorithmen bestimmen und gehören immer der vergrößerten Voxelapproximation an, jedoch nicht der verkleinerten Voxelapproximation. Das Volumen eines Voxels, dessen Mittelpunkt im Inneren des Dreiecksmodells liegt, kann nicht komplett im Äußeren des Dreiecksmodells liegen. Ebenso wie das Volumen eines Voxels, dessen Mittelpunkt im Äußeren des Dreiecksmodells liegt, nicht komplett im Inneren des Dreiecksmodells liegen kann. Ob nun ein (Voxelmittel-)Punkt im Inneren oder im Äußeren eines Dreiecksmodells liegt, kann mittels obiger Definitionen überprüft werden: Man betrachtet einen (beliebigen) Strahl ausgehend von diesem Ausgangspunkt und

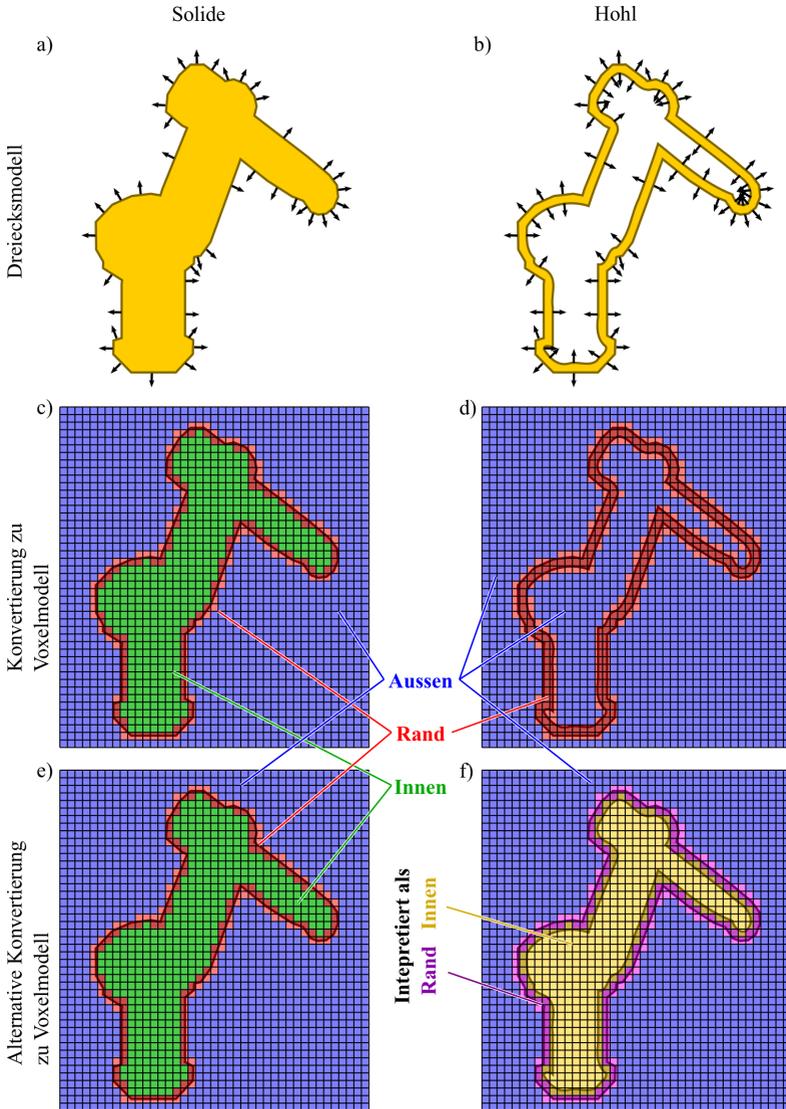


Abbildung A.2.: Darstellung eines solide (a) und eines hohl (b) modellierten Dreiecksmodells inklusive der Dreiecksnormalen sowie der Konvertierung zu einem Voxelmodell (c, d) und einer alternativen Konvertierung (e, f). Oft ist ein solides Voxelmodell erwünscht, trotz eines hohlen Dreiecksmodells.

bestimmt das Dreieck, dessen Schnittpunkt mit dem Strahl am nächsten zum Ausgangspunkt liegt. Ist die Distanz zu dem Dreieck größer Null und das Skalarprodukt von der Normalen des Dreiecks und dem Strahl-definierenden Vektor größer Null, so liegt der Punkt im Inneren des Dreiecksmodells.

Eine vergrößerte Voxelapproximation erhält man nun dadurch, dass man alle Voxel, deren Volumen sich mit dem Rand des Dreiecksmodells überschneidet mit den Voxeln, deren Mittelpunkt im Inneren des Dreiecksmodells liegen vereinigt (Abbildung A.3 a, b, c). Eine ver-

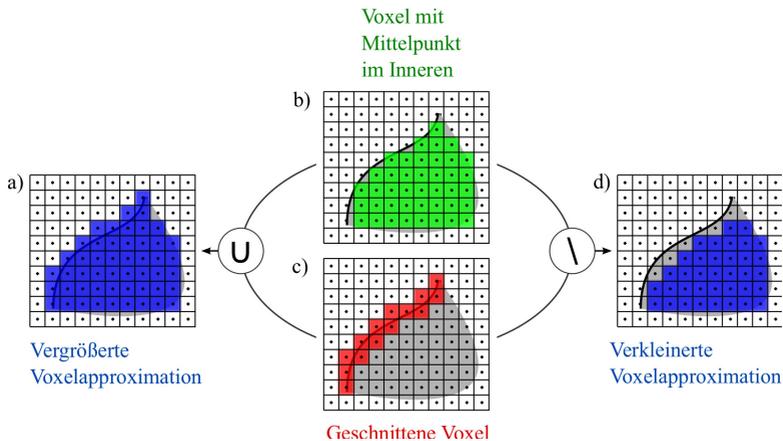


Abbildung A.3: Illustration des Prinzips zur Ermittlung einer vergrößerten Voxelapproximation (a, b, c) und einer verkleinerten Voxelapproximation (c, d, e) basierend auf Voxeln, deren Mittelpunkt im Inneren des Dreiecksmodells liegen und Voxeln, deren Volumen von der Oberfläche des Dreiecksmodells geschnitten werden.

kleinerte Voxelapproximation erhält man, indem man von allen Voxeln, deren Mittelpunkt im Inneren des Dreiecksmodells liegen die Voxel, deren Volumen sich mit dem Rand des Dreiecksmodells überschneiden abzieht (Abbildung A.3 b, c, d).

Besitzt das Dreiecksmodell jedoch Hohlräume wie in Abbildung A.2 b, und sollen diese dennoch dem Voxelmodell angehören, so eignet sich folgender Ansatz (Abbildung A.2 f) mit Einschränkungen, die im späteren Verlauf angesprochen werden: Man erzeugt zunächst eine

Bounding-Box um das Dreiecksmodell, vergrößert diese mindestens um eine Voxelkantenlänge der zugehörigen Dimension (Abbildung A.4 a). Danach nimmt man den Voxel, dessen Volumen eine Ecke dieser vergrößerten Bounding-Box schneidet. Dieser liegt garantiert im Äußeren des Dreiecksmodells. Ausgehend von diesem Voxel wendet man einen Flächennachbarschafts-Floodfill-Algorithmus (vergleichbar zum Clustering-Algorithmus 4.2) an, der alle Voxel innerhalb der vergrößerten Bounding-Box markiert, deren Mittelpunkte außerhalb des Dreiecksmodells liegen (Abbildung A.4 b, c). Im nächsten Schritt berechnet man alle Voxel, deren Volumen sich mit der Oberfläche des Dreiecksmodells überschneiden (Abbildung A.4 d) und zieht diese von der zuvor bestimmten Voxelmenge ab (Abbildung A.4 e). "Invertiert" man das Ergebnis innerhalb der Bounding-Box, so erhält man eine vergrößerte Voxelapproximation des zugrundeliegenden Dreiecksmodells (Abbildung A.4 f), wobei Hohlräume als solide erfasst wurden, jedoch möglicherweise auch zuzüglich Voxeln aus Bereichen, die keine abgeschlossenen Hohlräume darstellen (Abbildung A.5 b). Zieht man den Rand von dieser Voxelapproximation ab, so ist das Resultat nicht immer eine verkleinerte Voxelapproximation (Abbildung A.4 g, Abbildung A.5 c).

A.2. Kugelmodelle

Kugelmodelle sind durch eine Menge von Kugeln definiert und diese wiederum jeweils durch einen Kugelmittelpunkt im dreidimensionalen Raum und einem Skalar, welches den Radius der Kugel beschreibt:

Definition 79. Ein Tupel $K = (c, r)$, mit $c \in \mathbb{R}^3$ und $r \in \mathbb{R}^+$ heißt *Kugel*. Eine Menge von Kugeln $\kappa = \{K_1, \dots, K_n\}$, mit $n \in \mathbb{N}^+$ heißt *Kugelmodell*.

Im Vergleich zu den Dreiecksmodellen ist das Innere, das Äußere und der Rand deutlich einfacher zu beschreiben:

Definition 80. Sei $\kappa = \{K_1, \dots, K_n\}$, mit $n \in \mathbb{N}^+$ ein Kugelmodell, so liegt ein Raumpunkt $x \in \mathbb{R}^3$ *innen*, falls $\exists (c, r) \in \kappa : \|x - c\|_2 < r$; *aussen*, falls $\nexists (c, r) \in \kappa : \|x - c\|_2 \leq r$; *auf dem Rand*, falls x weder innen noch aussen liegt.

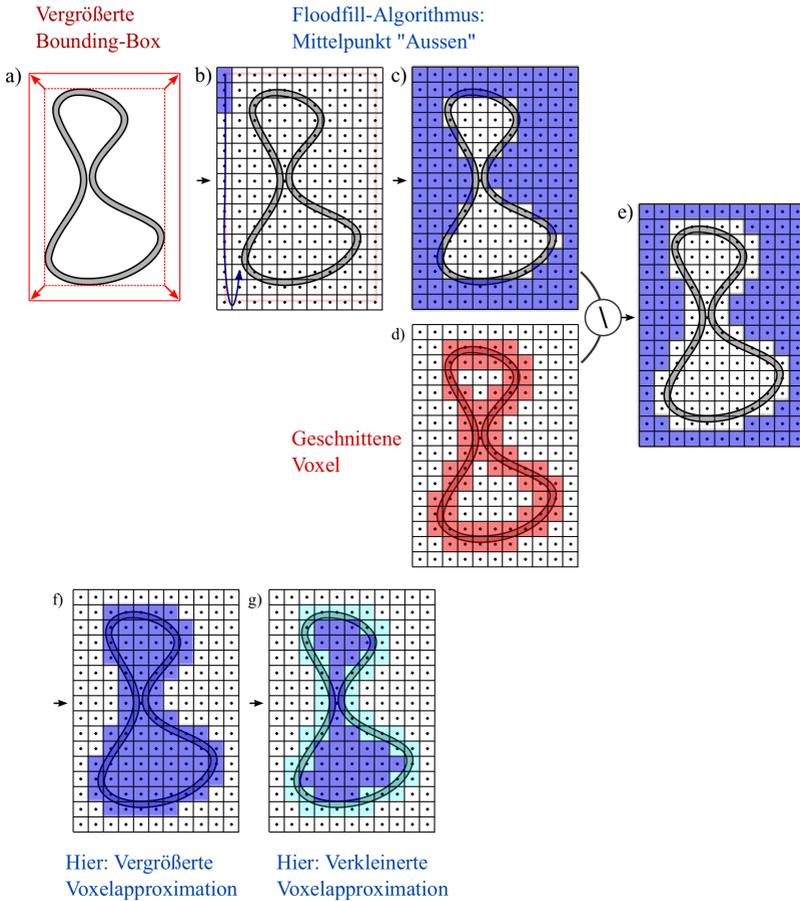


Abbildung A.4.: Illustration eines alternativen Ansatzes zur Ermittlung der vergrößerten und der verkleinerten Voxelapproximation. Hohlräume im Dreiecksmodell erscheinen im Voxelmodell als solide. (Die einzelnen Schritte werden im Text beschrieben.)

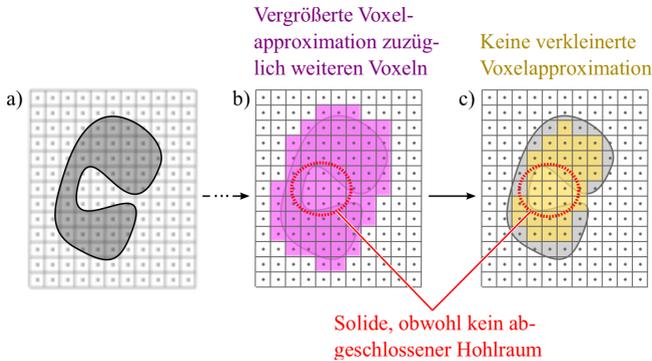


Abbildung A.5.: Illustration der Probleme, welche durch den Ansatz zum zusätzlichen Füllen abgeschlossener Hohlräume entstehen können: Ein Bereich, welcher keinen abgeschlossenen Hohlraum darstellt (a), wird fälschlicherweise als solcher verstanden und als solide im Voxelmodell erzeugt (b), was auch Folgen bei der Erzeugung einer verkleinerten Voxelapproximation hat (c).

A.2.1. Erzeugung einer vergrößerten Voxelapproximation

Alle Voxel deren Volumen sich mit dem Inneren und dem Rand eines Kugelmodells überschneiden gehören der vergrößerten Voxelapproximation an (Abbildung A.6 a, b). In Algorithmus A.1 wird dazu zunächst über jede Kugel iteriert (Zeile 2-9). Da nur solche Voxel überhaupt für das Voxelmodell in Frage kommen, welche auch die Bounding-Box der Kugel schneiden, müssen auch nur diese auf Überschneidung mit dem Kugelmodell überprüft werden (Zeile 4-8). Sofern ein Voxel noch nicht im Voxelmodell enthalten ist, aber sein Volumen sich mit dem der Kugel überschneidet, muss er dem Voxelmodell hinzugefügt werden (Zeile 5-7).

A.2.2. Bemerkung: Verkleinerte Voxelapproximation

Zur Erzeugung einer verkleinerten Voxelapproximation reicht es *nicht* aus, lediglich die Zeile 5 in Algorithmus A.1 in der Art '*sphereIncludes Voxel*' anzupassen, weil nicht das Volumen einer einzigen, sondern aller Kugeln im Kugelmodell zusammen relevant ist (A.6 a, c, d).

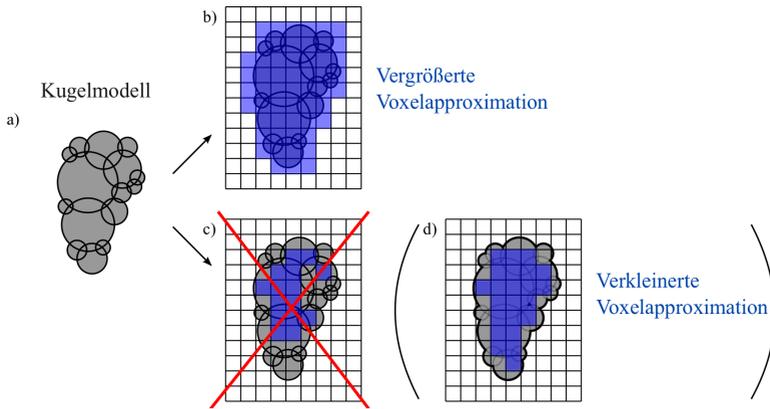


Abbildung A.6.: Illustration der Konvertierung eines Kugelmodells (a) zu einer vergrößerten Voxelapproximation (b) und einer falschen Konvertierung zu einer verkleinerten Voxelapproximation (c) im Vergleich zum korrekten Ergebnis (d).

Algorithmus A.1 Implementierung der Konvertierung eines Kugelmodells K zu einer vergrößerten Voxelapproximation V .

```

1: procedure KUGELZUVOXELMODELL( $K$ )
2:   for all  $k \in K$  do                                     ▷ Prüfe alle Kugeln
3:      $b \leftarrow \text{CREATEBOUNDINGBOX}(k)$                  ▷ Erzeuge Bounding-Box um  $k$ 
4:     for all  $v \in b$  do                                   ▷ Prüfe alle Voxel  $v$  innerhalb Bounding-Box
5:       if  $v \notin V \wedge \text{SPHEREINTERSECTSVOXEL}(v, k)$  then ▷ Falls Voxel nicht in
       Voxelmodell enthalten und sein Volumen Kugel schneidet
6:         ADDVOXEL( $V, v$ )                                  ▷ Füge Voxel dem Voxelmodell hinzu
7:       end if
8:     end for
9:   end for
10:  return  $V$                                              ▷ Liefere Voxelmodell zurück
11: end procedure

```

Algorithmus A.2 Erzeugung eines Kugelmodells K aus einem Voxelmodell V und Parametern $P[]$ basierend auf dem Greedy-Prinzip: Es werden solange beste Kugeln hinzugefügt, bis das Volumen des Voxelmodells komplett im Volumen des Kugelmodells enthalten ist.

```

1: procedure VOXELZUKUGELMODELL( $V, P[]$ )
2:    $K \leftarrow \text{CREATEEMPTYSPHEREMODEL}()$  ▷ Erzeuge leeres Kugelmodell
3:   while  $\neg \text{CONTAINSVOXELMODEL}(K, V)$  do ▷ Solange Voxelmodell  $\not\subseteq$  Kugelmodell
4:      $b \leftarrow \text{FINDBESTSPHERE}(V, K, P[])$  ▷ Finde beste Kugel
5:      $\text{ADDSPHERE}(K, b)$  ▷ Füge beste Kugel dem Kugelmodell hinzu
6:   end while
7:   return  $K$  ▷ Liefere Kugelmodell zurück
8: end procedure

```

A.3. Erzeugung von Kugelmodellen

In diesem Kapitel wird ein Greedy-Algorithmus beschrieben, der eine Konvertierung von einem durch seine belegten Voxel repräsentierten Objekt in eine Kugeldarstellung unter Angabe eines maximalen positiven Fehlers erlaubt. Die Erzeugung eines Kugelmodells aus einem Dreiecksmodell lässt sich über den Umweg eines Voxelmodells erreichen. Neben der Angabe des maximalen positiven Fehlers, lässt sich die Anzahl der Kugeln, deren Überlappung sowie deren zusätzlich eingenommener Raum gegenüber dem Ursprungsmodell mittels Parametern einer Bewertungsfunktion indirekt steuern. Je nach Anwendung kann eine möglichst geringe Anzahl von Kugeln oder eine möglichst geringe Überlappung dieser von Vorteil sein.

Im Folgenden wird zunächst der Greedy-Algorithmus vorgestellt (Kapitel A.3.1). Anschließend wird ein Experimentaufbau beschrieben und dessen Ergebnisse diskutiert (Kapitel A.3.2). Das letzte Kapitel beschreibt Schlussfolgerungen und Möglichkeiten einer Erweiterung des Algorithmus (Kapitel A.3.3). Das Hauptaugenmerk liegt dabei immer auf dem generierten Kugelmodell, nicht aber in der Ausführungszeit des Algorithmus.

A.3.1. Algorithmus

Gemäß des Greedy-Prinzips wird solange schrittweise die beste Kugel ermittelt und dem Kugelmodell hinzugefügt, bis das gesamte Voxelmodell im Kugelmodell vollständig enthalten ist (Algorithmus A.2). Es gilt daher im Folgenden zu beschreiben, was eine gültige Kugel ist und wie sie sich bewerten lässt. Dem entsprechend arbeitet auch die Funktion *FindBestSphere* in Algorithmus A.2.

Voxeldarstellung gegeben. Die *Kugelmodellzugehörigkeit* beschreibt, ob ein Voxel bereits durch das aktuelle, unfertige Kugelmodell im iterativen Greedy-Verfahren enthalten ist oder noch hinzugefügt werden muss. Anhand dieser beiden Merkmale lassen sich vier unterschiedliche Ausprägungen $S_1 - S_4$ eines Voxels identifizieren (Abbildung A.7, b), denen zur Bewertung einer Kugel jeweils ein Gewicht zugeordnet wird (Tabelle A.1). Eine potentiell neue Kugel lässt

		Kugelmodellzugehörigkeit	
		ja	nein
Objektzugehörigkeit	ja	S_1	S_2
	nein	S_3	S_4

Tabelle A.1.: Voxelausprägungen S_1 - S_4 basierend auf den beiden Merkmalen *Objektzugehörigkeit* und *Kugelmodellzugehörigkeit*

sich nun anhand der enthaltenen Voxel bewerten, indem die Gewichtung der Voxelausprägung eines jeden enthaltenen Voxels summiert wird. Die Kugel mit der höchsten Bewertung gilt als die beste und wird dem Kugelmodell hinzugefügt. Zu bemerken sei, dass mindestens eine maximale Abweichung $E \geq \frac{1}{2} \sqrt{v_x^2 + v_y^2 + v_z^2}$ zugelassen werden muss, da ansonsten für einen einzelnen Voxel keine gültige Kugel gefunden werden kann, die diesen enthält. Dabei sind v_x , v_y und v_z die Längen der Voxelkanten in den drei Raum-Dimensionen.

A.3.2. Experimente

In diesem Kapitel wird untersucht aus wie vielen Kugeln zwei Beispielmodelle tatsächlich bestehen und wie hoch der Überlappungsfaktor dieser Kugeln bezüglich unterschiedlicher Voxelraumaufösungen, unterschiedlicher maximaler positiver Fehler und unterschiedlicher Bewertungskriterien ist. Da jede Kugel eine Menge an Voxeln vollständig enthält, ergibt sich der Überlappungsfaktor einfach aus dem Quotienten der Summe aller Voxel in allen Kugeln und der Anzahl unterschiedlicher Voxel, die im Eingabeobjekt vorhanden sind. Durch überlagernde Kugeln fließen daher Voxel mehrfach in die Summe ein.

A.3.2.1. Aufbau

Als Eingabeobjekte stehen ein Menschmodell O_1 [Busi] und ein Modell des Roboters RX130b O_2 [Stau] als triangulierte Oberflächennetze zur Verfügung. Diese Objekte werden gemäß Kapitel A.1 in unterschiedlich feine Voxelaufösungen transformiert, indem das Objekt in der Dimension seiner größten Ausdehnung gleichmäßig in 50, 100 und 200 Abschnitte unterteilt wird und in den anderen beiden Dimensionen angepasst wird, so dass ein Voxel nahezu würfelförmig ist (Abbildung A.8). Die resultierende Anzahl an Voxeln die in der jeweiligen Auflösung dem Modell angehören sind in Tabelle A.2 aufgelistet. Die Auswirkung unterschiedlicher maximaler

		Voxelraumauflösung		
		V_{50}	V_{100}	V_{200}
Eingabeobjekt	O_1	2004 Voxel (37.52 mm × 36.59 mm × 36.77 mm)	16851 Voxel (18.30 mm × 18.29 mm × 18.20 mm)	137390 Voxel (9.04 mm × 9.15 mm × 9.05 mm)
	O_2	2393 Voxel (39.02 mm × 40.16 mm × 38.44 mm)	20085 Voxel (19.51 mm × 19.39 mm × 19.22 mm)	164234 Voxel (9.75 mm × 9.69 mm × 9.61 mm)

Tabelle A.2.: Anzahl und Dimension der im Experiment verwendeten Voxel

positiver Fehler fließt ebenfalls in die Betrachtung mit ein. Es werden daher vier unterschiedliche maximale positive Fehler $E_{0,5}$, E_1 , E_2 und E_4 in Bezug auf die Voxelraumauflösung gewählt. Der Index beschreibt das Vielfache der jeweiligen Voxel diagonalen, das als maximale positive Abweichung in den Algorithmus eingeht (Tabelle A.3). Es werden sechs unterschiedliche Bewertungskriterien im Experiment betrachtet. Das erste Kriterium $K_{1,0}$ belohnt lediglich Voxel, welche zum Eingabeobjekt gehören, aber noch nicht im Kugelmodell enthalten sind (S_3). Das letzte Kriterium $K_{0,-1}$ bestraft lediglich Voxel, welche bereits im Kugelmodell vorhanden sind (S_1 , S_4). Die Kriterien $K_{2,-1}$, $K_{1,-1}$, $K_{1,-2}$ und $K_{1,-1000}$ sind Abstufungen zwischen den beiden Extremen $K_{1,0}$ und $K_{0,-1}$ (Tabelle A.4). Da der implementierte Algorithmus intern mit ganzen Zahlen arbeitet, wurden die Kriterien hier ebenfalls in ganzen Zahlen angegeben. Es wäre ebenso möglich die Zahlen direkt im Verhältnis zueinander anzugeben.

A. Dreiecks-, Kugel- und Voxelmodelle

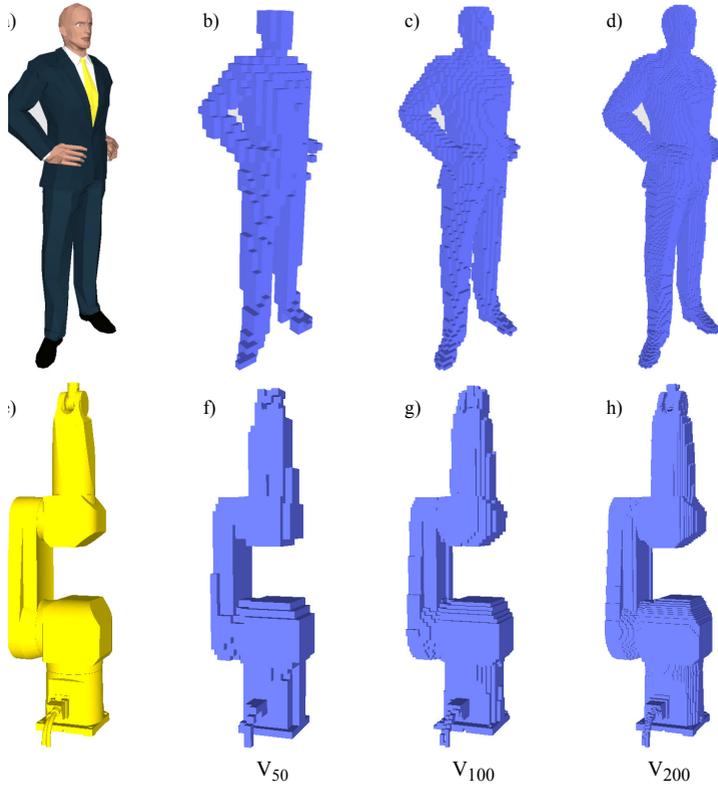


Abbildung A.8: Eingabeobjekte O_1 (a-d) und O_2 (e-h) in ihren Voxelraumauflösungen V_{50} (b, f), V_{100} (c, g) und V_{200} (d, h).

		Maximaler positiver Fehler			
		$E_{0,5}$	E_1	E_2	E_4
Eingabeobjekt O_1	V_{50}	32.33	64.02	128.04	256.09
	V_{100}	15.98	31.64	63.28	126.55
	V_{200}	7.94	15.73	31.46	62.92
O_2	V_{50}	34.30	67.92	135.83	271.67
	V_{100}	16.94	33.55	67.11	134.22
	V_{200}	8.47	16.78	33.55	67.11

Tabelle A.3.: Betrachtete maximale positive Fehler E_* bezüglich des Eingabeobjektes und der Voxelraumauflösung.

		Bewertungskriterium					
		$K_{1,0}$	$K_{2,-1}$	$K_{1,-1}$	$K_{1,-2}$	$K_{1,-1000}$	$K_{0,-1}$
Gewichtung der Voxelausprägung	S_1	0	-1	-1	-2	-1000	-1
	S_2	0	0	0	0	0	0
	S_3	1	2	1	1	1	0
	S_4	0	-1	-1	-2	-1000	-1

Tabelle A.4.: Betrachtete Bewertungskriterien $K_{*,*}$ mit ihren jeweiligen Gewichtungen der vier Voxelausprägungen S_1 - S_4 .

A.3.2.2. Evaluation

Die Anzahl von resultierenden Kugeln und der Überlappungsfaktor wurden für alle Kombinationen der veränderlichen Eingaben berechnet und sind in den Tabellen A.5 und A.6 nachlesbar.

Menschmodell (O1)		Voxelraumauflösung						
		Ergebniswerte						
		V_{50}		V_{100}		V_{200}		
		Kugeln	Faktor	Kugeln	Faktor	Kugeln	Faktor	
Maximaler positiver Fehler Bewertungskriterium	$E_{0.5}$	$K_{1,0}$	514	3.14	1934	7.28	7324	15.19
		$K_{2,-1}$	681	1.21	3220	1.26	13902	1.28
		$K_{1,-1}$	805	1.11	3908	1.14	17800	1.14
		$K_{1,-2}$	876	1.05	4435	1.08	21159	1.10
		$K_{1,-1000}$	936	1.04	5200	1.07	26751	1.11
		$K_{0,-1}$	2004	1	16851	1	137390	1
	E_1	$K_{1,0}$	86	2.47	303	4.53	1050	7.87
		$K_{2,-1}$	160	1.38	918	1.37	4559	1.33
		$K_{1,-1}$	211	1.30	1300	1.24	7146	1.19
		$K_{1,-2}$	258	1.27	1640	1.18	9954	1.14
		$K_{1,-1000}$	306	1.24	2292	1.16	14477	1.14
		$K_{0,-1}$	2004	1	16851	1	137390	1
	E_2	$K_{1,0}$	26	3.36	60	3.47	205	4.53
		$K_{2,-1}$	65	2.19	330	1.59	1879	1.42
		$K_{1,-1}$	95	2.13	579	1.51	3469	1.31
		$K_{1,-2}$	117	2.06	818	1.48	5209	1.27
		$K_{1,-1000}$	145	1.96	1170	1.45	9006	1.25
		$K_{0,-1}$	2004	1	16851	1	137390	1
	E_4	$K_{1,0}$	9	7.13	26	5.59	54	4.15
		$K_{2,-1}$	28	4.07	151	2.46	907	1.74
		$K_{1,-1}$	50	3.94	254	2.40	1794	1.66
		$K_{1,-2}$	61	3.92	420	2.33	2886	1.62
		$K_{1,-1000}$	76	3.79	776	2.23	5372	1.60
		$K_{0,-1}$	2004	1	16851	1	137390	1

Tabelle A.5.: Auswertung der Kugelanzahl und des Überlappungsfaktors bezogen auf verschiedene Bewertungskriterien am Eingabeobjekt "Menschmodell".

Jede Kugel des erzeugten Kugelmodells enthält eine Menge von Voxeln vollständig. Die Anzahl dieser Voxelmenge einer jeden Kugel aufsummiert, geteilt durch die Anzahl der Voxel des Eingabemodells beschreibt dabei den *Überlappungsfaktor*. In Abbildung A.9 sind die generierten

Robotermodell (O ₂)		Voxelraumauflösung						
		Ergebniswerte						
		V ₅₀		V ₁₀₀		V ₂₀₀		
		Kugeln	Faktor	Kugeln	Faktor	Kugeln	Faktor	
Maximaler positiver Fehler Bewertungskriterium	E _{0.5}	K _{1,0}	392	3.89	1310	6.80	4312	13.66
		K _{2,-1}	601	1.21	2736	1.25	11506	1.27
		K _{1,-1}	714	1.11	3360	1.13	15301	1.13
		K _{1,-2}	779	1.07	3982	1.08	19070	1.07
		K _{1,-1000}	869	1.05	4810	1.06	25366	1.05
		K _{0,-1}	2393	1	20085	1	164234	1
	E ₁	K _{1,0}	69	2.93	267	4.77	787	6.69
		K _{2,-1}	147	1.32	855	1.30	3760	1.31
		K _{1,-1}	205	1.23	1230	1.21	6102	1.18
		K _{1,-2}	241	1.20	1676	1.15	8995	1.12
		K _{1,-1000}	262	1.18	2234	1.14	14963	1.08
		K _{0,-1}	2393	1	20085	1	164234	1
	E ₂	K _{1,0}	14	2.55	63	4.43	214	4.92
		K _{2,-1}	42	1.83	310	1.47	1826	1.39
		K _{1,-1}	52	1.79	526	1.39	3444	1.27
		K _{1,-2}	106	1.70	737	1.34	5652	1.21
		K _{1,-1000}	112	1.71	1270	1.30	10705	1.17
		K _{0,-1}	2393	1	20085	1	164234	1
	E ₄	K _{1,0}	6	4.69	15	3.24	60	4.62
		K _{2,-1}	21	2.93	136	1.90	1020	1.56
		K _{1,-1}	23	2.92	231	1.82	1998	1.48
		K _{1,-2}	49	2.93	349	1.78	3532	1.41
		K _{1,-1000}	57	2.93	709	1.73	6893	1.36
		K _{0,-1}	2393	1	20085	1	164234	1

Tabelle A.6.: Auswertung der Kugelanzahl und des Überlappungsfaktors bezogen auf verschiedene Bewertungskriterien am Eingabeobjekt "Robotermodell".

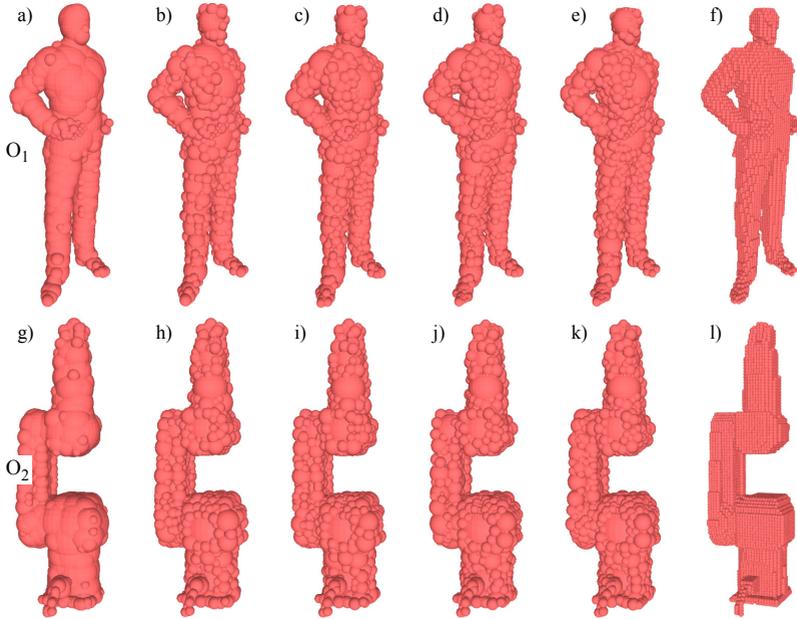


Abbildung A.9.: Darstellung der Ergebnisse beider Eingabeobjekte O_1 (a-f) und O_2 (g-l) und der Voxelraumauflösung V_{100} , dem maximalen positiven Fehler E_1 und den Bewertungskriterien $K_{1,0}$ (a, g), $K_{2,-1}$ (b, h), $K_{1,-1}$ (c, i), $K_{1,-2}$ (d, j), $K_{1,-1000}$ (e, k) und $K_{0,-1}$ (f, l).

Kugelmodelle für die konkrete Voxelaufauflösung V_{100} in Verbindung mit einem maximalen positiven Fehler E_1 für Mensch- und Robotermodell dargestellt. Die Diagramme in den Abbildungen A.10 und A.11 veranschaulichen die Kugelanzahl und den Überlappungsfaktor in Abhängigkeit zur gewählten Bewertungsparametrisierung.

Es lässt sich leicht nachvollziehen, dass je stärker die Überlappung (S_4) bestraft wird desto mehr Kugeln werden generiert. Weiterhin ist erkennbar, dass wenn nur Überlappungen negativ bewertet werden, die Kugelanzahl gleich der Anzahl an Voxeln des Eingabeobjektes ist, was sich durch den Algorithmus selbst und die Implementierung erklären lässt: Die am besten bewertete Kugel wird niemals einen höheren Wert als *null* haben - egal ob der Radius der Kugel groß oder klein ist. Da die Implementierung die erstbeste Kugel aus einer Menge von Kugeln mit einer gleichen Bewertung nimmt und die erstbeste Kugel dort immer den Radius eines einzigen Voxels besitzt, besteht letztendlich das gesamte Kugelmodell aus Kugeln die jeweils einen Voxel komplett umfassen.

Der Überlappungsfaktor hingegen fällt, je stärker eine Überlappung (S_4) negativ bewertet wird zunächst stark, bleibt danach aber auf etwa gleichem Niveau (Abbildung A.10 und A.11). Erst wenn ausschließlich eine Überlappung negativ bewertet wird und sämtliche Kugeln jeweils nur einen einzigen Voxel umfassen, sinkt der Überlappungsfaktor in allen Fällen auf den Wert *eins*. Der Grund für das relativ hohe Niveau im mittleren Bereich des Diagramms ($K_{(2,-1)} \dots K_{(1,-1000)}$) lässt sich damit erklären, dass der durch das Kugelmodell zusätzlich belegte Raum im Experiment nicht negativ sondern neutral bewertet wird. Es entstehen viele Kugeln, die den maximalen positiven Fehler ausnutzen können um eine bessere Bewertung zu erlangen. Würde der zusätzlich belegte Raum negativ bewertet werden, hätte dies zur Folge, dass der Überlappungsfaktor sinken würde.

Aus den beiden Tabellen A.5 und A.6 kann man ablesen, dass bei gleichem maximalen positiven Fehler aber höherer Voxelaufauflösung für das Kriterium $K_{1,0}$ die Anzahl der Kugeln fällt (dazu bewege man sich in den beiden Tabellen jeweils ausgehend von einer Zelle süd-östlich). Diese Gegebenheit lässt sich durch die größere Auswahl möglicher Kugelmittelpunkte beim Ermitteln der besten Kugel im Greedy-Algorithmus erklären. Kugeln können dadurch "geschickter" platziert werden, sodass letztendlich weniger Kugeln benötigt werden. Es lässt sich außerdem feststellen, dass wenn die Voxelaufauflösung und die Bewertungsparametrisierung konstant gehalten werden bei Vergrößerung des maximalen positiven Fehlers die Anzahl der Kugeln grundsätzlich sinkt.

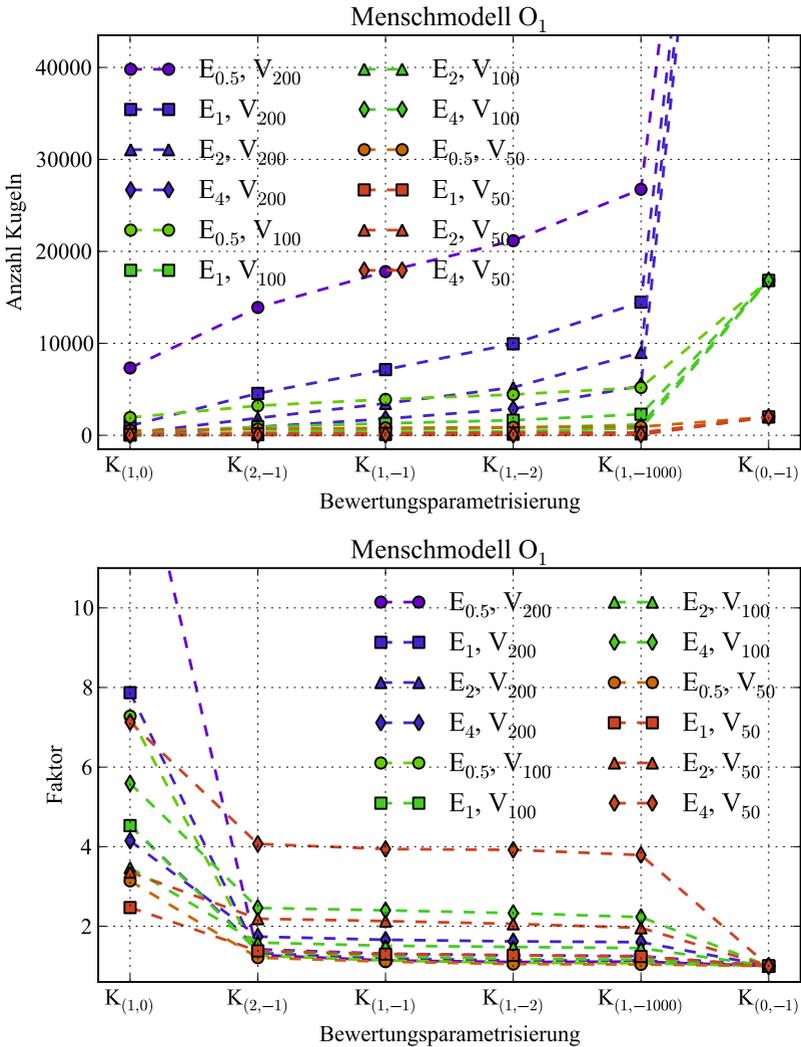


Abbildung A.10.: Darstellung der Kugelanzahl und des Überlappungsfaktors bezüglich der Bewertungsparametrisierung für unterschiedliche Voxelauflösungen und maximale positive Fehler am Eingabeobjekt "Menschmodell".

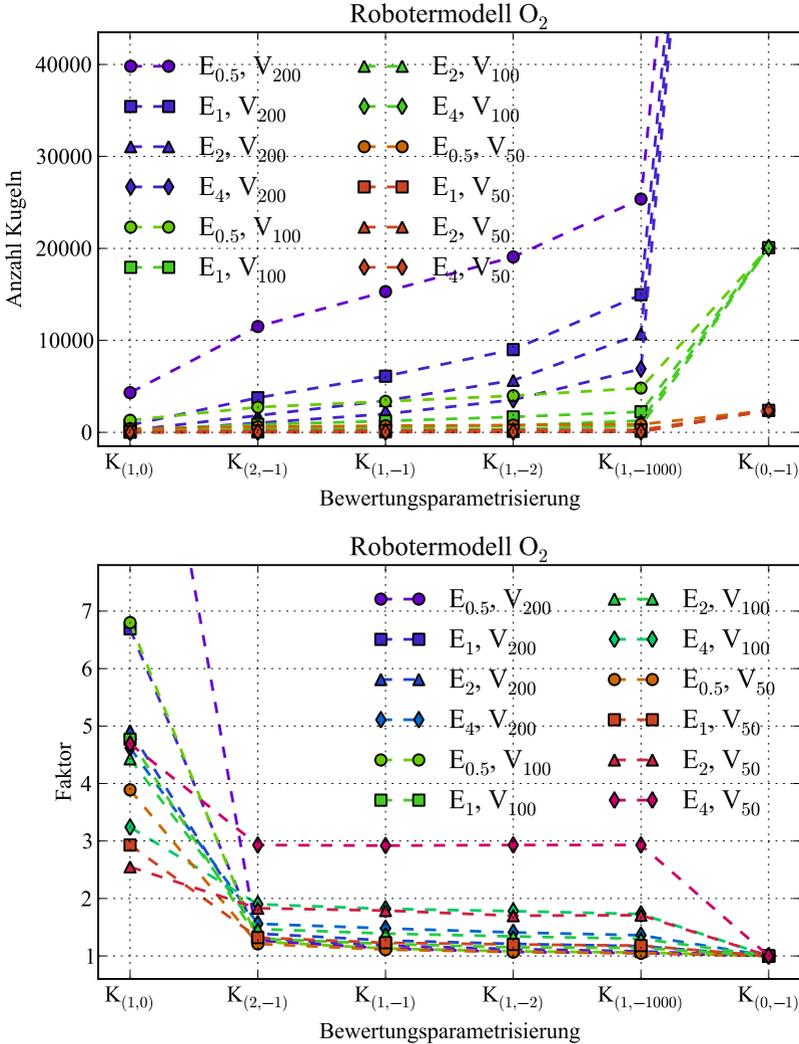


Abbildung A.11.: Darstellung der Kugelanzahl und des Überlappungsfaktors bezüglich der Bewertungsparametrisierung für unterschiedliche Voxelauflösungen und maximale positive Fehler am Eingabeobjekt "Robotermodell".

Als Randbemerkung sei hinzugefügt, dass die Berechnungsdauer zur Erzeugung der Kugelmodelle bei der mäßig optimierten Implementierung auf einer Intel Quad-Core CPU mit 2.66 GHz zwischen knapp einer Minute und etwas über 24 Stunden lag. Die Berechnung dauerte insbesondere bei hoher Voxelaufösung, hoher Bestrafung der Überlappung von Kugeln und großem maximalen positiven Fehler lange. Ein höherer Grad der Parallelisierung im Rahmen der Bewertung möglicher Kugeln innerhalb eines Schrittes im Greedy-Algorithmus birgt einen großen Zugewinn an Schnelligkeit.

A.3.3. Schlussfolgerungen und Ausblick

Es wurde ein Algorithmus präsentiert, mit dem sich aus Voxelmodellen Kugelmodelle erzeugen lassen, die einen maximalen positiven Fehler einhalten. Mittels der Gewichtung von vier Voxelausprägungen kann die Qualität des Kugelmodells beeinflusst werden, sodass entweder weniger Kugeln benötigt werden, oder die verwendeten Kugeln sich wenig überlappen. Diese Eigenschaften können sich in Algorithmen, welche auf Kugelmodellen arbeiten positiv bezüglich des zeitlichen Aufwandes auswirken.

Es ist denkbar durch eine zusätzliche Voxelausprägung *Oberflächenzugehörigkeit* und dem Abbruchkriterium *Alle Oberflächenvoxel im Modell enthalten* ein Kugelmodell zu generieren, welches innen „hohl“ ist. Eine solche Art von Kugelmodell kann beispielsweise von Vorteil sein, wenn ein künstliches Tiefenbild davon erzeugt werden soll, da hier der Innenraum nicht benötigt wird.

A.4. Expansion und Kontraktion von Voxelmodellen

Beim Expandieren und Kontrahieren geht es um die Erzeugung eines vergrößerten bzw. eines verkleinerten Modells. Zunächst wird definiert, was Expansion und Kontraktion formal bedeuten:

Definition 81. Sei $A \subset \mathbb{R}^n$. Eine Menge $E_r(A) = \{x \in \mathbb{R}^n \mid \min\{\|x - a\|_2 \mid a \in A\} \leq r\}$ heißt *Expansion* und eine Menge $K_r(A) = \{a \in A \mid \min\{\|a - x\|_2 \mid x \in \mathbb{R}^n \setminus A\} > r\}$ heißt *Kontraktion* der Menge A mit dem Radius r .

Da Voxel eine Diskretisierung darstellen – sie besitzen ein quaderförmiges Volumen, kann das Volumen einer Voxelmenge weder eine Expansion noch eine Kontraktion des Volumens

einer anderen Voxelmenge im Allgemeinen darstellen (Abbildung A.12). Vielmehr wird sich eine

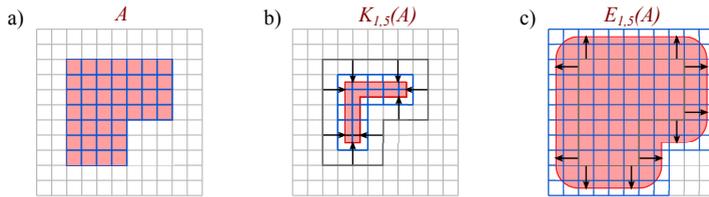


Abbildung A.12.: Illustration a) einer Menge A (rot), b) deren Kontraktion $K_{1,5}(A)$ (rot) und c) deren Expansion $E_{1,5}(A)$ (rot) sowie der Tatsache, dass Voxelmodelle (hellblau) im Allgemeinen keine (exakte) Kontraktion oder Expansion sein können (b, c).

Vergrößerung oder Verkleinerung von Voxelmodellen in einem Intervall abspielen.

Definition 82. Seien $A, B \subset \mathbb{R}^n$. Das Intervall

$$[\max\{r \in \mathbb{R}^+ \mid E_r(A) \subset B\}, \min\{r \in \mathbb{R}^+ \mid E_r(A) \supset B\}]$$

heißt *Expansions-Intervall* der Menge B um A . Das Intervall

$$[\max\{r \in \mathbb{R}^+ \mid K_r(A) \subset B\}, \min\{r \in \mathbb{R}^+ \mid K_r(A) \supset B\}]$$

A. Dreiecks-, Kugel- und Voxelm Modelle

heißt *Kontraktions-Intervall* der Menge B um A .

Betrachtet man konkret Voxelm Modelle, so ist die Diagonale die größte Distanz innerhalb des Volumens eines einzelnen Voxels. Da entweder ein Voxel einer Voxelm Menge angehört, oder eben nicht, muss die Länge eines Expansions- und Kontraktions-Intervalles genau der Länge dieser Strecke entsprechen. (Abbildung A.13).

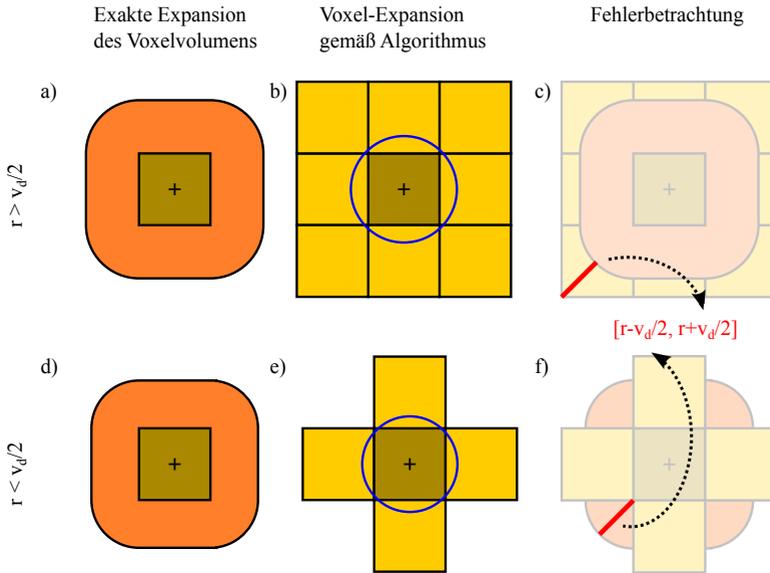


Abbildung A.13.: Darstellung des resultierenden Expansions-Intervalls bei Voxelm Modellen. Ist der Expansionsradius etwas größer als die halbe Länge der Voxeldiagonalen v_d (a, b, c), so resultiert ein Voxelmodell, welches maximal um die halbe Länge der Voxeldiagonalen größer als die exakte Expansion ist. Ist der Expansionsradius hingegen etwas kleiner als die halbe Länge der Voxeldiagonalen (d, e, f), so resultiert ein Voxelmodell, welches maximal um die halbe Länge der Voxeldiagonalen kleiner als die exakte Expansion ist.

Zum Expandieren eines Voxelmodells (Abbildung A.14 a) legt man in jeden enthaltenen Voxel eine Kugel mit dem gewünschten Expansions-Radius r (Abbildung A.14 b). Danach

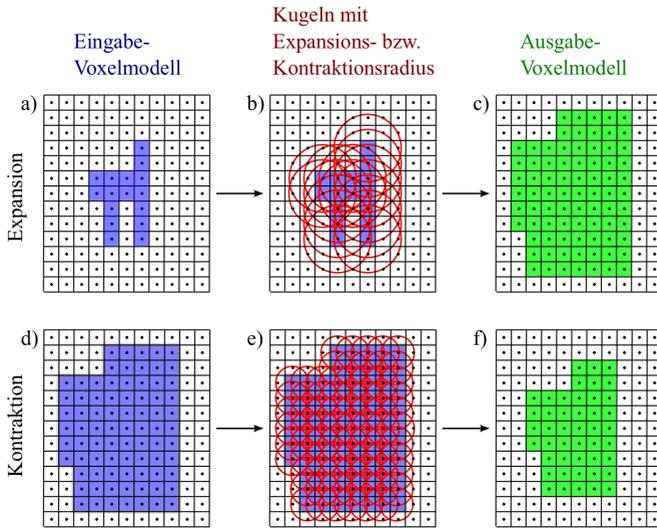


Abbildung A.14.: Darstellung der Expansion (a, b, c) und der Kontraktion (d, e, f) eines Voxelmodells.

erzeugt man ein Voxelmodell aus allen Voxeln, deren Volumen eine dieser Kugeln schneiden (Abbildung A.14 c). Das Ergebnis ist ein Voxelmodell, dessen Volumen im Expansions-Intervall von $[r - \frac{v_d}{2}, r + \frac{v_d}{2}]$ liegt, wobei v_d die Länge der Voxeldiagonalen ist (Abbildung A.13).

Zum Kontrahieren eines Voxelmodells (Abbildung A.14 d) legt man ebenfalls in jeden enthaltenen Voxel eine Kugel mit dem gewünschten Kontraktions-Radius r (Abbildung A.14 e). Danach erzeugt man ein Voxelmodell aus allen Voxeln, deren zugehörige Kugel sich ausschließlich mit dem Volumen des Voxelmodells überschneiden (Abbildung A.14 f). Das Ergebnis ist ein Voxelmodell, dessen Volumen im Kontraktions-Intervall von $[r - \frac{v_d}{2}, r + \frac{v_d}{2}]$ liegt.

A.5. Beispiel: Schnelle verkleinerte Voxelapproximation

Angenommen man hat einen Roboter in Form von CAD-Daten und möchte dessen Volumen als ausschließbaren Raumbereich im Rahmen der wissens- und sensorbasierten geometrischen Rekonstruktion nutzen. Wie bekannt ist, lassen sich CAD-Daten üblicherweise problemlos in das STL-Format exportieren, sodass ein (nahezu) exaktes Dreiecksmodell als Basis zur Verfügung steht. Da der Roboter sich bewegen kann, ist eine einmalige Konvertierung in ein Voxelmodell nicht ausreichend – er muss zu jedem Zeitpunkt in welchem eine Rekonstruktion berechnet wird in ein Voxelmodell umgewandelt werden. Der Algorithmus zur Bestimmung eines Voxelmodells auf Basis des Dreiecksmodells ist zu langsam, sodass das Ganze über ein Kugelmodell realisiert werden muss.

Kugelmodelle lassen sich auf der Grundlage von Voxelmodellen mit einer spezifizierbaren maximalen Abweichung erzeugen. Da nun aber der Roboter als Dreiecksmodell vorliegt, man aber das Kugelmodell nur auf Basis eines Voxelmodells erzeugen kann, ist es notwendig das Dreiecksmodell zunächst in ein Voxelmodell zu konvertieren. Würde man nun auf Basis des resultierenden Voxelmodells ein Kugelmodell erzeugen, so hätte man das Problem, das wiederum dessen Konvertierung in ein Voxelmodell weder eine verkleinerte Voxelapproximation noch eine Teilmenge davon gegenüber des ursprünglichen Dreiecksmodells darstellen würde – es wäre zu groß. Es ist also notwendig, das Voxelmodell zunächst zu kontrahieren, damit das daraus erzeugte Kugelmodell später tatsächlich “dünn” genug ist, mittels seiner Konvertierung zu einer vergrößerten Voxelapproximation eine verkleinerte Voxelapproximation bzw. einer Teilmenge der verkleinerten Voxelapproximation in bezug auf das Dreiecksmodell zu erzeugen. Um welchen Radius nun kontrahiert werden muss, hängt unter anderem davon ab, mit welcher Voxelgröße

man später arbeiten möchte. Bevor die Parametrisierung der einzelnen Schritte betrachtet wird, werden diese zunächst nochmal in Kürze aufgelistet und in “Offline” und “Online” gruppiert (vgl. Abbildung A.1):

1. “Offline”:
 - a) Exportiere CAD-Daten in (exaktes) Dreiecksmodell
 - b) Konvertiere Dreiecksmodell in verkleinerte Voxelapproximation
 - c) Kontrahiere Voxelapproximation
 - d) Erzeuge Kugelmodell aus kontrahierter Voxelapproximation
2. “Online”:
 - a) Konvertiere Kugelmodell zu vergrößerter Voxelapproximation

Es stellt sich nun also die Frage, wie man die einzelnen Schritte parametrisieren muss, damit tatsächlich in 2a eine verkleinerte Voxelapproximation oder eine Teilmenge dieser bezüglich des in 1a erzeugten Dreiecksmodells herauskommt. Die in 1b bis 1d verwendete Voxelapproximation hat nichts mit der späteren “Online”-Voxelapproximation in 2a zu tun. Hier kann also im Rahmen des verfügbaren Speicherplatzes und der verfügbaren Zeit eine beliebig hohe Voxelraumauflösung gewählt werden.

Man muss beachten, dass das Resultat in 1b direkt die Oberfläche des Dreiecksmodells berühren kann. Liegen in den Schritten 1b bis 1d Voxel mit einer Diagonalen von 1 cm Länge vor und will man später mit einer Voxelraumauflösung online (2a) arbeiten, deren Voxel diagonalen eine Länge von 4 cm haben, so muss man beispielsweise in Schritt 1c das Voxelmodell um 7 cm kontrahieren, sodass man im Schritt 1d 2,5 cm als Maximalabweichung bei der Erstellung des Kugelmodells angeben kann: Bei der Kontraktion um 7 cm und einem resultierenden Kontraktions-Intervall von $[7\text{ cm} - 0,5\text{ cm}, 7\text{ cm} + 0,5\text{ cm}]$ kann also das Volumen eines Voxels 6,5 cm dicht an der Oberfläche des Dreiecksmodells liegen. Bei der Verwendung der vergrößerten Voxelapproximation eines Kugelmodells in 2a können die resultierenden Voxel 4 cm dichter an der Oberfläche liegen, sodass also 2,5 cm als Maximalabweichung zur Erzeugung der Kugelmodelle übrig bleiben.

Durch diese Parametrisierung der einzelnen Schritte ist damit sichergestellt, dass die Konvertierung in 2a bei einer Verwendung von Voxel mit einer Diagonalen von 4 cm Länge zu einer

verkleinerten Voxelapproximation oder einer Teilmenge dieser bezüglich des Dreiecksmodells aus 1a führt, also alle Voxel im Inneren des Dreiecksmodells liegen.

A.6. Schlussfolgerungen

Mit den in diesem Kapitel geschaffenen Werkzeugen lassen sich Dreiecksmodelle in vergrößerte und verkleinerte Voxelapproximationen und Kugelmodelle in vergrößerte Voxelapproximationen konvertieren. Außerdem lassen sich aus Voxelmodellen Kugelmodelle erzeugen. Mit dem Umweg über Voxelmodelle lassen sich aus Dreiecksmodellen Kugelmodelle generieren. Um mittels Kugelmodellen verkleinerte Voxelapproximationen – oder genauer Teilmengen dieser – erstellen zu können müssen Voxelmodelle im Zwischenschritt geeignet kontrahiert werden. Im Rahmen von Hüllenbetrachtungen in Plausibilisierungsfunktionen ist eine Expansion der Voxelmodelle notwendig.

B. Symbole und Bezeichnungen

In diesem Anhangskapitel findet sich eine alphabetisch sortierte Übersicht über die verwendeten Symbole und Bezeichnungen jeweils mit einer kurzen Beschreibung.

2^A Potenzmenge von A .

$[p_1, p_2]$ Diese Darstellung beschreibt einen Strahl, ausgehend von Punkt p_1 in Richtung und durch Punkt p_2 .

$\lambda(A)$ Lebesgue-Maß (Volumeninhalte) von A

$\bar{\mathbb{R}}$ Menge der erweiterten reellen Zahlen $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$

Allgemeine globale Erfassbarkeitsfunktion (meist p_e) Die allgemeine globale Erfassbarkeitsfunktion ist eine Plausibilisierungsfunktion vom Typ R und verwirft Regionen einer gegebenen geometrischen Rekonstruktion, die in weniger als einer spezifizierbaren Anzahl von Sensoren als gesuchte Objekte erfasst werden. Es spielt dabei keine Rolle ob die Regionen komplett oder nur zum Teil als gesuchte Objekte erfasst wird. Siehe Definition 30.

Allgemeine lokale Erfassbarkeitsfunktion (meist p_E) Die allgemeine lokale Erfassbarkeitsfunktion ist eine Abwandlung der allgemeinen globalen Erfassbarkeitsfunktion. Der Unterschied besteht darin, dass im Gegensatz zur globalen Variante der Parameter für die Mindestanzahl an Sensoren die eine Region als gesuchtes Objekt detektieren müssen ortsabhängig angegeben wird. Siehe Definition 32.

Als gesuchtes Objekt erfassen (meist D) Der ausschließbare Raumbereich eines Sensors in einer Umwelt ohne gesuchte Objekte abzüglich des ausschließbaren Raumbereiches des gleichen

B. Symbole und Bezeichnungen

Sensors in der Umwelt jedoch mit gesuchten Objekten wird als Raumbereich der als gesuchtes Objekt erfasst wird bezeichnet. Siehe Definition 29.

Anwesenheitsfunktion (meist a) Die Anwesenheitsfunktion wird in einem Anwesenheitssensor benutzt und liefert den Wert 1, wenn sich ein gesuchtes Objekt in einem zugehörigen überwachten Raumbereich befindet und ansonsten den Wert 0. Siehe Definition 12.

Ausschließbarer Raumbereich einer Tiefenkamera (meist A_D^K) Der ausschließbare Raumbereich einer Tiefenkamera ist der Raumbereich, der mittels einer Tiefenkamera als frei von gesuchten Objekten ermittelt wird. K steht dabei für die kalibrierte Kamera, und D kennzeichnet hier, dass diese Tiefeninformaton liefert. Siehe Definition 17.

Ausschließbarer Raumbereich eines Anwesenheitssensors (meist A_A) Der ausschließbare Raumbereich eines Anwesenheitssensors ist der vom Anwesenheitssensor überwachte Raumbereich, wenn dessen Anwesenheitsfunktion den Wert 0 liefert. Siehe Definition 13.

Ausschließbarkeitsfunktion (meist p_a) Die Ausschließbarkeitsfunktion ist eine Plausibilisierungsfunktion vom Typ E und verwirft Elemente einer gegebenen geometrischen Rekonstruktion, in denen sich kein gesuchtes Objekt befinden kann, beispielsweise auf Basis von Sensorinformation. Siehe Definition 11.

Dreiecksmodell Ein Dreiecksmodell ist eine geometrische Repräsentation – beispielsweise zur Beschreibung der Geometrie von Gegenständen – und besteht aus einer Menge von Dreiecken, die wiederum aus drei Punkten im \mathbb{R}^3 bestehen. Siehe Definition 71.

Eckennachbarn eines Voxels Eckennachbarn eines Voxels sind alle Voxel, deren Position sich in genau drei Koordinaten um den Wert 1 gegenüber dem betrachteten Voxel unterscheiden. Siehe Definition 48.

Elementbasiertes Plausibilisierungstupel Ein Plausibilisierungstupel wird als elementbasiert bezeichnet, wenn für jedes einzelne Element unabhängig von allen anderen Elementen einer gegebenen geometrischen Rekonstruktion berechnet wird, ob es auch in der Ausgabe der Plausibilisierungsfunktion enthalten ist. Siehe Definition 6.

Elementbasiertes Plausibilisierungstupel (voxelbasiert) Als elementbasiert wird ein voxelbasiertes Plausibilisierungstupel bezeichnet, wenn für jeden einzelnen Voxel unabhängig von allen anderen Voxeln einer gegebenen voxelbasierten geometrischen Rekonstruktion berechnet wird, ob dieser auch in der Ausgabe der voxelbasierten Plausibilisierungsfunktion enthalten ist. Siehe Definition 55.

Erweiterte Hülle Die erweiterte Hülle $\hat{M}_r^G \subset \mathbb{R}^n$ eines Gegenstandes $G \subset \mathbb{R}^n$ mit dem Radius $r \in \mathbb{R}^+$ ist die Hülle M_r^G vereinigt mit G selbst. Siehe Definition 36.

Erweiterte, mittels Farbmodell tiefenbeschreibende Farbkamera-Pixelfunktion (meist \hat{i}_C^F) Eine mittels Farbmodell tiefenbeschreibende Farbkamera-Pixelfunktion, erweitert um andere ausschließbare Raumbereiche. Das F steht für *Farbmodell* und C für die zugrunde liegende Pixelinformation, nämlich die *Farbe* der Pixel (einer Farbkamera). Siehe Definition 26.

Erweiterte, mittels Hintergrundmodell tiefenbeschreibende Farbkamera-Pixelfunktion (meist \hat{i}_C^B) Eine mittels Hintergrundmodell tiefenbeschreibende Farbkamera-Pixelfunktion, erweitert um andere ausschließbare Raumbereiche. Das H steht für *Hintergrundmodell* und C für die zugrunde liegende Pixelinformation, nämlich die *Farbe* der Pixel (einer Farbkamera). Siehe Definition 27.

Erweiterte tiefenbeschreibende Real-Pixelfunktion (meist \hat{i}_r) Die erweiterte tiefenbeschreibende Real-Pixelfunktion ist das Pendant zu der erweiterten tiefenbeschreibenden Pixelfunktion, mit dem Unterschied, dass statt einzelner Sichtrichtungen hier Mengen von Sichtrichtungen betrachtet werden. Siehe Definition 62.

Erweiterte Tiefenkamera-Pixelfunktion (meist \hat{i}_D) Eine Tiefenkamera-Pixelfunktion, erweitert um andere ausschließbare Raumbereiche. Siehe Definition 28.

Expansion einer Menge Die Expansion einer bestimmten Menge $M \subset \mathbb{R}^n$ um einen bestimmten Radius beschreibt die Menge selbst, zuzüglich aller Elemente $m \in \mathbb{R}^n$, deren Distanz zu dieser Menge geringer oder gleich dem Radius ist. Siehe Definition 81.

Farbbasierte Segmentierung (meist d_F) Ist die Menge aller Farben gesuchter Objekte bekannt, so lässt sich ein Kamerabild in potentiell gesuchte Objekt bzw. nicht-gesuchte Objekte segmentieren. Siehe Definition 19.

Farbkamera-Pixelfunktion (meist i_C) Eine Farbkamera-Pixelfunktion ist eine Funktion, die jeder Sichtrichtung einen Farbwert zuordnet. Sie modelliert damit das Verhalten einer Farbkamera. Siehe Definition 16.

Fläche eines Dreiecks Die Fläche eines Dreiecks enthält alle Punkte in der Ebene des Dreiecks, welche gleichzeitig innerhalb der drei Kanten des Dreiecks liegen. Siehe Definition 73.

Flächennachbarn eines Voxels Flächennachbarn eines Voxels sind alle Voxel, deren Position sich in genau einer Koordinate um den Wert 1 gegenüber dem betrachteten Voxel unterscheiden. Siehe Definition 48.

Formfunktion (meist p_f) Die Formfunktion ist eine Plausibilisierungsfunktion vom Typ R. Sie verwirft Regionen einer gegebenen geometrischen Rekonstruktion, die komplett innerhalb spezifizierbarer Raumbereiche liegen. Siehe Definition 34.

Geometrische Rekonstruktion (meist $R \subseteq Z \subset \mathbb{R}^n$) Eine konservative Approximation von gesuchten Objekten innerhalb einer Rekonstruktionszone Z . Siehe Definition 1.

Geschlossenes Dreiecksmodell Ein Dreiecksmodell heißt geschlossen, wenn es keinen Punkt im \mathbb{R}^3 gibt dem sowohl Vorder- als auch Rückseiten von sichtbaren Dreiecken zugewandt sind. Siehe Definition 77.

Gesuchte Objekte einfach voxelbasiert erfasst Dies bezeichnet die direkte Übertragung in eine voxelbasierte Betrachtung der Bezeichnung "Als gesuchtes Objekt erfassen". Es wird als "einfach" benannt, da hierbei keinerlei Auswirkungen der diskreten Betrachtungsweise berücksichtigt werden. Siehe Definition 63.

Gesuchte Objekte konservativ voxelbasiert erfasst Dies bezeichnet die Übertragung in eine voxelbasierte Betrachtung der Bezeichnung "Als gesuchtes Objekt erfassen", jedoch mit

Berücksichtigung der Auswirkungen der diskreten Betrachtungsweise gegenüber der “einfachen” Variante. Siehe Definition 64.

Globale Distanzfunktion (meist p_d) Die globale Distanzfunktion ist eine Plausibilisierungsfunktion vom Typ R. Sie verwirft Regionen einer gegebenen geometrischen Rekonstruktion, welche einen größeren Abstand zu einer spezifizierbaren Menge haben als ein spezifizierbarer Maximalabstand. Siehe Definition 39.

Globale Volumenfunktion (meist p_v) Die globale Volumenfunktion ist eine Plausibilisierungsfunktion vom Typ R. Sie verwirft Regionen einer gegebenen geometrischen Rekonstruktion, welche einen geringeren Volumeninhalt haben als der spezifizierbare Mindestvolumeninhalt. Siehe Definition 37.

Güte einer geometrischen Rekonstruktion Eine geometrische Rekonstruktion $R_1 \subseteq Z \subset \mathbb{R}^n$ besitzt eine höhere Güte als eine geometrische Rekonstruktion $R_2 \subseteq Z \subset \mathbb{R}^n$, wenn sie eine echte Teilmenge von R_2 ist. Sie besitzt eine geringere Güte, wenn sie eine echte Obermenge von R_2 ist. Sie ist von gleicher Güte, wenn sie gleich R_2 ist. Siehe Definition 2.

Güte einer voxelbasierten geometrischen Rekonstruktion Analog zu der Güte einer geometrischen Rekonstruktion, mit dem Unterschied, dass hier Voxelpositionen verwendet werden, also \mathbb{Z}^3 statt \mathbb{R}^n . Siehe Definition 52.

Hintergrundmodell (meist h_C^E) Ein Hintergrundmodell enthält die Farbwerte C , die eine Farbkamera-Pixelfunktion in einer Umwelt E ohne gesuchte Objekte liefern würde. Ein Hintergrundmodell kann zur Vordergrund-/Hintergrund-Segmentierung verwendet werden. Siehe Definition 22.

Hülle Die Hülle $M_r^G \subset \mathbb{R}^n$ eines Gegenstandes $G \subset \mathbb{R}^n$ mit dem Radius $r \in \mathbb{R}^+$ beschreibt alle Elemente die eine kleinere oder gleiche Distanz zu G haben als r und nicht in G liegen. Siehe Definition 35.

Kalibrierte Kamera (meist K) Eine kalibrierte Kamera ist ein Tupel, bestehend aus einem Projektionszentrum und den Sichtrichtungen, welche eine Teilmenge der Oberfläche der Ein-

B. Symbole und Bezeichnungen

heitskugel darstellen. Siehe Definition 14.

Kalibrierte reale Kamera (meist K_r) Eine kalibrierte reale Kamera K_r ist wie die kalibrierte Kamera ein Tupel, hier jedoch bestehend aus einem Projektionszentrum und aus Teilmengen von Sichtrichtungen. Durch diese Teilmengen werden reale Pixel, d.h. mit einer gewissen Ausdehnung auf dem Sensor modelliert. Siehe Definition 59.

Kantennachbarn eines Voxels Kantennachbarn eines Voxels sind alle Voxel, deren Position sich in genau zwei Koordinaten um den Wert 1 gegenüber dem betrachteten Voxel unterscheiden. Siehe Definition 48.

Kontraktion einer Menge Die Kontraktion einer bestimmten Menge $M \subset \mathbb{R}^n$ um einen bestimmten Radius beschreibt die Menge selbst, abzüglich aller Elemente $m \in M$, deren Distanz zu $\mathbb{R}^n \setminus M$ geringer als der Radius ist. Siehe Definition 81.

Kugelmodell Ein Kugelmodell besteht aus einer Menge von Kugeln, die durch eine Position und einen Radius definiert sind. Siehe Definition 79.

Lokale Distanzfunktion (meist p_D) Die lokale Distanzfunktion ist eine Abwandlung der globalen Distanzfunktion. Der Unterschied besteht darin, dass hier der spezifizierbare Maximalabstand ortsabhängig angegeben wird. Siehe Definition 40.

Lokale Volumenfunktion (meist p_V) Die lokale Volumenfunktion ist eine Abwandlung der globalen Volumenfunktion. Der Unterschied besteht darin, dass hier das spezifizierbare Mindestvolumen ortsabhängig angegeben wird. Siehe Definition 38.

Menge aller Farben gesuchter Objekte (meist F_U) Die Menge aller Farben gesuchter Objekte sind diejenigen Farben, die von einer Farbkamera-Pixelfunktion ausgegeben werden, wenn sie auf gesuchte Objekt angewendet wird. Siehe Definition 18.

Menge aller Regionen einer Menge (meist \mathbb{Q}_M) Die Menge \mathbb{Q}_M aller Regionen einer Menge $M \subset \mathbb{R}^n$ enthält alle Teilmengen von M , die eine Region darstellen. Siehe Definition 5.

Menge aller sichtbaren Dreiecke bezüglich eines Punktes Die Menge aller sichtbaren Dreiecke bezüglich eines Punktes besteht aus den Dreiecken, für die es ein Strahl, ausgehend von dem Punkt, gibt, bezüglich dessen sie sichtbar sind. Siehe Definition 75.

Menge aller Voxelregionen einer Menge (meist \mathbb{Q}_M^V) Die Menge \mathbb{Q}_M^V aller Voxelregionen einer Menge $M^V \subset \mathbb{Z}^3$ enthält alle Teilmengen von M^V , die eine Voxelregion darstellen. Siehe Definition 49.

Mittels Farbmodell ausschließbarer Raumbereich einer Farbkamera (meist $A_C^{K,F}$) Hat man eine mittels Farbmodell tiefenbeschreibende Farbkamera-Pixelfunktion, so lässt sich – analog zum ausschließbaren Raumbereich einer Tiefenkamera – der mittels Farbmodell ausschließbare Raumbereich einer Farbkamera ableiten. K steht dabei für die kalibrierte Kamera, und C kennzeichnet hier, dass diese Farbinformation liefert, die mittels eines Farbmodells F verarbeitet werden. Siehe Definition 21.

Mittels Farbmodell tiefenbeschreibende Farbkamera-Pixelfunktion (meist i_C^F) Liegt eine farbbasierte Segmentierung in der Form vor, dass man Pixel als potentiell gesuchtes Objekt bzw. nicht-gesuchtes Objekt klassifizieren kann, so lässt sich in Kombination mit einer geometrischen Modellierung der bekannten Umwelt eine tiefenbeschreibende Pixelfunktion daraus ableiten. F kennzeichnet hier, dass ein *Farbmodell* verwendet wird und C , dass die zugrundeliegende Pixelinformation *Farbe* ist. Siehe Definition 20.

Mittels Hintergrundmodell ausschließbarer Raumbereich einer Farbkamera (meist $A_C^{K,B}$) Ist eine mittels Hintergrundmodell tiefenbeschreibende Farbkamera-Pixelfunktion gegeben, so lässt sich – analog zum ausschließbaren Raumbereich einer Tiefenkamera – der mittels Hintergrundmodell ausschließbare Raumbereich einer Farbkamera ableiten. K steht dabei für die kalibrierte Kamera, und C kennzeichnet hier, dass diese Farbinformation liefert, die mittels eines Hintergrundmodells B verarbeitet werden. Siehe Definition 25.

Mittels Hintergrundmodell tiefenbeschreibende Farbkamera-Pixelfunktion (meist i_C^B) Liegt einer Vordergrund-Hintergrund-Segmentierung vor, so lässt sich analog zur mittels Farbmodell tiefenbeschreibenden Farbkamera-Pixelfunktion eine tiefenbeschreibende Pixelfunktion

B. Symbole und Bezeichnungen

daraus ableiten. B kennzeichnet hier, dass ein *Hintergrundmodell* verwendet wird und C , dass die zugrundeliegende Pixelinformation *Farbe* ist. Siehe Definition 24.

Normale eines Dreiecks Die Normale eines Dreiecks ist basierend auf dem Kreuzprodukt zweier Kanten des Dreiecks definiert. Siehe Definition 72.

Oberfläche der Einheitskugel (meist ζ) Die Oberfläche der Einheitskugel im \mathbb{R}^n besteht aus allen Elementen mit einer Distanz von 1 zum Ursprung. Sie wird zur Definition einer kalibrierten Kamera verwendet. Siehe Definition 14.

Optimalrekonstruktion Die geometrische Rekonstruktion, die exakt den gesuchten Objekten innerhalb einer Rekonstruktionszone entspricht. Siehe Definition 3.

Pixel (meist s) Ein Pixel ist ein einzelnes Element aus den Sichtrichtungen einer kalibrierten Kamera. Siehe Definition 14.

Plausibel bezüglich Eine Menge $M \subseteq Z \subset \mathbb{R}^n$ innerhalb einer Rekonstruktionszone Z ist plausibel bezüglich eines Plausibilisierungstupels, wenn die darin gegebenen Plausibilisierungsfunktion und gegebene Information, angewendet auf M , als Ausgabe wieder M hat. M ist bezüglich einer Menge von Plausibilisierungstupeln plausibel, wenn M bezüglich jedes einzelnen enthaltenen Plausibilisierungstupels plausibel ist. Siehe Definition 4. Gleiches, jedoch im Kontext einer Voxelrepräsentation gilt auch bezüglich voxelbasierten Plausibilisierungstupeln bzw. bezüglich Mengen von voxelbasierten Plausibilisierungstupeln. Siehe Definition 54.

Plausibilisierungsfunktion (meist p_x) Eine Plausibilisierungsfunktion nutzt bereitgestellte Information, um aus einer gegebenen geometrischen Rekonstruktion eine geometrische Rekonstruktion mit gleicher oder höherer Güte zu erzeugen. Siehe Definition 4.

Plausibilisierungspermutation Eine Plausibilisierungspermutation stellt eine Reihenfolge aller Plausibilisierungstupel einer Menge von Plausibilisierungstupeln dar. Siehe Definition 9.

Plausibilisierungstupel (meist (p_x, i_x)) Ein Plausibilisierungstupel ist ein Tupel, bestehend aus einer Plausibilisierungsfunktion und gegebener Information. Siehe Definition 4.

Rand einer Voxelmenge Der Rand einer Menge von Voxelpositionen sind daraus die Voxelpositionen, deren Voxel Nachbarn besitzen, die nicht der betrachteten Menge an Voxelpositionen angehören. Siehe Definition 50.

Randfunktion (meist p_r) Die Randfunktion ist eine Plausibilisierungsfunktion vom Typ R. Sie verwirft Regionen einer gegebenen geometrischen Rekonstruktion, welche sich nicht mit dem Rand der Rekonstruktionszone überschneiden. Siehe Definition 43.

Reales Pixel (meist $s_r \in S_r$) Reale Pixel haben gegenüber den (zuvor eingeführten) Pixeln eine gewisse Ausdehnung auf dem Sensor und werden im Zusammenhang mit realen Kameras verwendet. Sie bestehen daher aus mehreren Sichtrichtungen: $S_r \subseteq 2^S$ ist eine Menge von Teilmengen der Sichtrichtungen $S \subseteq \zeta$, wobei ζ die Oberfläche der Einheitskugel darstellt. Siehe Definition 59.

Region einer Menge (meist $\varrho \subseteq M \subset \mathbb{R}^n$) Eine Region $\varrho \subseteq M$ einer Menge $M \subset \mathbb{R}^n$ ist eine größte zusammenhängende Teilmenge. Siehe Definition 5.

Regionenbasiertes Plausibilisierungstupel Ein Plausibilisierungstupel wird als regionenbasiert bezeichnet, wenn für jede einzelne Region unabhängig von allen anderen Regionen einer gegebenen geometrischen Rekonstruktion berechnet wird, ob diese auch in der Ausgabe der Plausibilisierungsfunktion enthalten ist. Siehe Definition 6.

Regionenbasiertes Plausibilisierungstupel (voxelbasiert) Ein voxelbasiertes Plausibilisierungstupel wird als regionenbasiert bezeichnet, wenn für jede einzelne Voxelregion unabhängig von allen anderen Voxelregionen einer gegebenen voxelbasierten geometrischen Rekonstruktion berechnet wird, ob diese auch in der Ausgabe der voxelbasierten Plausibilisierungsfunktion enthalten ist. Siehe Definition 55.

Rekonstruktionszone (meist $Z \subset \mathbb{R}^n$) Gesuchte Objekte werden mittels der wissens- und sensorbasierten geometrischen Rekonstruktion ausschließlich innerhalb der Rekonstruktionszone betrachtet und rekonstruiert. Diese lässt sich allerdings frei wählen. Siehe Definition 1.

Rückprojiziertes freies Volumen (meist V_{s_r}) Das rückprojizierte freie Volumen V_{s_r} eines realen Pixels $s_r \in S_r \subseteq 2^S$ beschreibt alle Raumelemente, die sich entlang der in diesem realen Pixel enthaltenen Sichtrichtungen mit keinem Gegenstand überschneiden. Die Definition basiert auf der tiefenbeschreibenden Real-Pixelfunktion. Siehe Definition 61.

Richtungsspezifische globale Distanzfunktion (meist $p_{\hat{d}}$) Die richtungsspezifische globale Distanzfunktion ist eine Abwandlung der globalen Distanzfunktion. Der Unterschied besteht darin, dass Distanzen nur in einer spezifizierbaren Richtung betrachtet werden. Siehe Definition 41.

Richtungsspezifische lokale Distanzfunktion (meist $p_{\hat{d}}$) Die richtungsspezifische lokale Distanzfunktion ist eine Abwandlung der lokalen Distanzfunktion. Der Unterschied besteht darin, dass Distanzen nur in einer spezifizierbaren Richtung betrachtet werden. Siehe Definition 42.

Sichtbares Dreieck bezüglich eines Strahls Ein Dreieck ist sichtbar bezüglich eines Strahls, wenn sich die Elemente des Strahls mit denen dieses Dreiecks überschneiden und die Distanz vom Startpunkt des Strahls zu dieser Schnittmenge geringer ist als alle Distanzen zu Schnittmengen des Strahls mit anderen Dreiecken. Siehe Definition 74.

Spezielle globale Erfassbarkeitsfunktion (meist $p_{\hat{e}}$) Die spezielle globale Erfassbarkeitsfunktion ist eine Abwandlung der allgemeinen globalen Erfassbarkeitsfunktion. Der Unterschied besteht darin, dass eine Region gesuchter Objekte von einer Mindestanzahl an Sensoren an der "gleichen Stelle" als gesuchte Objekte detektiert werden müssen. Siehe Definition 31.

Spezielle lokale Erfassbarkeitsfunktion (meist $p_{\hat{e}}$) Die spezielle lokale Erfassbarkeitsfunktion ist eine Abwandlung der speziellen globalen Erfassbarkeitsfunktion. Ebenso wie bei der allgemeinen lokalen Erfassbarkeitsfunktion wird hier der Parameter zur Angabe der Mindestanzahl an Sensoren, die eine Region als gesuchtes Objekt erfassen müssen, ortsabhängig angegeben. Definition 33.

Standardrekonstruktion Die geometrische Rekonstruktion, die exakt der Rekonstruktionszone entspricht. Siehe Definition 3.

Teilregionen-verwerfende Plausibilisierungsfunktion Wird durch die Anwendung einer Plausibilisierungsfunktion eine Region einer gegebenen Rekonstruktion in deren Ausgabe verworfen, und würde diese Plausibilisierungsfunktion auch jede Teilmenge der verworfenen Region in der Ausgabe verwerfen, so wird die Plausibilisierungsfunktion als Teilregionen-verwerfend bezeichnet. Siehe Definition 8.

Teilregionen-verwerfende Plausibilisierungsfunktion (voxelbasiert) Wird durch die Anwendung einer voxelbasierten Plausibilisierungsfunktion eine Voxelregion einer gegebenen voxelbasierten Rekonstruktion in deren Ausgabe verworfen, und würde diese voxelbasierte Plausibilisierungsfunktion auch jede Teilmenge der verworfenen Voxelregion in der Ausgabe verwerfen, so wird die voxelbasierte Plausibilisierungsfunktion als Teilregionen-verwerfend bezeichnet. Siehe Definition 57.

Temporalfunktion (meist p_t) Die Temporalfunktion ist eine Plausibilisierungsfunktion vom Typ R. Sie verwirft Regionen einer gegebenen geometrischen Rekonstruktion, welche sich zur vorgehenden geometrischen Rekonstruktion um eine größere Distanz als einer spezifizierbaren Maximaldistanz entfernt haben und gleichzeitig eine größere Distanz zum Rand der Rekonstruktionszone haben als die spezifizierte Maximaldistanz. Siehe Definition 44.

Tiefenbeschreibende Pixelfunktion (meist i) Eine tiefenbeschreibende Pixelfunktion ist eine Funktion die jedem Pixel einen positiven skalaren Wert zuordnet, der eine Distanz entlang einer Sichtrichtung repräsentiert. Siehe Definition 15.

Tiefenbeschreibende Real-Pixelfunktion (meist i_r) Eine tiefenbeschreibende Real-Pixelfunktion ist das Pendant zu einer tiefenbeschreibenden Pixelfunktion – jedoch im Kontext realer Kameras. Es werden statt einzelner Sichtrichtungen Mengen von Sichtrichtungen verwendet. Siehe Definition 60.

Tiefenkamera-Pixelfunktion (meist i_D) Eine Tiefenkamera-Pixelfunktion ist eine konkrete tiefenbeschreibende Pixelfunktion, die das Verhalten einer Tiefenkamera modelliert. Siehe Definition 15.

B. Symbole und Bezeichnungen

Typ E Eine Plausibilisierungsfunktion wird als vom Typ E bezeichnet, wenn sie elementbasiert und vollständig ist. Siehe Definition 10.

Typ R Eine Plausibilisierungsfunktion wird als vom Typ R bezeichnet, wenn sie regionenbasiert, vollständig und Teilregionen-verwerfend ist. Siehe Definition 10.

Vergrößerte Voxelapproximation Die vergrößerte Voxelapproximation einer Menge $M \subset \mathbb{R}^3$ sind alle Voxelpositionen der Voxel eines Voxelraumes, deren Volumen sich mit M überschneidet. Siehe Definition 47.

Verkleinerte Voxelapproximation Die verkleinerte Voxelapproximation einer Menge $M \subset \mathbb{R}^3$ sind alle Voxelpositionen der Voxel eines Voxelraumes, deren Volumen komplett in M liegt. Siehe Definition 47.

Vollständige Plausibilisierungsfunktion Eine Plausibilisierungsfunktion wird als vollständig bezeichnet, wenn ihre Ausgabe plausibel bezüglich des zugehörigen Plausibilisierungstupels ist. Siehe Definition 7.

Vollständige Plausibilisierungsfunktion (voxelbasiert) Eine voxelbasierte Plausibilisierungsfunktion wird als vollständig bezeichnet, wenn ihre Ausgabe plausibel bezüglich des zugehörigen voxelbasierten Plausibilisierungstupels ist. Siehe Definition 56.

Volumen einer Voxelmenge Das Volumen einer Voxelmenge $v_X \subset \mathbb{R}^3$ der Voxel mit den Voxelpositionen $X \subset \mathbb{Z}^3$ ist die Vereinigung der Volumina der einzelnen Voxel. Siehe Definition 46.

Volumen eines Voxels Das Volumen $v_x \subset \mathbb{R}^3$ eines Voxels mit der Voxelposition $x \in \mathbb{Z}^3$ beschreibt eine quaderförmige Menge in einem Voxelraum. Siehe Definition 46.

Vordergrund-Hintergrund-Segmentierung (meist d_B) Ist ein Hintergrundmodell gegeben, so lassen sich Pixel als Vordergrund bzw. Hintergrund klassifizieren. Liefert die Farbkamera-Pixelfunktion den gleichen Wert wie das Hintergrundmodell, so handelt es sich um einen Hintergrundpixel. Siehe Definition 23.

Voxel Ein Voxel ist hier als solches nicht näher definiert. Ihm lässt sich allerdings eine Position, ein Volumen, ein Volumeninhalt und eine Nachbarschaft zuordnen.

Voxelbasierte Ausschließbarkeitsfunktion (meist p_a^V) Die voxelbasierte Ausschließbarkeitsfunktion ist die Ausschließbarkeitsfunktion im Kontext einer Voxelrepräsentation. Siehe Definition 58.

Voxelbasierte Formfunktion (meist p_f^V) Die voxelbasierte Formfunktion ist das Pendant zur Formfunktion: Sie ist eine voxelbasierte Plausibilisierungsfunktion vom Typ R und verwirft Voxelregionen einer gegebenen voxelbasierten geometrischen Rekonstruktion, die komplett innerhalb spezifizierbarer Raumbereiche liegen. Siehe Definition 65.

Voxelbasierte geometrische Rekonstruktion (meist $R^V \subseteq Z^V \subset \mathbb{Z}^3$) Das Volumen der Menge von Voxelpositionen R^V stellt eine konservative Approximation von gesuchten Objekten innerhalb einer voxelbasierten Rekonstruktionszone Z^V . Siehe Definition 51.

Voxelbasierte globale Distanzfunktion (meist p_d^V) Die voxelbasierte globale Distanzfunktion ist das Pendant zur globalen Distanzfunktion, jedoch im Kontext einer Voxelbetrachtung: Die voxelbasierte globale Distanzfunktion ist eine voxelbasierte Plausibilisierungsfunktion vom Typ R. Sie verwirft Voxelregionen einer gegebenen voxelbasierten geometrischen Rekonstruktion, welche einen größeren Abstand zu einer spezifizierbaren Menge haben als ein spezifizierbarer Maximalabstand. Siehe Definition 68.

Voxelbasierte globale Volumenfunktion (meist p_v^V) Die voxelbasierte Volumenfunktion ist das Pendant zur globalen Volumenfunktion, jedoch im Kontext einer Voxelbetrachtung: Die voxelbasierte globale Volumenfunktion ist eine voxelbasierte Plausibilisierungsfunktion vom Typ R. Sie verwirft Voxelregionen einer gegebenen voxelbasierten geometrischen Rekonstruktion, welche einen geringeren Volumeninhalt haben als ein spezifizierbarer Mindestvolumeninhalt. Siehe Definition 66.

Voxelbasierte lokale Volumenfunktion (meist p_l^V) Die voxelbasierte lokale Volumenfunktion ist das Pendant zur lokalen Volumenfunktion, jedoch im Kontext einer Voxelbetrachtung:

B. Symbole und Bezeichnungen

Die voxelbasierte lokale Volumenfunktion ist eine Abwandlung der voxelbasierten globalen Volumenfunktion. Der Unterschied besteht darin, dass hier das spezifizierbare Mindestvolumen ortsabhängig angegeben wird. Siehe Definition 67.

Voxelbasierte Optimalrekonstruktion Die voxelbasierte Optimalrekonstruktion ist die vergrößerten Voxelapproximation der gesuchten Objekte innerhalb der voxelbasierten Rekonstruktionszone. Siehe Definition 53.

Voxelbasierte Plausibilisierungsfunktion (meist p_x^V) Entspricht der Plausibilisierungsfunktion, jedoch im Kontext einer Voxelrepräsentation: Eine voxelbasierte Plausibilisierungsfunktion nutzt bereitgestellte Information, um aus einer gegebenen voxelbasierten geometrischen Rekonstruktion eine voxelbasierte geometrische Rekonstruktion mit gleicher oder höherer Güte zu erzeugen. Siehe Definition 54.

Voxelbasiertes Plausibilisierungstupel (meist (p_x^V, i_x)) Entspricht dem Plausibilisierungstupel, jedoch im Kontext einer Voxelrepräsentation: Das voxelbasierte Plausibilisierungstupel besteht aus einer Plausibilisierungsfunktion und der Information, die diese Plausibilisierungsfunktion nutzt. Siehe Definition 54.

Voxelbasierte Randfunktion (meist p_r^V) Die voxelbasierte Randfunktion ist das Pendant zur Randfunktion, jedoch im Kontext einer Voxelbetrachtung: Die voxelbasierte Randfunktion ist eine voxelbasierte Plausibilisierungsfunktion vom Typ R. Sie verwirft Voxelregionen einer gegebenen voxelbasierten geometrischen Rekonstruktion, welche sich nicht mit dem Rand der voxelbasierten Rekonstruktionszone überschneiden. Siehe Definition 69.

Voxelbasierte Rekonstruktionszone (meist $Z^V \subset \mathbb{Z}^3$) Im Kontext einer Voxelrepräsentation werden gesuchte Objekte mittels der wissens- und sensorbasierten geometrischen Rekonstruktion ausschließlich innerhalb der voxelbasierten Rekonstruktionszone betrachtet und rekonstruiert. Diese lässt sich im Rahmen eines Voxelaums frei wählen. Siehe Definition 51.

Voxelbasierte Standardrekonstruktion Die voxelbasierte Standardrekonstruktion entspricht der voxelbasierten Rekonstruktionszone. Siehe Definition 53.

Voxelbasierte Temporalfunktion (meist p_t^V) Die voxelbasierte Temporalfunktion ist das Pendant zur Temporalfunktion, jedoch im Kontext einer Voxelbetrachtung: Die voxelbasierte Temporalfunktion ist eine voxelbasierte Plausibilisierungsfunktion vom Typ R. Sie verwirft Voxelregionen einer gegebenen voxelbasierten geometrischen Rekonstruktion, welche sich zur vorgehenden voxelbasierten geometrischen Rekonstruktion um eine größere Distanz als einer spezifizierbaren Maximaldistanz entfernt haben und gleichzeitig eine größere Distanz zum Rand der voxelbasierten Rekonstruktionszone haben als die spezifizierte Maximaldistanz. Siehe Definition 70.

Voxelposition (meist $x \in \mathbb{Z}^3$) Eine Voxelposition besteht aus drei Koordinaten im \mathbb{Z}^3 . Siehe Definition 46.

Voxelraum (meist V) Ein Voxelraum ist definiert durch ein Tripel bestehend aus einem minimalen und einem maximalen Punkt sowie einer äquidistanten Unterteilung pro Dimension. Siehe Definition 45.

Voxelregion einer Menge M^V (meist ϱ^V) Eine Region $\varrho^V \subseteq M^V$ einer Menge von Voxelpositionen $M \subset \mathbb{Z}^3$ ist eine größte zusammenhängende Teilmenge in einem Voxel-Nachbarschaftsbeschreibenden Graphen. Siehe Definition 49.