

**Next-Best-View-Planning
auf Voxelkarten durch
einen mit Tiefensensoren
ausgestatteten mobilen
Roboter**

Christopher Buschor



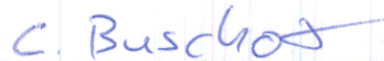
NEXT-BEST-VIEW-PLANNING AUF
VOXELKARTEN DURCH EINEN MIT
TIEFENSENSOREN AUSGESTATTETEN
MOBILEN ROBOTER

Freigabe:

Der Bearbeiter:

Unterschriften

Christopher Buschor

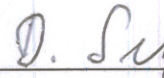


Die Betreuer:

Dipl. Math. Christian Rink

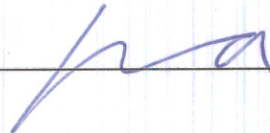


Dipl. Math. techn. Daniel Seth



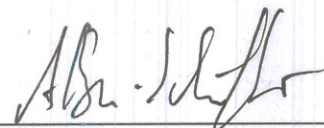
Der Abteilungsleiter:

Dr. -Ing. Michael Suppa



Der Institutsdirektor:

Prof. Dr. -Ing. Alin Albu-Schäffer



Dieser Bericht enthält 49 Seiten, davon 11 Abbildungen und 5 Tabellen

Bachelorarbeit

Next-Best-View-Planning auf Voxelkarten zur Exploration durch einen mit Tiefensensoren ausgestatteten mobilen Roboter

Christopher Buschor

5. August 2013



Lehrstuhl für Datenverarbeitung
Technische Universität München



Christopher Buschor. *Next-Best-View-Planning auf Voxelkarten zur Exploration durch einen mit Tiefensensoren ausgestatteten mobilen Roboter*. Bachelorarbeit, Technische Universität München, München, 2013.

Betreut von Prof. Dr.-Ing. K. Diepold und Dipl.-Ing Johannes Günther, sowie Dipl.-Math Christian Rink (DLR); eingereicht am 5. August 2013 bei der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München.

© 2013 Christopher Buschor

Lehrstuhl für Datenverarbeitung, Technische Universität München, 80290 München, <http://www.ldv.ei.tum.de>.

Dieses Werk ist unter einem Creative Commons Namensnennung 3.0 Deutschland Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by/3.0/de/> oder schicken Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Zusammenfassung

Es sind am Institut für Robotik und Mechatronik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) Algorithmen auf mobilen Robotern implementiert worden, die sich dadurch in bekannten 3D-Karten lokalisieren und eine 2D-Navigation zur Verfügung stellen können. Am Roboter sind ToF-Kameras¹, sowie entsprechende Sensoren zur Odometrie² vorhanden. Um auch unbekannte Umgebungen kartieren zu können, wird derzeit ein SLAM³-Verfahren implementiert. Soll der Roboter diese Aufgabe autonom erfüllen, ist eine Explorationskomponente notwendig, die nach jedem Kartierungsschritt einen Standpunkt für die nächsten Kameraaufnahmen berechnet. Für den Fall der Lokalisierung in einer bekannten Umgebung existiert ein sehr einfaches Verfahren, welches aus einer lokalen Karte Anfahrtspunkte berechnet. Dies ermöglicht ein autonomes Navigieren des Roboters, bis die Lokalisierung erfolgreich ist und auf eine Navigation in der bekannten Karte der Umgebung zurückgegriffen werden kann.

Dieses Verfahren hat sich für eine effiziente Exploration jedoch als untauglich erwiesen. Zur Entwicklung eines geeigneten Verfahrens wurden zunächst verschiedene, bereits erprobte Next-Best-View-Explorationsverfahren analysiert und bzgl. der Adaptierbarkeit für die Problemstellung untersucht. Es hat sich dabei herausgestellt, dass es sinnvoll ist, einen Algorithmus zu verwenden, welcher Next-Best-View-Positionen stets an Übergängen von bereits erkundeten zu noch unbekanntem Gebiet berechnet. Werden mehrere mögliche nächste Anfahrtpunkte gefunden, wird über eine Gewichtung von Fahrtkosten und erwarteten Informationsgewinn, der als am geeignetsten bestimmte Punkt gewählt. Dadurch wird erreicht, dass die Fahrtrichtung prinzipiell Richtung unbekanntem Raum liegt und pro Messung möglichst viel unbekanntes Gebiet vermessen werden kann. Bei Räumen beispielsweise werden Durchfahrten zu weiteren unbekanntem Räumen schnell gefunden, wie es auch in der Simulation nachzuvollziehen war.

Implementiert wurde das neu entwickelte Verfahren auf Voxelkarten⁴, in dem sog. Probabilistic Dynamic Octree Space. In diesem Raum werden Voxeln Besetzungswahrscheinlichkeiten zugewiesen und solche mit gleichem Inhalt, zur Speicherplatzminimierung, zu Octrees zusammengefasst. Das entwickelte Verfahren stellt eine effiziente Explorationskomponente dar und kann für SLAM Verfahren genutzt werden.

¹Time-of-Flight-Kameras messen über die Laufzeit simultan die Entfernung zu mehreren zusammenhängenden Raumpunkten. Daraus kann anschließend eine Tiefenkarte erstellt werden, in der z.B. über Grauwerte der jeweilige Abstand zu den Pixeln dargestellt wird.

²Positionsbestimmung eines mobilen Systems anhand der Daten seines Antriebssystems

³Simultaneous localization and mapping: Eine Technik, um mit autonomen Robotern eine Karte in unbekannter Umgebung zu erstellen oder aktualisieren während fortlaufender Positionsbestimmung des Roboters.

⁴Dreidimensionale Rastergraphik, welche aus Datenpunkten, den Voxeln, besteht.

Inhaltsverzeichnis

1	Einführung	11
1.1	Motivation	11
1.2	Aufgabenstellung und Ziele der Arbeit	12
1.3	Aufbau der Arbeit	12
2	Überblick über aktuelle Next-Best-View-Explorationsverfahren	15
2.1	Die Sensor-based Random Tree (SRT) Methode	15
2.2	Frontier-Based SRT	18
2.3	Frontier-Based Exploration mit Evidence Grid	18
2.4	Die MCDM Strategie	19
3	Auswahl von geeigneten Verfahren für die Problemstellung	21
3.1	Gegebenheiten und Anforderungen	21
3.1.1	Gegebenheiten	21
3.1.2	Allgemeine Anforderungen	23
3.1.3	Spezielle Anforderungen	24
3.2	Überprüfung der recherchierten Verfahren bzgl. den Anforderungen	24
4	Implementierung und Anpassung des ausgewählten Verfahrens	27
4.1	Voraussetzungen	27
4.2	Ausgangslage	27
4.3	Detektion möglicher nächster Observationspunkte	28
4.4	Auswahl des besten aller detektierten Anfahrpunkte	32
4.5	Problembehandlung falls kein Anfahrpunkt bestimmbar	33
5	Experimentelle Erprobung und Vergleich mit bisherigem Verfahren	37
5.1	Experimentelle Erprobung des neuen Verfahrens in der Simulationsumgebung	37
5.2	Performancevergleich des neuen Verfahrens mit dem Bisherigen	38
6	Zusammenfassung und Ausblick	45

Abbildungsverzeichnis

2.1	Explorationsprozess	16
2.2	SRT Modelle	17
2.3	Definition der obstacle, free und frontier arcs	18
3.1	Simulationsumgebung	23
4.1	Ausgangssituation nach initialer Messung	30
4.2	Erreichbare Voxel sowie Grenzvoxel nach initialem Scan	31
5.1	Simulationsumgebung nach vollständiger Exploration	38
5.2	Situation nach zweitem Explorationsschritt	39
5.3	Bestimmung des Anfahrpunktes einer Grenzregion	40
5.4	Zeitlicher Verlauf der beiden Verfahren	42
5.5	Verlauf der beiden Verfahren mit variablen Startpunkten	43

Tabellenverzeichnis

2.1	Gegenüberstellungsmatrix des AHP	19
2.2	Bedeutung der Werte in der AHP Matrix	20
2.3	AHP Matrix mit berechneten Eigenvektoren	20
4.1	Zustände der Voxel in den Ebenen	29
5.1	Auswertung der beiden Verfahren, gemittelte Werte nach 10 Durchläufen, zufällig variiertes Startpunkt	41

1 Einführung

1.1 Motivation

Am Institut für Robotik und Mechatronik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) wurden Algorithmen entwickelt, die eine Lokalisierung in bekannten 3D-Karten, sowie eine 2D-Navigation in diesen Karten der mobilen Plattform des „Rollin’ Justin“ oder dem „omniRob“ von Kuka ermöglichen. Die Navigation findet derzeit in einer bekannten Karte statt. Der Roboter lokalisiert sich dazu durch Messungen mit seinen ToF-Kameras¹. Falls der Roboter sich an seiner Lage nicht erfolgreich lokalisieren kann, existiert ein sehr einfaches Verfahren, um den Roboter fahren zu lassen und erneut Messungen durchzuführen, solange, bis die Lokalisierung erfolgreich ist.

Eine angestrebte Weiterentwicklung ist die Erstellung von 3D-Modellen auch in unbekanntem Gebiet, welches dabei fortlaufend exploriert werden muss. Zu diesem Zweck wird ein SLAM²-Verfahren entwickelt. Bei dieser Erkundung von unbekanntem Terrain und der damit verbundenen Erstellung eines akkuraten 3D-Modells der Umgebung ist eine effektive Navigation sehr wichtig. Derzeit existiert dazu am Institut noch kein Verfahren, um die Umgebung kosten-optimiert erkunden zu können. Diese fehlende Explorationskomponente soll entwickelt und für das SLAM-Verfahren verwendet werden.

Durch ein optimiertes Verfahren können der Fahrweg, die Erkundungszeit sowie der Energieverbrauch reduziert werden. Unbekannte Räume sollen dadurch, unter Erstellung eines akkuraten 3D-Modells, effizient autonom exploriert werden. So können genaue Karten von Räumen schnell erstellt werden, was durch Messungen von Menschen mit Laser-Entfernungsmessern und ähnlichen Werkzeugen sehr aufwendig und ungenau sein kann – insbesondere bei Räumen mit vielen Objekten.

Mithilfe des Explorationsverfahrens kann außerdem Terrain erkundet werden, das für den Menschen zu gefährlich ist. Wenn die örtlichen Gegebenheiten (Bäume, Gebäude, etc.) eine Aufklärung aus sicherer Entfernung, z.B. durch Flugsysteme oder Satelliten, nicht zulassen oder die Sensoren auf die Entfernung zu ungenau sind, ist ein Einsatz vor Ort notwendig. Eine Fernsteuerung von Erkundungsrobotern ist bei entsprechender Abschirmung, z.B. durch dicke Betonwände oder große Distanzen, auch nicht immer möglich oder zu aufwendig. Für diese Aufgabe können so programmierte Roboter eingesetzt werden,

¹Time-of-Flight-Kameras messen über die Laufzeit simultan die Entfernung zu mehreren zusammenhängenden Raumpunkten. Daraus kann anschließend eine Tiefenkarte erstellt werden, in der z.B. über Grauwerte der jeweilige Abstand zu den Pixeln dargestellt wird.

²Simultaneous localization and mapping: Eine Technik, um mit autonomen Robotern eine Karte in unbekannter Umgebung zu erstellen oder aktualisieren während fortlaufender Positionsbestimmung des Roboters.

1 Einführung

die schnell ein hinreichend genaues 3D-Modell der Umgebung aufbauen. Mögliche Einsatzszenarien sind dabei unter anderem die Erkundung von Gebäuden nach einem Unfall mit gefährlichen Stoffen oder von einsturzgefährdeten Gebäuden oder Tunneln nach einem Erdbeben. Dabei können Risse akkurat erfasst und mögliche Gefahren erkannt werden. Auch für Sicherheitsbehörden sind solche Erkundungsverfahren interessant. SWAT³-Einheiten verwenden bereits autonom explorierende Roboter bei Einsätzen um schnell die Umgebung zu erkunden [7]. Mit den so gewonnenen Daten kann das weitere Vorgehen unter Berücksichtigung aller gewonnenen Information besser geplant werden und dadurch die Gefährdung von Leben und Gesundheit beteiligter Personen minimiert werden.

1.2 Aufgabenstellung und Ziele der Arbeit

Durch die effiziente Wahl von nächstbesten Observationspositionen und somit einem optimierten Fahrweg können unbekannte Umgebungen schnell erkundet werden. Die dabei akquirierte Datenmenge der Tiefenmessungen und damit das 3D-Modell wachsen somit schnell an. Es soll dazu eine Explorationskomponente erstellt werden, welche die Kosten (Zeit, Fahrweg, Kollisionen, etc.) während der Exploration in lokalen Umgebungskarten optimiert. Diese Explorationskomponente beschränkt sich hierbei im Wesentlichen auf das Planen der nächsten besten Ansicht der Tiefensensoren. Zunächst sollen bereits erprobte Verfahren recherchiert und analysiert werden. Ein geeignetes Verfahren für die Aufgabe soll ausgewählt und den Anforderungen entsprechend angepasst werden. Das zu entwickelnde Verfahren soll die benötigte Explorationskomponente für ein aktives, autonomes SLAM darstellen, welches parallel entwickelt wird. Implementiert werden kann das Verfahren auf Voxelkarten⁴ in einer bereits existierenden Simulationsumgebung, in der auch die Funktionalität nachgewiesen werden soll. Abschließend soll die Fahrtplanung des momentan verwendeten Lokalisierungsverfahrens, angewendet als Explorationsverfahren, mit dem neu zu entwickelnden Explorationsverfahren verglichen werden.

1.3 Aufbau der Arbeit

In dieser Arbeit werden zunächst in Kapitel 2 bereits bestehende Next-Best-View-Explorationsverfahren vorgestellt, welche den Stand der Technik repräsentieren. Die Gegebenheiten und Voraussetzungen an das Verfahren werden in Kapitel 3 dargelegt und die vorgestellten Verfahren diesbezüglich überprüft. Aus den Ergebnissen dieser Überprüfung wird festgestellt, welches grundlegende Verfahren sich am besten für die Problemstellung eignet und welche Teile dieses oder anderer Verfahren noch adaptiert werden können. Im Weiteren wird in Kapitel 4 aus den übernommenen Ideen ein neues Verfahren entwickelt, welches der Aufgabenstellung entspricht, um kosten-optimiert unbekanntem Raum

³Special Weapons and Tactics, taktische Spezialeinheit innerhalb einer Polizeibehörde

⁴Dreidimensionale Rastergraphik, welche aus Datenpunkten, den Voxeln, besteht.

explorieren zu können. Die Voraussetzungen und Bedingungen, die dieser speziellen Anwendung zugrunde liegen, werden dabei erläutert. Das neue Verfahren wird im Anschluss in Kapitel 5 analysiert und ein Vergleich mit der Fahrplanung der Lokalisierung des bestehenden Verfahrens durchgeführt. Abschließend werden in Kapitel 6 die Ergebnisse zusammengefasst und ein Ausblick gegeben.

2 Überblick über aktuelle Next-Best-View-Explorationsverfahren

Next-Best-View-Verfahren beschäftigen sich damit nach jedem Explorationsschritt¹ die nächst beste Observationsposition für die anschließende Messung zu finden [16]. Die erste Messung findet an einem bekannten, von Menschen ausgewählten, Ort statt. Dadurch wird sichergestellt, dass der Roboter auf passendem Untergrund steht und Bereiche direkt um den Roboter, die möglicherweise außerhalb des Sensorsichtfeldes liegen, keine Kollisionsgefahr darstellen. Alle folgenden Messungen müssen von Orten durchgeführt werden, welche durch die bereits akquirierten Daten vorausgegangener Messungen als geeignete Viewpoints evaluiert wurden. Kriterien zur Auswahl einer geeigneten Observationsposition sind dabei typischerweise die Fahrtkosten und der erwartete Informationsgewinn, welche über entsprechende Kosten-/Nutzenfunktionen ausgewertet werden. Aus den während der Exploration erfassten Daten kann dann eine zusammenhängende Karte erstellt werden. Der allgemeine Explorationsprozess von Next-Best-View-Verfahren nach [1] ist in Abb. 2.1 veranschaulicht.

Es gibt dabei sehr einfache Methoden, welche den Roboter nur entlang von vordefinierten Trajektorien fahren lassen [4, 14]. Dazu werden Trajektorien auf bereits als befahrbar bekanntem Untergrund definiert, worauf sich der Roboter dann ausschließlich bewegen darf. Der Boden, auf dem sich der Roboter bewegen soll, muss also a priori bekannt sein. Anwendung findet dieses Verfahren, wenn z.B. von bereits explorierten Räumen ein detailliertes 3D-Modell erstellt werden soll. Auf diese Methoden wird hier nicht eingegangen, da hier der zu explorierende Raum als unbekannt anzunehmen ist und demnach keine sinnvollen Trajektorien vordefiniert werden können.

Im Folgenden werden die grundlegenden Eigenschaften einiger aktuell verwendeter Next-Best-View-Verfahren erläutert, welche den Stand der Technik repräsentieren.

2.1 Die Sensor-based Random Tree (SRT) Methode

Das Prinzip dieser in [17] vorgestellten Methode liegt darin, den Roboter nur in bereits erkundeten Bereichen zu bewegen um Kollisionen auszuschließen. Von den Sensoren wird dazu vor der Bewegung des Roboters der Bereich um den Roboter erfasst. Die Position, zu der sich der Roboter bewegt, muss innerhalb diesen Bereichs liegen und als

¹Hier: Fahrt zu neuer Observationsposition und Abtastung der Umgebung mit den Sensoren sowie entsprechende Speicherung und Auswertung der erfassten Messdaten

2 Überblick über aktuelle Next-Best-View-Explorationsverfahren

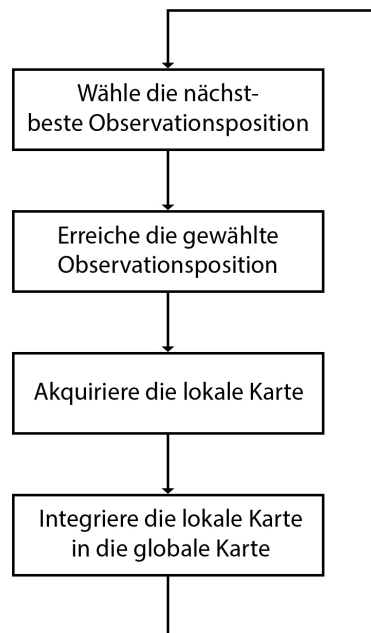


Abbildung 2.1: Explorationsprozess nach [1], eigene Zeichnung

kollisionsfrei gelten.

Im ersten Schritt vermessen die Sensoren, durch ihre Reichweite begrenzt, den Bereich um den Roboter. Die Form dieses Bereichs kann modellhaft als stern- oder (als Spezialfall) kugelförmig angenommen werden. Diese Modellierungen sind für den 2D-Fall (ebener Schnitt durch die Kugel bzw. den Stern) in Abb. 2.2 dargestellt. Im Falle des kugelförmigen Modells begrenzt (neben der Sensorreichweite) bereits ein einziges Hindernis den Radius der den Roboter umgebenden Kugel, auf die Entfernung von dem Roboter zum Hindernis wie in Abb. 2.2(a) dargestellt wird. Bei dem sternförmigen Modell wird die Kugel in gleichgroße Kegel unterteilt. Es werden nur diejenigen Kegel auf die Entfernung zum Hindernis angepasst, in welchen sich das Hindernis befindet. Es verbleiben dabei also Kegel in denen sich der Roboter weiter bewegen kann, als bis auf die Entfernung zum Hindernis, wie in Abb. 2.2(b) dargestellt. Für mehrere detektierte Hindernisse verhält es sich entsprechend. Das Gebiet, welches nun als kollisionsfrei und somit sicher befahrbar eingestuft wurde, heißt **Local Safe Region (LSR)** und ist in den Abbildungen jeweils grün dargestellt. Ist der Roboter an einer Position (Knotenpunkt) angekommen oder wird er initial gestartet, vermisst er zunächst die Umgebung. Die erfassten Umgebungsdaten werden dann in der Datenstruktur **Sensor-based Random Tree (SRT)** gespeichert. Diese Datenstruktur besteht aus den Koordinaten der jeweiligen Knotenpunkte, die der Roboter bereits angefahren hat, mit den dazugehörigen Messdaten.

2.1 Die Sensor-based Random Tree (SRT) Methode

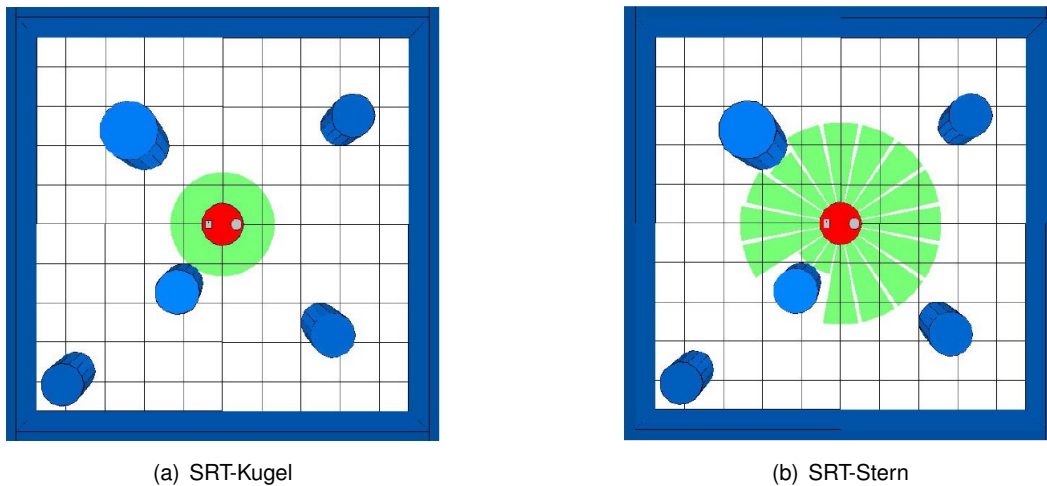


Abbildung 2.2: SRT Modelle aus [17]

Es wird anschließend nach einem nächsten Observationspunkt gesucht, der drei Eigenschaften erfüllen muss. Der Punkt

- (1) muss innerhalb der LSR des aktuellen Knotens liegen,
- (2) darf nicht innerhalb der LSR eines anderen Knotens liegen
- (3) und muss eine Mindestdistanz zur aktuellen Position aufweisen.

Kann ein solcher Punkt nicht gefunden werden, begibt sich der Roboter zum letzten Knotenpunkt zurück („backtracking“).

Um diese Funktionalität zu erreichen, werden folgende Schritte ausgeführt:

Zunächst wird die Explorationsrichtung zufällig gewählt. Die zu fahrende Distanz wird auf das Produkt des Radius der LSR mit einer zuvor festgelegten Konstante zwischen 0 und 1 bestimmt. Dadurch ist gewährleistet, dass der Punkt innerhalb der LSR des aktuellen Knotens liegt (1). Nun wird überprüft, ob Bedingung (2) und (3) erfüllt sind. Ist die Überprüfung der beiden Bedingungen jeweils positiv, bewegt sich der Roboter zu dem Punkt, ansonsten wird die Generation eines möglichen Zielpunktes wiederholt. Dazu wird der Algorithmus neu gestartet und kommt durch die zufällige Wahl der Explorationsrichtung sehr wahrscheinlich zu einem anderen zu evaluierenden Punkt. Ist nach einer festgelegten maximalen Durchlaufanzahl noch kein gültiger Punkt gefunden, begibt sich der Roboter zum letzten Knotenpunkt zurück und beginnt dort erneut mit der Suche nach einem neuen Observationspunkt.

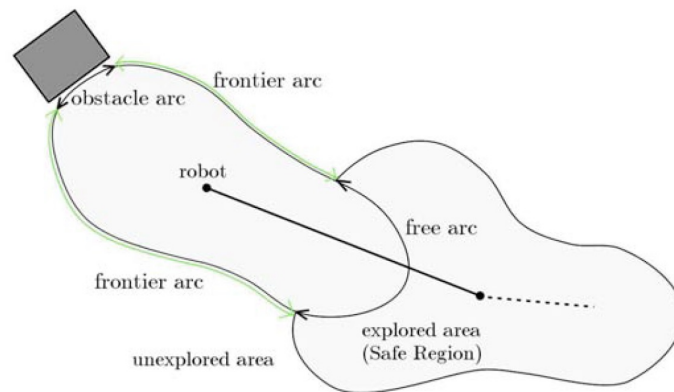


Abbildung 2.3: Definition der obstacle, free und frontier arcs aus [5]

2.2 Frontier-Based SRT

Eine Verbesserung der obigen SRT Methode wird in [5] beschrieben. Es werden Grenzlinien definiert, welche die äußere Begrenzung der LSR darstellen. Diese Grenzlinien werden in drei Typen klassifiziert: 1. obstacle arc („Hindernisbogen“), welcher den Teil, der mit einem Hindernis in Kontakt steht, repräsentiert, 2. free arc („freier Bogen“), der Teil, der innerhalb der LSR eines anderen Knotens liegt und 3. frontier arc („Grenzbogen“), der den verbleibenden Teil, den Übergang zu unbekanntem Terrain, darstellt. In Abb. 2.3 ist dies dargestellt, wobei sich der Roboter von der Position unten rechts nach oben links bewegt hatte und nun von dieser Position aus die arcs bestimmt wurden.

Die Bestimmung des nächsten Knotens erfolgt dann nur durch Auswahl zwischen den frontier arcs, zufällig, mit einer Wahrscheinlichkeit, proportional zu ihrer jeweiligen Länge. Längere frontier arcs werden also mit höherer Wahrscheinlichkeit exploriert, da sich davon ein höherer Informationsgewinn als bei kürzeren frontier arcs versprochen wird. Dadurch, dass nicht in Richtung der anderen arcs exploriert wird, werden Umwege durch häufiges backtracking vermieden, was auch eine zeitlich schnellere Erkundung zur Folge hat. Außerdem wird die Karte der zu explorierenden Umgebung dabei vollständiger, wie in [5] experimentell² ermittelt wurde.

2.3 Frontier-Based Exploration mit Evidence Grid

Bei dem in [6] vorgestellten Ansatz wird der (unbekannte) Raum in Zellen aufgeteilt, welche jeweils eine Besetzungswahrscheinlichkeit wiedergeben.

Zunächst werden alle Zellen auf einen Initialwert gesetzt (z.B. 0,5). Während der Exploration werden die Werte dann entsprechend den Messdaten angepasst. Liegt die

²Simulation in Move3D, einer Software Plattform von KineoWorks (www.kineocam.com)

Tabelle 2.1: Gegenüberstellungsmatrix des AHP

	Kriterium 1	Kriterium 2	Kriterium 3
Kriterium 1	1	8	3
Kriterium 2	1/8	1	1/4
Kriterium 3	1/3	4	1

Besetzungswahrscheinlichkeit einer Zelle über dem Initialwert, gilt der Bereich als besetzt, liegt sie unterhalb, gilt der Bereich als offen und ist sie gleich dem Initialwert, so gilt der Bereich als unbekannt. Jede offene Zelle, an die eine unbekannte Zelle angrenzt, wird als Grenzzelle bezeichnet. Zusammenhängende Grenzzellen werden zu Grenzregionen zusammengefasst, welche wiederum ab einer Mindestgröße (in etwa der Größe des Roboters) als Grenze bezeichnet werden. Nachdem Grenzen innerhalb eines gewissen Bereichs detektiert worden sind, versucht der Roboter zur nächstliegenden, erreichbaren und noch unbesuchten Grenze zu fahren, um dort die nächste Messung durchzuführen.

2.4 Die MCDM Strategie

Die **Multi-Criteria Decision Making** (MCDM) Strategie, die in [2] für die Next-Best-View-Planung Anwendung findet, beschäftigt sich mit der Auswahl eines Observationspunktes aus mehreren Kandidaten unter Berücksichtigung und Gewichtung diverser Kriterien. Innerhalb dieser Strategie gibt es viele Methoden, die angewendet werden können.

Hier soll nur auf den **Analytic Hierarchy Process** (AHP) nach [18] eingegangen werden, da sich dieser am besten für das Problem eignet. Der Prozess ist analytisch, da er alle Abhängigkeiten genau analysiert. Das Vorgehen ist außerdem hierarchisch strukturiert. Es können Kriterien in Gruppen unterteilt werden. Jede Gruppe beeinflusst jeweils nur andere, höher eingestufte Gruppen und wird nur von anderen niedriger eingestuft Gruppen beeinflusst. Dies ist von Vorteil, wenn z.B. die Fahrtkosten weiter unterteilt werden sollen, von der Gesamtbewegung des Roboters, bis hin zur der dazu nötigen Drehung einzelner Zahnräder.

Bei dem Verfahren werden zunächst alle zu berücksichtigenden Kriterien definiert und nach Bedarf gruppiert. Anschließend werden diese Kriterien gegenübergestellt. In einer Matrix wird die Gewichtung jedes Kriteriums mit jedem anderen dargestellt, wie ein Beispiel in Tabelle 2.1 zeigt, wobei die Bedeutung der Werte Tabelle 2.2 zu entnehmen ist. Durch Berechnung der Eigenvektoren können die jeweiligen Prioritäten der Kriterien bestimmt werden. Nach [9] kann der Eigenvektor sehr gut approximiert werden, indem die Einträge jeder Reihe zunächst multipliziert und dann die nte Wurzel gezogen wird. Zum Normalisieren werden die Wurzeln aufsummiert und der Eigenvektor dadurch geteilt. Tabelle 2.3 zeigt dies anhand eines Beispiels.

2 Überblick über aktuelle Next-Best-View-Explorationsverfahren

Tabelle 2.2: Bedeutung der Werte in der AHP Matrix aus [28]

Wert	Bedeutung
1	gleiche Bedeutung der beiden Elemente
3	etwas höhere Bedeutung eines Elements
5	deutlich höhere Bedeutung eines Elements
7	viel höhere Bedeutung eines Elements
9	sehr viel höhere Bedeutung eines Elements
2, 4, 6, 8	Zwischenwerte
$\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{9}$	Reziprokwerte

Tabelle 2.3: AHP Matrix mit berechneten Eigenvektoren aus [9]

	A	B	C	D	nte Wurzel der Produkte der Werte	Eigenvektor
A	1	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{5}$	0,293	0,058
B	3	1	1	1	1,316	0,262
C	9	1	1	3	2,279	0,454
D	5	1	$\frac{1}{3}$	1	1,136	0,226
Gesamt					5,024	1,000

Kriterium A würde nun also mit 5,8 %, Kriterium B mit 26,2%, Kriterium C mit 45,4 % und Kriterium D mit 22,6% gewichtet werden.

Anschließend kann sehr schnell und ohne weiteren großen Rechenaufwand durch die zuvor berechneten Gewichtungsfaktoren eine Entscheidung getroffen werden.

3 Auswahl von geeigneten Verfahren für die Problemstellung

3.1 Gegebenheiten und Anforderungen

3.1.1 Gegebenheiten

- Bei dem Roboter, auf dem das Verfahren implementiert werden soll, handelt es sich um den „**omniRob**“ von Kuka:
 - Länge, Breite, Höhe: 1,20 m, 0,71 m, 0,65 m
 - Gewicht: ca. 270 kg zzgl. Bleigel-Akku 48 V, 80 Ah, ca. 100 kg
 - Antrieb: Mecanum-Räder für omnidirektionale Fahrmanöver
 - Sensorik: 8 ToF-Kameras vom Typ O3D100
 - Anforderungen an den Untergrund: Eben, da der Roboter keine Federung hat.
- Zum Mapping wird eine **Voxelkarte** in dem **Probabilistic Dynamic Octree Space** verwendet. Der Dynamic Octree Raum, wie in [3] beschrieben, besteht aus Voxeln, welche bei gleichem Inhalt zu einem Octree, bestehend aus acht solchen Voxeln, zusammengefasst werden. Diese acht Voxel müssen so angeordnet sein, dass aus den Voxel-Würfeln ein Octree-Würfel entsteht und kein Quader. Ein Octree Würfel besitzt also eine Kantenlänge, die gleich der zweifachen Kantenlänge der eingeschlossenen Voxel ist. Acht homogene Octrees können wiederum zu einem größeren Octree zusammengefasst werden. Somit sinkt der Speicherbedarf, insbesondere bei Karten mit vielen homogenen Objekten, sehr stark. Durch die weitere Verwendung des Probabilistic Space, welcher von [26] eingeführt wurde, wird für jedes Voxel eine Besetzungswahrscheinlichkeit gespeichert. Dadurch kann die Unsicherheit der Sensoren bei Messungen implementiert werden.
Es gibt neben den Voxelkarten auch andere Karten, die zum Mapping benutzt werden können, wie z.B.
 - Tiefendaten
Eine Liste von Tripeln, bestehend aus dem Standpunkt des Sensors, der Richtung, in die gemessen wurde, sowie der gemessenen Tiefe.
 - Punktwolken
Eine Liste von kartesischen 3D-Koordinaten. Sie lässt sich direkt aus den Tiefendaten erstellen.

3 Auswahl von geeigneten Verfahren für die Problemstellung

- Topologische Karten
Eine topologische Repräsentation der Umwelt, bestehend aus Knoten, Kanten und Gewichten. Ein Knoten stellt jeweils einen diskreten Ort dar, Kanten beschreiben die Wege zwischen den Knoten und die Gewichte sind ein Maß für die Länge der Wege [25]. Knoten werden in der Regel für wichtige Positionen, wie z.B. einen Eingang, einen Ausgang, eine Gefahrenquelle, etc. erstellt.
- Selbstorganisierende Karten
Selbstorganisierende Karten wurden von Teuvo Kohonen entwickelt und in [10, 11] vorgestellt. Es handelt sich dabei um ein spezielles künstliches neuronales Netz, das sich, nach dem Vorbild des menschlichen Gehirns, selbst organisiert.

Punktwolken und Tiefendaten sind für diese Anwendung ungeeignet, da die Sensoren viele Daten produzieren, welche in diesen Datenstrukturen sehr viel Speicherplatz belegen würden. Sowohl topologische, als auch selbstorganisierende Karten sind geeignet, um große Gebiete mit wenig interessanten Bereichen darzustellen, jedoch weniger, um die Geometrie der Umgebung akkurat wiederzugeben, was hier gefordert wird.[22]

Die verwendete Voxelkarte eignet sich demnach hierfür am besten, da sie, in dem Probabilistic Dynamic Octree Space implementiert, Umgebungsdaten speicherplatzeffizient und akkurat speichern kann. Die Qualität der Karte lässt sich über die Wahl der Raster-Auflösung (Kantenlänge eines Voxels) problemlos einstellen. Es wird eine Auflösung von 10 cm gewählt und darauf geachtet, dass der Programmcode so geschrieben ist, dass eine Änderung der Auflösung keinen Einfluss auf die Funktionalität hat, indem nicht mit absoluten Werten gearbeitet wird. Die Auflösung lässt sich somit nachträglich verändern, ohne dass weitere Anpassungsmaßnahmen nötig sind.

Eine Alternative zu dem Dynamic Octree Space stellt die in [8] vorgestellte Octomap dar.

Diverse 2D-Mapping Methoden wurden ausführlich in [21] beschrieben und teilweise in [26] adaptiert und verwendet. Diese Methoden kommen aufgrund der erforderlichen Dreidimensionalität nicht infrage.

- Es existiert eine vom Institut entwickelte Simulationsumgebung, in der getestet werden kann. Dort ist ein Modell des zu programmierenden Roboters sowie der Testumgebung vorhanden. Es werden alle Voxel der Karte dargestellt, welche sich in den ersten fünf Voxelbenen über dem Boden befinden.

Die Farben der dargestellten Voxel haben folgende Bedeutung:

- hellgrau, hoch transparent: unbekannt
- grau bis schwarz, leicht transparent bis intransparent: besetzt
Durch die Graustufen wird die Besetzungswahrscheinlichkeit ausgedrückt. Je dunkler und weniger transparent ein Voxel, desto wahrscheinlicher ist es be-



Abbildung 3.1: Simulationsumgebung

setzt. Dadurch kann die Messunsicherheit dargestellt werden, welche sich durch mehrfaches Messen derselben Voxel reduziert.

- vollständig transparent: frei

Das Modell der Umgebung (Wände, Mobiliar, etc.) wird zur Visualisierung durch die Simulationsumgebung automatisch auf die Voxelkarte gesetzt. Dies geschieht, da der Simulation die Position des Roboters bekannt ist. Im praktischen Betrieb außerhalb der Simulation muss sich der Roboter erst durch Messungen lokalisieren. Dies ist außerdem nur bei bereits bekannten Karten, wie der hier verwendeten Testumgebung, welche auch real existiert, möglich.

3.1.2 Allgemeine Anforderungen

Es gibt drei grundlegende Anforderungen an ein Next-Best-View-Verfahren, die in [1] formuliert wurden.

- Zum einen muss der Algorithmus effizient sein, was bedeutet, dass eine Erkundung möglichst schnell stattfindet. Dies impliziert einen möglichst kurzen zurückzulegenden Weg, da die Bewegungsgeschwindigkeit nicht verändert wird. Durch den

3 Auswahl von geeigneten Verfahren für die Problemstellung

kürzeren Fahrweg wird auch der Energieverbrauch reduziert, falls dieser Weg nicht durch ungünstigeres Terrain führt. Ungünstig wäre in diesem Fall Untergrund mit sehr hoher Reibung, loser Untergrund, welcher erhöhten Schlupf des Antriebs zur Folge hat, sowie Steigungen und Gefälle, wobei dabei die potentielle Energie nicht genutzt werden kann, da gebremst werden muss.

- Die Erstellung der Karte muss akkurat sein, sodass die entstehende Karte präzise und vertrauenswürdig ist. Da die Erstellung einer akkuraten Karte jedoch nicht Bestandteil dieser Arbeit ist, kann die Akkuratessse als gegeben angenommen werden.
- Außerdem soll das Verfahren adaptierbar sein, also auch in anderen, als der Testumgebung funktionieren bzw. einfach daran angepasst werden können.

Eine weitere allgemeine Anforderung, die in [20] erwähnt wird, ist, dass der Roboter die Exploration nach ausreichender Erkundung autonom abbricht.

3.1.3 Spezielle Anforderungen

Spezielle Anforderungen für diese Problemstellung, welche aus den genannten Gegebenheiten resultieren, sind, dass

- das Verfahren gut auf Voxelkarten implementiert werden kann
- und für die Verwendung von Tiefensensoren, wie die verwendeten ToF-Kameras geeignet ist.

3.2 Überprüfung der recherchierten Verfahren bzgl. den Anforderungen

Wie sich durch Versuche in [5] gezeigt hat, ist die Frontier-Based SRT mit Verwendung einer sternförmigen LSR das effizienteste der vorgestellten SRT Verfahren.

Die Frontier-Based Exploration mit Evidence Grid verfolgt den gleichen Grundgedanken, dass stets in Richtung unbekanntes Gebietes exploriert wird. Da mit Voxelkarten gearbeitet wird, kommt die zellenweise Aufteilung des Gebietes, der Datenstruktur, im Gegensatz zu den vorgestellten SRT Methoden, sehr entgegen und könnte leicht adaptiert werden. Auch die Verwendung des Evidence Grids passt sehr gut zu der verwendeten Voxelkarte, welche bereits Besetzungswahrscheinlichkeiten enthält. Da die ToF-Kameras nicht exakt messen und somit eine Wahrscheinlichkeitsverteilung für jeden Bildpunkt wiedergeben, bietet sich die Verwendung dieses Evidence Grids an bzw. ist bereits implementiert. Mit diesen Wahrscheinlichkeiten kann dann im weiteren Verfahren gerechnet werden und es muss nicht sofort entschieden werden, ob eine Zelle als definitiv besetzt oder unbesetzt

3.2 Überprüfung der recherchierten Verfahren bzgl. den Anforderungen

gilt. Es können so Wahrscheinlichkeiten durch mehrmaliges Messen entsprechend angepasst und die Zuverlässigkeit der Karte erhöht werden.

Bei der Auswahl zwischen den Kandidaten für die nächstbeste Observationsposition wird bei diesem Verfahren stets die nächstliegende Grenze angefahren, um dort die folgende Messung durchzuführen. Dieses Verhalten ist sehr unflexibel und kann effizienter gestaltet werden. Nicht immer ist die Grenze mit dem geringsten Abstand zum Roboter die beste Wahl, da als Kriterium nur die Fahrtkosten betrachtet werden und z.B. der erwartete Informationsgewinn unbeachtet bleibt. Zur Verbesserung soll dazu prinzipiell auf den Analytic Hierarchy Process der MCDM Strategie zurückgegriffen werden – je nach Anzahl der Kriterien kann auch eine einfache Gewichtung ausreichend sein. Es ist somit möglich, verschiedene Kriterien, wie die Fahrtkosten und den erwarteten Informationsgewinn, zu berücksichtigen und die Gewichtung dieser Kriterien den jeweiligen Bedingungen anzupassen (Adaptierbarkeit).

Die Arbeit wird folglich auf der Frontier-Based Exploration mit Evidence Grid aufbauen und eine Gewichtung von entsprechenden Kriterien, bei der Auswahl zwischen mehreren möglichen NBV-Punkten, berücksichtigen.

4 Implementierung und Anpassung des ausgewählten Verfahrens

4.1 Voraussetzungen

Folgende Voraussetzungen sind erfüllt oder können als erfüllt angenommen werden:

- Die Odometrie¹ ist als fehlerfrei anzunehmen. Mit der Fehlerkorrektur, im Rahmen einer SLAM-Implementierung, beschäftigt sich eine parallel am Institut laufende Arbeit.
- Die Umgebung ist statisch. Es wird also davon ausgegangen, dass sich während des Explorationsvorgangs die Umgebung nicht ändert, z.B. durch Lebewesen die sich in dem Gebiet befinden und bewegen.
- Ein Mapping ist bereits vorhanden. Die Berechnung der Besetzungswahrscheinlichkeit einzelner Voxel mithilfe der Tiefenmessungen erfolgt nach [26]. Im Folgenden wird direkt über freie und besetzte Voxel gesprochen und nicht auf die Auswertung der Wahrscheinlichkeiten eingegangen.
- Der Untergrund befindet sich auf einer konstanten Höhe. Steigungen sind nicht vorhanden.

4.2 Ausgangslage

Der Roboter befindet sich zu Beginn der Exploration mittig in einer Voxelkarte, welche groß genug ist, um den gesamten zu explorierenden Raum darin erfassen zu können.

Den Voxeln können Werte zwischen 0 (frei) und 255 (besetzt) zugewiesen werden, welche die Besetzungswahrscheinlichkeit repräsentieren [23]. Der spezielle Wert 127 wird benutzt, um unbekanntem, noch nicht exploriertem Raum darzustellen.

Es gibt in der Voxelkarte zwei Ebenen, bei $z = 0 \cdot resolution$, und $z = 1 \cdot resolution$, denen eine besondere Bedeutung zukommt, wobei *resolution* die Auflösung, also die minimale Kantenlänge eines Octrees, ist.

Bei $z = 1 \cdot resolution$ befindet sich der Boden, worin stets alle Voxel besetzt sind und nicht verändert werden können.

¹Positionsbestimmung eines mobilen Systems anhand der Daten seines Antriebssystems

4 Implementierung und Anpassung des ausgewählten Verfahrens

Der Roboter vermisst den Raum in einer Höhe von 1,00 m bis zum Boden. Jedes detektierte Hindernis, egal in welcher Höhe, wird als potentielles Kollisionsobjekt betrachtet. Der Roboter weißt bei montiertem Greifarm auf der Oberseite eine Höhe von über 1,00 m auf. Dieser Fall wird hier nicht betrachtet. Es existieren dafür Stereokameras am Greifarm, welche den Bereich ab einer Höhe von 1,00 m vermessen.

Für die Navigation werden zur Reduktion des Rechenaufwandes von $O(n^3)$ auf $O(n^2)$, alle Hindernisse in die sog. collision plane projiziert. Diese unterste Ebene, welche sich unterhalb des Bodens bei $z = 0$ anschließt wird collision plane genannt, da dort alle Kollisionsobjekte eingetragen sind und zur Navigation abgefragt werden können (2D-Navigation in 3D-Karte). Dazu werden an jeder neu explorierten² x, y Position der darüber liegende Stapel an Voxeln in positive z -Richtung ab $z = 3 \cdot resolution$ ³ durchlaufen. Wird dabei mindestens ein besetztes Voxel gefunden, wird das entsprechende Voxel in der collision plane auf den collision state gesetzt. Falls kein besetztes Voxel gefunden wurde, muss sich also der Zustand mindestens eines Voxels von unbekannt auf frei geändert haben, da eine Veränderung mindestens eines Voxels Voraussetzung für diesen Funktionsaufruf ist. Für diesen Fall wird das Voxel in der collision plane auf den free state gesetzt. Initial befinden sich alle Voxel in der Ebene in dem unknown state.

Alle über dem Boden liegenden Voxel sind zunächst unbekannt und werden dann entsprechend der Messungen angepasst.

Um den Roboter Mittelpunkt wird zu Beginn ein Bereich von 2,00 m x 2,00 m x 1,00 m als frei angenommen. Dies ist notwendig, da der Roboter zum einen nur in freiem Raum stehen kann und zum anderen um den Roboter ein gewisser Raum nicht von den Sensoren erfasst werden kann. Der Roboter wird von Menschen an die Startposition gebracht und es wird durch Inaugenscheinnahme sichergestellt, dass dieser Bereich frei ist.

Alle möglichen, sowie initialen Zustände der Voxel in den jeweiligen Ebenen sind in Tabelle 4.1 nochmals dargestellt.

4.3 Detektion möglicher nächster Observationspunkte

Das Vorgehen, um geeignete Kandidaten nächster Observationspunkte zu finden, wurde wie folgt gewählt. Zunächst führt der Roboter einen Scan der Umgebung mit den

²Wenn mindestens ein Voxel an einer z Position größer $3 \cdot resolution$ seinen Wert ändert

³ Es hat sich herausgestellt, dass es sinnvoll ist bei $z = 3 \cdot resolution$ zu beginnen, um Messfehler bei $z = 2 \cdot resolution$ an den Ecken zwischen dem Boden und einem darauf stehenden Objekt zu berücksichtigen. Die Messung einer Kollision mit den Strahlen der ToF-Kameras liefert eine abnehmende Besetzungswahrscheinlichkeit um den Kollisionspunkt. Da bei Ecken die Voxel am Boden mit den schräg darüber liegenden Voxeln eines Objektes so dicht zusammen liegen und sich die Besetzungswahrscheinlichkeiten (nichtlinear) addieren, werden dazwischenliegende Voxel, sowie teilweise auch weitere benachbarte, meist fälschlicherweise als besetzt angenommen - dies ist u.A. in Abb. 4.1 gut zu erkennen. Durch diese Vernachlässigung der $z = 3 \cdot resolution$ Ebene werden allerdings eventuelle sich am Boden befindliche Objekte mit einer erfassten Höhe von weniger als einem Voxel nicht berücksichtigt. Da solche Objekte bei dem konkreten Anwendungsfall ausgeschlossen werden können, stellt dies hier kein Problem dar. Unter anderen Voraussetzungen muss das Verfahren möglicherweise angepasst werden.

Tabelle 4.1: Zustände der Voxel in den Ebenen

Ebene	mögliche Zustände	initial
collision plane ($z = 0$)	unknown, free und collision state	unknown state
Boden ($z = \text{resolution}$)	besetzt	besetzt
Darüber ($z > \text{resolution}$)	frei (0-126), unbekannt (127), besetzt (128-255)	unbekannt

ToF-Kameras durch. Mit den akquirierten Tiefendaten wird die Voxelkarte entsprechend aktualisiert. Von besonderem Interesse sind dabei nun die Grenzen zwischen als kollisionsfrei gemessenem Gebiet und unbekanntem Gebiet. Diese Grenzen müssen jedoch für den Roboter von seinem aktuellen Standpunkt aus auch erreichbar sein. Wie in Abb. 4.1 links unten zu erkennen ist, können die ToF-Kameras beispielsweise über eine niedrige Mauer messen und dahinter so eine Grenze detektieren. Der Roboter kann dort jedoch nicht gefahrlos hinfahren, da kein als kollisionsfrei gemessener Weg dorthin existiert. Um die Erreichbarkeit zu gewährleisten, werden ausgehend vom Roboter alle zusammenhängenden free state Voxel rekursiv durchlaufen und währenddessen als erreichbar markiert. Dabei muss darauf geachtet werden, dass stets eine Mindestbreite vorhanden ist, sodass der Roboter die Voxel auch erreichen kann und keine zu engen Durchfahrten vorhanden sind. Ohne diese Einschränkung würde die Rekursion, selbst bei ein Voxel breiten Durchfahrten, durchlaufen und dahinter könnten (große) zusammenhängende Bereiche entstehen, die ebenfalls als erreichbar markiert würden. Die erreichbaren Voxel nach der Initialmessung sind in Abb. 4.2(a) dargestellt (alle dargestellten Voxel, egal welche Farbe), wobei das rote Voxel den Roboter Mittelpunkt und den Beginn der Rekursion darstellt. Zur Grenzbestimmung werden nun nur erreichbare free state Voxel durchlaufen und auf Nachbarschaft zu unknown state Voxel geprüft. Es ist aus Performance Gründen wichtig, diese Reihenfolge einzuhalten und nicht zuerst alle, auch nicht erreichbare, Grenzvoxel zu detektieren und anschließend die nicht erreichbaren wieder zu entfernen. Insbesondere bei großen Karten ist es viel schneller, nur die erreichbaren Voxel auf die Eigenschaft der Nachbarschaft zu unbekanntem Voxeln zu überprüfen, als alle Voxel der Karte. Die nun gefundenen Voxel werden als Grenzvoxel bezeichnet und sind in Abb. 4.2(b) dargestellt. In Abb. 4.1 ist der Bereich rot markiert, in dem sie sich befinden.

Im nächsten Schritt werden benachbarte Grenzvoxel zu Grenzregionen, welche jeweils aus einer Liste zusammenhängender Grenzvoxel bestehen, zusammengefasst. Dies geschieht unter der Voraussetzung einer Mindestanzahl an Grenzvoxeln. Dadurch werden sowohl fehlerhafte Messungen, welche einzelne Grenzvoxel hervorbringen, behoben, als auch für den Roboter zu enge Durchfahrten aussortiert. Für jede Region wird anschließend ein Anfahrpunkt berechnet. Zu diesem Zweck werden die Grenzvoxel zunächst um einen Sicherheitsabstand in Richtung bekannten Gebietes verschoben. Dies ist zwingend erforderlich, da der Roboter eine gewisse Ausdehnung hat und somit sein Mittelpunkt

4 Implementierung und Anpassung des ausgewählten Verfahrens

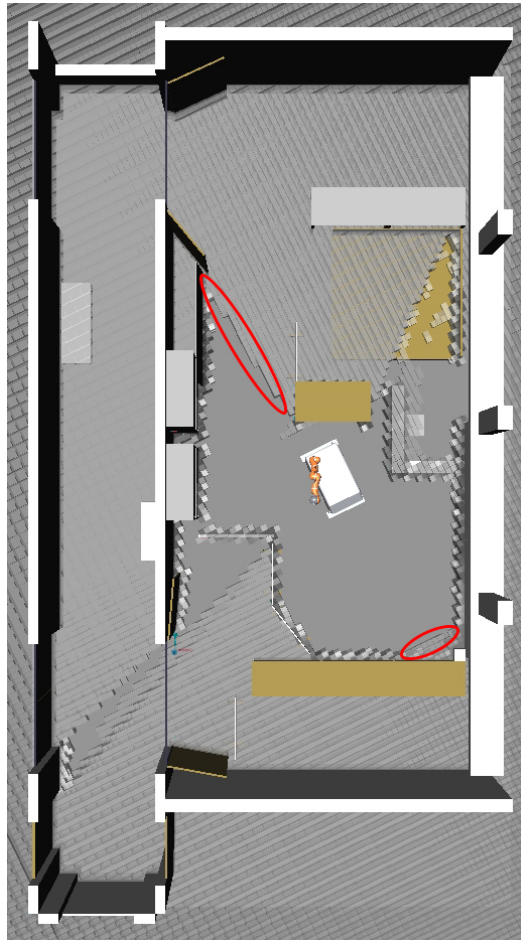


Abbildung 4.1: Ausgangssituation nach initialer Messung. Bereiche in denen Grenzvoxel auftreten sind rot markiert.

entsprechend weit vor den Grenzvoxeln stehen bleiben muss. Dazu werden zunächst die Eckpunkte der Grenzlinien detektiert und durch einen Vektor verbunden. Anschließend wird ein Vektor senkrecht auf den Verbindungsvektor, welcher in Richtung bekannten Gebiets zeigt, berechnet und die Grenzvoxel der Grenzlinie um eine definierte Länge in Richtung des Vektors verschoben. Von diesen verschobenen Voxeln wird der Mittelpunkt berechnet, welcher den Anfahrpunkt darstellt. Dieses Vorgehen ist sehr schnell, jedoch ist nicht garantiert, dass bei der Anfahrt dieses Punktes keine Kollision mit anderen Hindernissen auftritt, da lediglich zu den Grenzvoxeln ein Sicherheitsabstand vorhanden ist. Die um den Sicherheitsabstand verschobenen Anfahrpunkte, müssen deshalb nochmals evaluiert werden. Es wird dazu geprüft, ob ein Kreis um den Anfahrpunkt mit einem

4.3 Detektion möglicher nächster Observationspunkte

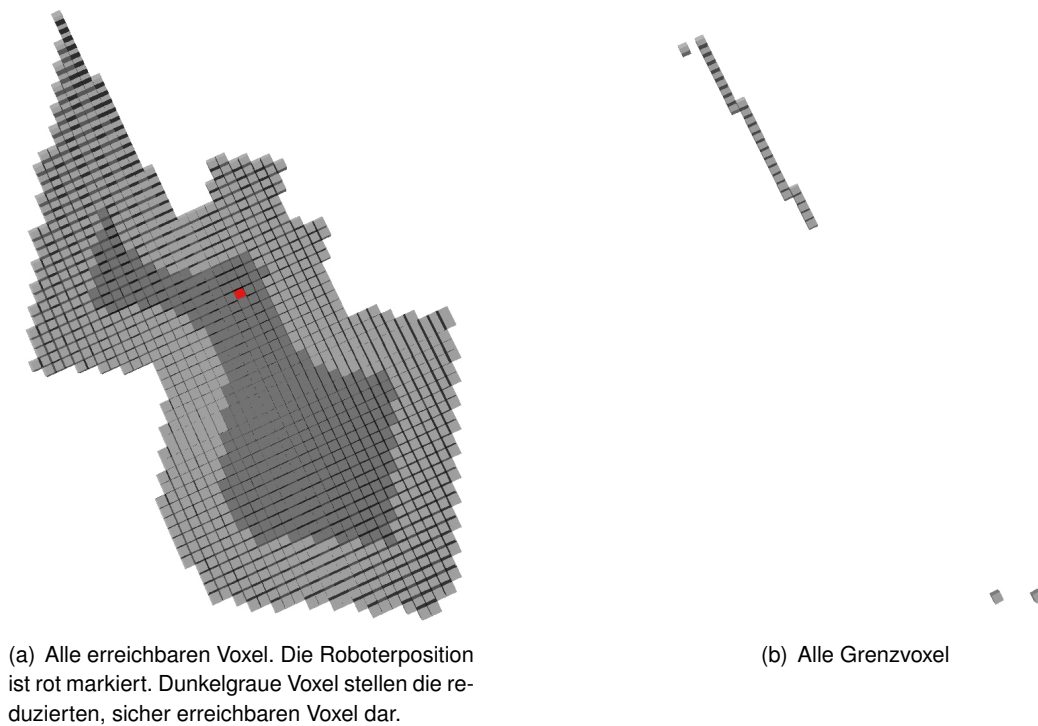


Abbildung 4.2: Erreichbare Voxel sowie Grenzvoxel nach initialem Scan

Radius, der den gesamten Roboter inklusive einem Mindestabstand der Sensoren⁴ zu unbekanntem Gebiet, einschließt, nur freie Voxel enthält. Ist dies nicht der Fall, so ist eine weitere Verschiebung des Anfahrpunktes notwendig.

Dazu werden zunächst alle Voxel aus der Liste der erreichbaren Voxel detektiert, welche mindestens ein Voxel als Nachbar haben, welches nicht in der Liste enthalten ist. Somit erhält man alle Voxel die den Rand bilden. Diese Voxel werden nun durchlaufen und um jedes Randvoxel mit einem definierten Radius alle darin liegenden Voxel aus der Liste der erreichbaren Voxel entfernt. Der Radius wird dabei so groß gewählt, wie er bei dem, den Roboter inklusive dem Mindestabstand der Sensoren umschließenden, Kreis sein müsste. In Abb. 4.2 sind alle erreichbaren Voxel nach der Initialmessung dargestellt und diejenigen Voxel, die durch das Verfahren entfernt werden, hellgrau gekennzeichnet. Es verbleiben also die dunkelgrauen Voxel als mögliche nächste Observationsposition. Ein neuer Anfahrpunkt wird so gewählt, indem der Punkt innerhalb der reduzierten erreichbaren Voxel berechnet wird, welcher den geringsten euklidischen Abstand zu dem zuvor berechneten Anfahrpunkt hat.

⁴Die ToF-Kameras sind in einer gewissen Höhe am Roboter montiert und der Messwinkel Richtung Boden ist kleiner 90 Grad, sodass der Bereich unmittelbar um den Roboter nicht vermessen werden kann.

4 Implementierung und Anpassung des ausgewählten Verfahrens

Der Sicherheitsabstand wird stets so gewählt, dass der Roboter, egal in welcher Rotation, kollisionsfrei den Punkt anfahren kann. Außerdem wird der Mindestabstand der ToF-Kameras zu messendem Gebiet beachtet, um keine unbekanntes Voxel unmittelbar um den Roboter zu hinterlassen. Dies könnte dazu führen, dass der Roboter mögliche Anfahrpunkte nicht anfahren kann, da es nicht erlaubt ist, durch unbekanntes Gebiet zu fahren. Es wäre möglich, den Roboter so zu rotieren, dass der Abstand verringert werden kann. Bei Robotern welche sich in Breite und Länge deutlicher unterscheiden oder in Umgebungen mit sehr engen Durchfahrten kann dies von Vorteil oder sogar notwendig sein. In diesem Fall wird das nicht gemacht, da sich der Roboter, aufgrund der Sensoranordnung, zur vollständigen Messung auf der Stelle um seinen Mittelpunkt dreht und somit nur der Abstand des entferntesten Punktes am Roboter ausschlaggebend ist.

4.4 Auswahl des besten aller detektierten Anfahrpunkte

Wie bereits aus den Abbildungen 4.1 - 4.3 ersichtlich wurde, funktioniert die Next-Best-View-Planung durch die Bestimmung von Grenzregionen in diesem Fall sehr gut, da an der Startposition nur die eine, die Mindestgröße überschreitende, Grenzregion detektiert wird, welche auch für eine schnelle Exploration gewählt werden muss. Die drei einzelnen Grenzvoxel werden aufgrund der resultierenden Grenzregionsgröße, welche die Mindestgröße nicht erreicht, verworfen. Algorithmen, welche Fahrtrichtungen zufällig wählen, würden den Roboter mit großer Wahrscheinlichkeit zunächst im unteren Bereich des Raumes herumfahren lassen, bevor der Roboter die Durchfahrt zwischen Schrank und Tisch findet und neue Tiefendaten akquirieren kann.

Im weiteren Verlauf werden an jeder NBV-Position stets mehrere Grenzregionen mit ihren jeweiligen Anfahrpunkten generiert, welche evaluiert werden müssen. Um dort den am besten geeigneten, der berechneten Anfahrpunkte, zu finden, gewichtet eine Kosten-/Nutzenfunktion verschiedene Kriterien. Implementiert werden die Kriterien **Fahrtkosten**, welche sich aus der Länge des zurückzulegenden Weges ergeben, sowie **erwarteter Informationsgewinn**. Für die Gewichtung zwischen den beiden Kriterien wird der AHP nicht angewendet, da dieser erst ab mindestens drei Kriterien sinnvoll anzuwenden ist.

Da für die Implementierung die Fahrtplanung zwischen zwei Punkten erst noch angepasst werden muss, wird der Roboter derzeit nur von einem Punkt auf einen anderen Punkt in der Simulation gesetzt, ohne den Weg dahin zu berücksichtigen. Für die absoluten Fahrtkosten wird demnach einfach der euklidische Abstand der beiden Punkte gewählt. Die relativen Fahrtkosten berechnen sich aus dem Verhältnis der absoluten Fahrtkosten und der längst möglichen Strecke zwischen zwei Punkten. Diese längst mögliche Strecke wird aus der maximalen Sensorreichweite, abzüglich des Sicherheitsabstandes, bestimmt.

Bei der Bestimmung des zu erwartenden Informationsgewinns spielt der Begriff der Entropie, wie er in [24] definiert wurde, eine Rolle. Demnach ist Entropie der mittlere Informationsgehalt eines Zeichens über einem endlichen Alphabet. Sie quantifiziert die Häufigkeitsdichteverteilung von Zeichen. Je unwahrscheinlicher also das Messen einer bestimmten

4.5 Problembehandlung falls kein Anfahrpunkt bestimmbar

Information, desto höher ihre Entropie. Verwendung für die Exploration findet das Verfahren in [13, 12]. Die Entropie H eines Voxels in Abhängigkeit der Besetzungswahrscheinlichkeit p wird dabei wie folgt berechnet:

$$H_{\text{Voxel}}(p) = - \underbrace{p \log(p)}_{\text{besetzt}} - \underbrace{(1-p) \log(1-p)}_{\text{frei}} \quad (4.1)$$

Der gesamte erwartete Informationsgewinn ergibt sich dann aus der Summe aller Entropien der messbaren Voxel.

Bei dieser Exploration ist die Erkundung möglichst vieler unbekannter Voxel von höchster Priorität. Das erneute Scannen von Objekten, um die Modellierung zu verbessern, ist hier nicht die Hauptaufgabe, weshalb ein anderer Ansatz gewählt wird.

Zur Ermittlung des erwarteten Informationsgewinns wird von den jeweiligen Positionen betrachtet, wie viele Voxel für die ToF-Kameras sichtbar sind und ob diese Voxel frei oder unbekannt sind. Dazu werden Sensorbeams⁵ simuliert, welche die ToF-Kameracharakteristik und -anordnung nachbilden, ausgewertet. Nicht sichtbare Voxel wären solche, welche hinter einem Hindernis liegen, welches der Beam nicht durchdringt. Freie Voxel bringen keine neue Information⁶. Demnach errechnet sich der relative erwartete Informationsgewinn aus der Anzahl der unbekannt messbaren Voxel, im Verhältnis zu der Gesamtzahl der Voxel in Messreichweite der ToF-Kameras.

4.5 Problembehandlung falls kein Anfahrpunkt bestimmbar

Falls mit dem hier vorgeschlagenen Verfahren kein Anfahrpunkt gefunden wird, bleibt der Roboter stehen und beendet die Exploration. Während der Fahrt in der Testumgebung tritt der Fall erst nach vollständiger Erkundung auf, da die Sensorenreichweite groß genug ist, um die Räume der Testumgebung auf dem gewählten Weg zu vermessen, ohne dazu zurückfahren zu müssen. Da dieses Verfahren jedoch auch in anderen Umgebungen funktionieren bzw. leicht adaptiert werden können soll, sei auf Problembehandlungsmöglichkeiten hingewiesen. Falls der Fall eintritt, dass vor ausreichender Exploration kein Anfahrpunkt mehr gefunden wird, werden im Folgenden zwei Lösungsvorschläge erläutert. Eine einfaches Verfahren ohne hohen Rechenaufwand, sowie eines, das mehr Rechenaufwand benötigt, jedoch den Fahrweg effizienter gestaltet.

Die **erste**, einfache und zeit-effiziente **Möglichkeit** ist, dass einmal berechnete Anfahrpunkte gespeichert bleiben und nicht nach der Bewegung zum nächsten Observationspunkt verworfen werden. Durch die implementierte Kostenfunktion wird der Fahrweg zu diesen Anfahrpunkten entsprechend gewichtet und sollte die allgemeine Explorationsrichtung nicht groß behindern.

⁵Lichtstrahlen der ToF-Kamera zur Entfernungsbestimmung der Reflexionspunkte

⁶Da im Hintergrund mit Besetzungswahrscheinlichkeiten gerechnet wird, bringen sie genau genommen insofern doch Information, als sich neue Besetzungswahrscheinlichkeiten innerhalb freier Voxel einstellen können. Dies soll hier jedoch gem. den Voraussetzungen vernachlässigt werden.

4 Implementierung und Anpassung des ausgewählten Verfahrens

Der Grund dafür, dass dies nicht generell so gehandhabt wird, liegt darin, dass sich der erwartete Informationsgewinn für diesen Punkt durch Messungen von anderen Punkten aus ändern kann. Es könnten alle unbekannt messbaren Voxel dieses Anfahrpunktes von anderer Stelle bereits gemessen werden und der erwartete Informationsgewinn für den Punkt wäre gleich Null und eine Anfahrt somit sinnlos. Inwieweit sich der erwartete Informationsgewinn tatsächlich ändert, hängt von vielen Faktoren wie z.B. der Umgebung und der Sensorreichweite ab. Ob es sich für den speziellen Anwendungsfall jeweils lohnt ist spezifisch zu testen. Der große Vorteil bei dieser Methode ist, dass keine zusätzlichen Berechnungen durchgeführt, sondern lediglich bereits berechnete Anfahrpunkte gespeichert werden müssen.

Als Erweiterung dieser Methode ist die Aufnahme eines zusätzlichen Kostenkriteriums möglich. Im einfachen Fall könnte das Kriterium die Anzahl durchgeführter Messungen nach Generierung dieses Anfahrpunktes repräsentieren. Genauer müsste man noch beachten, ob diese Messungen, Voxel im Messbereich des Anfahrpunktes betreffen. Dies wäre jedoch mit entsprechendem Rechenaufwand verbunden und würde den großen Vorteil dieser Methode, dass wenig Berechnungen durchgeführt werden müssen, stark reduzieren.

Für den Fall, dass erneut kein Anfahrpunkt zur Verfügung steht und noch nicht davon ausgegangen werden soll, dass somit das Gebiet vollständig erkundet ist, gibt es eine zusätzliche Erweiterungsmöglichkeit dennoch einen Anfahrpunkt zu generieren. Dies bietet sich an, wenn das Modell besonders akkurat werden soll und ein schneller Abschluss der Exploration nicht im Vordergrund steht. Dazu wird die Minimalgröße heruntersetzt, unter der Grenzregionen verworfen werden. Es wird dadurch auch in Richtung kleiner evtl. unpassierbarer Grenzen exploriert. Des Weiteren werden bei sehr geringer Mindestgröße auch Bereiche mit Messfehlern, wodurch manche Voxel nicht korrekt angepasst wurden und noch fälschlicherweise als unbekannt gelten, als Anfahrpunkte gewählt. Somit wird die Qualität der Karte in diesem Bereich, der offensichtlich nicht akkurat vermessen wurde, verbessert. Nach dem Heruntersetzen der Minimalgröße müssen die Messungen erneut durchgeführt und evaluiert werden.

Die **andere Möglichkeit**, bei der nicht direkt auf Anfahrpunkte vergangener Messungen zurückgegriffen wird, ist Folgende. Es müssen dazu die angefahrenen Observationspunkte, sowie die jeweilige Anzahl berechneter Anfahrpunkte (also auch nicht ausgewählter Anfahrpunkte), chronologisch während der Exploration gespeichert werden. Wird an einer Position kein Anfahrpunkt mehr gefunden, bewegt sich der Roboter zu der letzten Position, an der mehr als ein Anfahrpunkt generiert wurde. Dort wird der normale Explorationsalgorithmus fortgeführt, bis erneut kein Anfahrpunkt gefunden wird. Dadurch werden zuvor nicht ausgewählte Anfahrpunkte erneut geprüft und ggf. angefahren. Ein Heruntersetzen der Minimalgröße für Regionen ist bei dieser Problembehandlung ebenso möglich. Das Ende der Exploration ist erreicht, wenn der Roboter an seiner Initialposition angekommen ist und dort kein Anfahrpunkt mehr gefunden werden kann.

Der Vorteil dieser Methode ist, dass der Roboter keine bzgl. des zu erwartenden Informationsgewinns unrentablen Punkte anfährt. Nachteilig an der Methode ist der erhöhte

4.5 Problembehandlung falls kein Anfahrpunkt bestimmbar

Rechenaufwand, da Anfahrpunkte erneut berechnet werden müssen.

Welche Methode die geeigneter ist, hängt von den jeweiligen Anforderungen und Voraussetzungen ab und kann pauschal nicht beantwortet werden.

5 Experimentelle Erprobung und Vergleich mit bisherigem Verfahren

5.1 Experimentelle Erprobung des neuen Verfahrens in der Simulationsumgebung

Das vorgestellte Verfahren wurde in der Simulation getestet und hat sich, vor allem bzgl. des Fahrtweges, als sehr effizient erwiesen. Die vollständige Erkundung der Testräume bis zum autonomen Abbruch ist in Abb. 5.1 dargestellt. Es ist zu erkennen, dass der Roboter den unbekanntem Raum auf direktem Weg exploriert und dabei nur einen sehr geringen Teil unerkundet lässt.

Zur Veranschaulichung des Algorithmus sei die Situation an dem dritten Viewpoint, nach zwei Explorationsschritten, in Abb. 5.2 und 5.3 detailliert dargestellt. Der zurückgelegte Pfad ist dabei durch weiße Linien dargestellt, die Verbindung zum nächsten ausgewählten Viewpoint durch eine graue Linie. In Abb. 5.2(b) sind 3 Grenzregionen (rot) ersichtlich, welche die Mindestgröße überschreiten und demnach dafür Anfahrpunkte berechnet werden. Die Berechnung eines Anfahrpunktes ist für die Grenzregion, welche schließlich gewählt wird, in Abb. 5.3 dargestellt. Die Grenzregion wird in Abb. 5.3(a) um den Sicherheitsabstand entsprechend verschoben und ihr Mittelpunkt berechnet (blauer Pfeil). Der dadurch bestimmte, noch zu überprüfende, Anfahrpunkt wird im Anschluss in blau dargestellt. Beim folgenden Kollisionstest in Abb. 5.3(b) existieren Voxel (rot), innerhalb des Sicherheitsbereichs (blauer Kreis), um den berechneten Anfahrpunkt (blau), welche sich nicht unter den erreichbaren Voxel befinden. Es werden folglich die erreichbaren Voxel (grau und grün) um den Sicherheitsabstand reduziert und es resultieren die reduzierten, sicher erreichbaren Voxel (grün). Anschließend wird der in Kollision stehende Anfahrpunkt (blau) auf das Voxel, innerhalb der reduzierten erreichbaren Voxel, mit geringstem euklidischen Abstand verschoben (roter Pfeil). Durch die Auswertung von erwartetem Informationsgewinn und Fahrtkosten wird der in der Simulation in Abb. 5.2(a) dargestellte Anfahrpunkt (mit dem Roboter durch dunkelgraue Linie verbunden) gewählt. Es wird dabei der relative erwartete Informationsgewinn mit dem Faktor 0,8 und die relativen Fahrtkosten mit dem Faktor 0,2 gewichtet. Andere Gewichtungen sowie das Hinzufügen von weiteren Kriterien unter Verwendung des AHP können für andere zu explorierende Umgebungen oder Anforderungen sinnvoll sein.

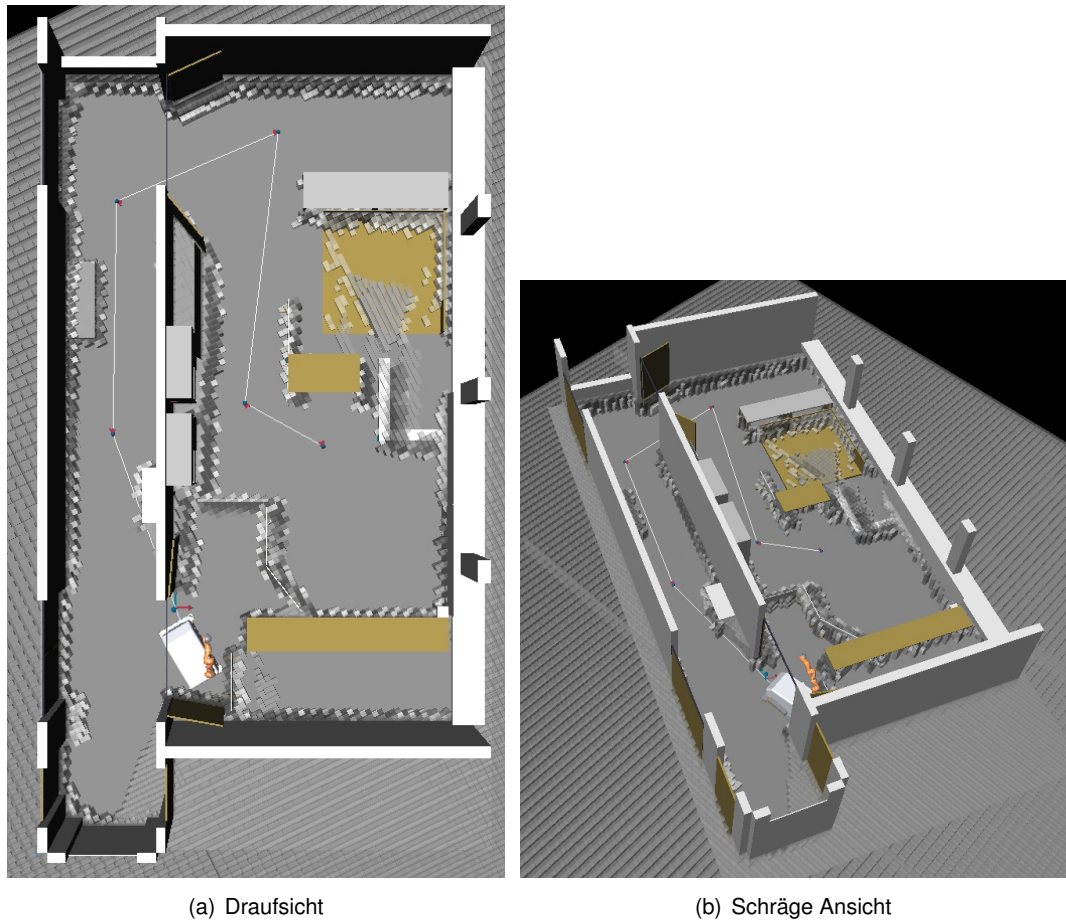
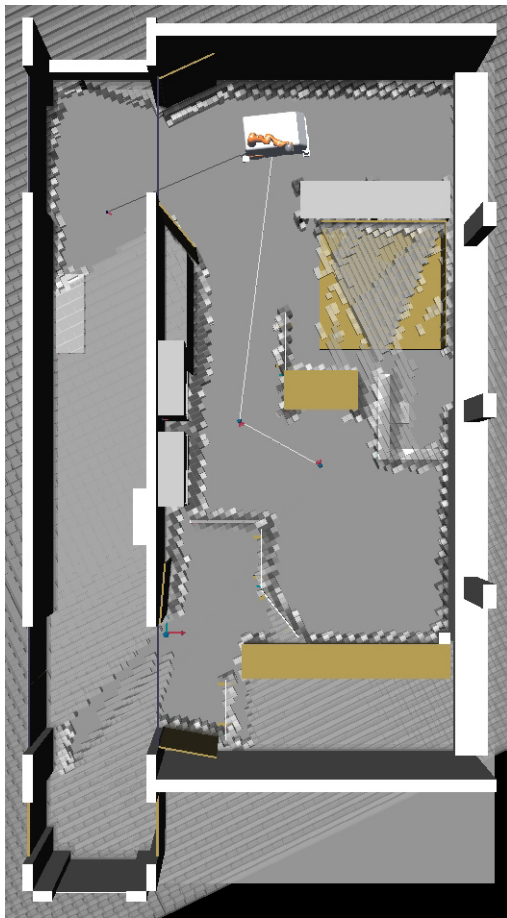


Abbildung 5.1: Simulationsumgebung nach vollständiger Exploration

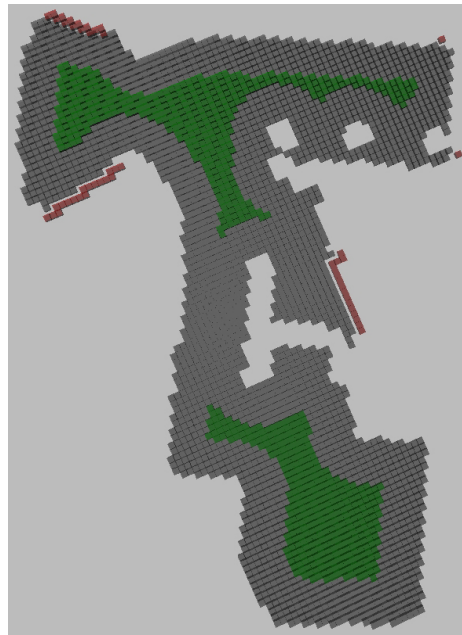
5.2 Performancevergleich des neuen Verfahrens mit dem Bisherigen

Wie aus Abb. 5.1 ersichtlich ist, findet der Roboter bei dem neuen Verfahren die Durchgänge zu noch unbekanntem Abschnitten sofort. Somit ist eine schnelle Exploration möglich. Bei dem bisherigen Verfahren handelt es sich um ein Verfahren, welches dazu benutzt wird, den Roboter in einer bekannten Karte zu lokalisieren. Dazu dreht sich der Roboter zunächst vollständig um die eigene Hochachse und führt Messungen mit den ToF-Kameras durch. Kann sich der Roboter aus den dabei erstellten Tiefenbildern nicht lokalisieren, fährt er eine definierte Distanz geradeaus. Ist dies nicht möglich, da für die bevorstehende Bewegung eine Kollision detektiert wird, dreht er sich um 90 Grad nach links und versucht die Bewegung erneut. Falls auch in diese Richtung keine Bewegung möglich ist, dreht

5.2 Performancevergleich des neuen Verfahrens mit dem Bisherigen



(a) Simulationsumgebung



(b) Erreichbare Voxel (grau und grün), sicher erreichbare Voxel (grün), sowie Grenzvoxel (rot)

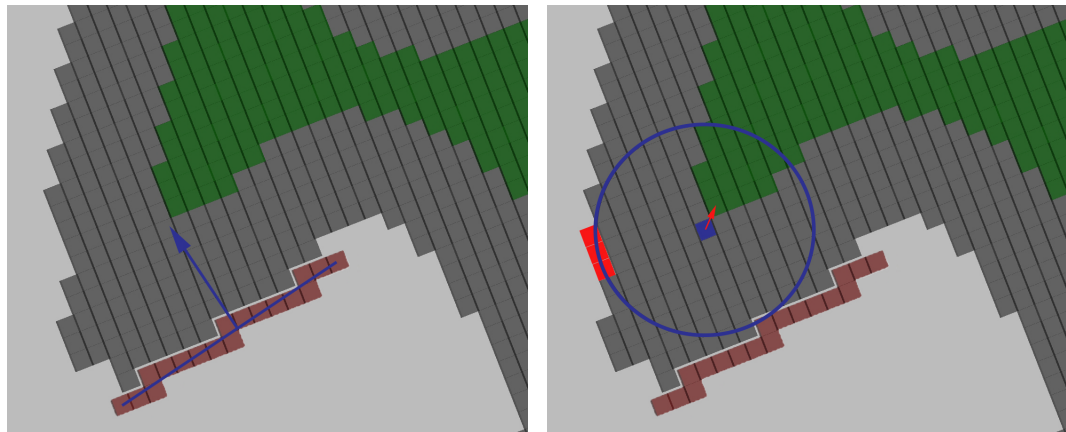
Abbildung 5.2: Situation nach zweitem Explorationsschritt

er sich um 180 Grad nach rechts und falls wieder keine Bewegung möglich ist, dreht er sich 90 Grad nach links in die Ausgangsrotation und fährt rückwärts. Nach einer erfolgten Bewegung akquiriert der Roboter jeweils, während einer vollständigen Drehung um seine Hochachse, erneut Tiefenbilder. Dieses Vorgehen wird solange durchgeführt, bis sich der Roboter in der Karte erfolgreich lokalisieren kann. Ist keine Bewegung mehr möglich bricht der Roboter den Versuch ab.

Um das Verfahren mit dem neu entwickelten Explorationsverfahren vergleichen zu können, wird eine erfolgreiche Lokalisierung in der Programmierung ausgeschlossen, wodurch der Roboter die Karte solange exploriert, bis er sich nicht mehr bewegen kann. In Abb. 5.4 ist der zeitliche Verlauf der beiden Verfahren dargestellt.

Aus dem gezeigten Ablauf wird deutlich, dass das bisherige Verfahren, angewendet zur Ex-

5 Experimentelle Erprobung und Vergleich mit bisherigem Verfahren



(a) Verschiebung der Grenzregion (rot) und Mittelpunktbestimmung zur Berechnung eines Anfahrpunktes (blauer Pfeil)

(b) Evaluierung des Anfahrpunktes durch Test, ob alle Voxel innerhalb eines Sicherheitsradius (blauer Kreis) erreichbare Voxel sind. Da der Test negativ ausfällt (rote Voxel sind nicht erreichbare Voxel) folgt die Neubestimmung des Anfahrpunktes, indem die sicher erreichbaren Voxel (grün) bestimmt werden und darin das Voxel mit dem geringsten euklidischen Abstand zum bisherigen Anfahrpunkt als neuer Anfahrpunkt gewählt wird.

Abbildung 5.3: Bestimmung des Anfahrpunktes einer Grenzregion

ploration, für diesen Raum gänzlich ungeeignet ist. Aufgrund des einfachen Algorithmus, bewegt sich der Roboter stets zwischen zwei Hindernissen, ohne dabei einen Ausweg zu finden und neues Gebiet zu explorieren. Auch nach einer Zeitdauer von vier Stunden hatte sich nichts an der Situation geändert.

Bisher war der Startpunkt stets fix, um direkte Vergleichbarkeit zu gewährleisten. Im Folgenden sollen die Verfahren für einen zufällig generierten Startpunkt, innerhalb eines Bereichs um den zuvor verwendeten Startpunkt, getestet werden. In Abb. 5.5 sind die Explorationsverläufe bei zufällig generierten Startpunkten dargestellt. Bei dem bisherigen Verfahren gelingt es dem Roboter nicht, das Startgebiet zu verlassen um den weiteren Raum zu explorieren. Es wurden insgesamt zwanzig Versuche mit unterschiedlichen Startpunkten durchgeführt, welche jeweils mindestens zwanzig Minuten laufen gelassen wurden. In keinem Fall war es dem so programmierten Roboter gelungen, den Startbereich zu verlassen. Für das neue Verfahren ist die Wahl des ersten Anfahrpunktes und somit auch der weitere Verlauf, kaum abhängig von einem leicht variierten Startpunkt. Dies ist auch sinnvoll, da dieser Anfahrpunkt zur schnellst möglichen Erkundung führt. Es wurden weitere zehn Durchläufe, mit zufällig variierten Startpunkten im Startbereich, beider Verfahren durchgeführt und sind in Tabelle 5.1 dargestellt. Das bisherige Verfahren konnte die Erkundung weder abschließen noch hat es die Exploration autonom abgebrochen. Der

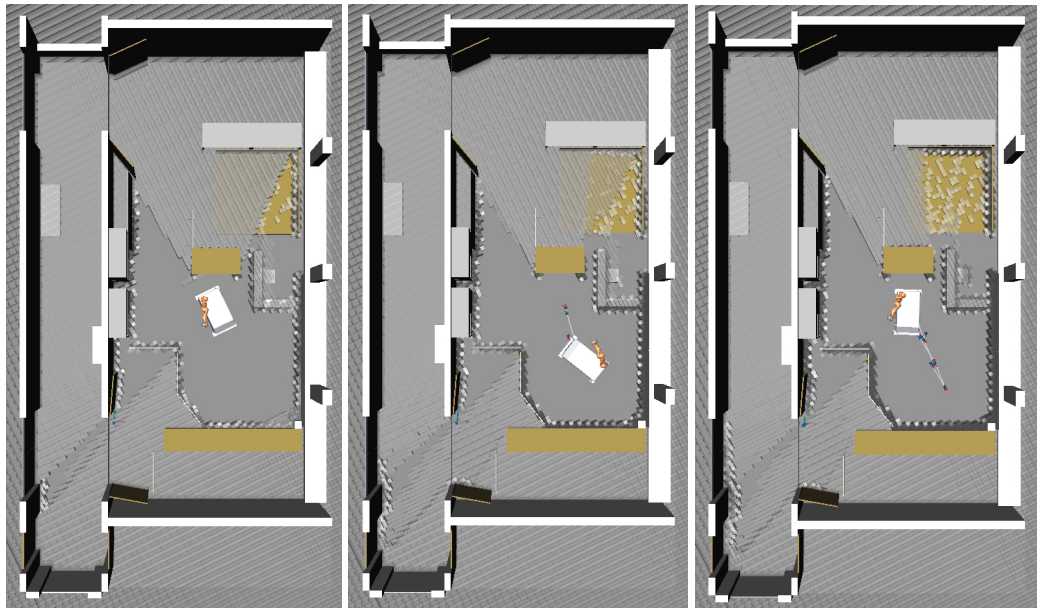
5.2 Performancevergleich des neuen Verfahrens mit dem Bisherigen

Tabelle 5.1: Auswertung der beiden Verfahren, gemittelte Werte nach 10 Durchläufen, zufällig variiertes Startpunkt

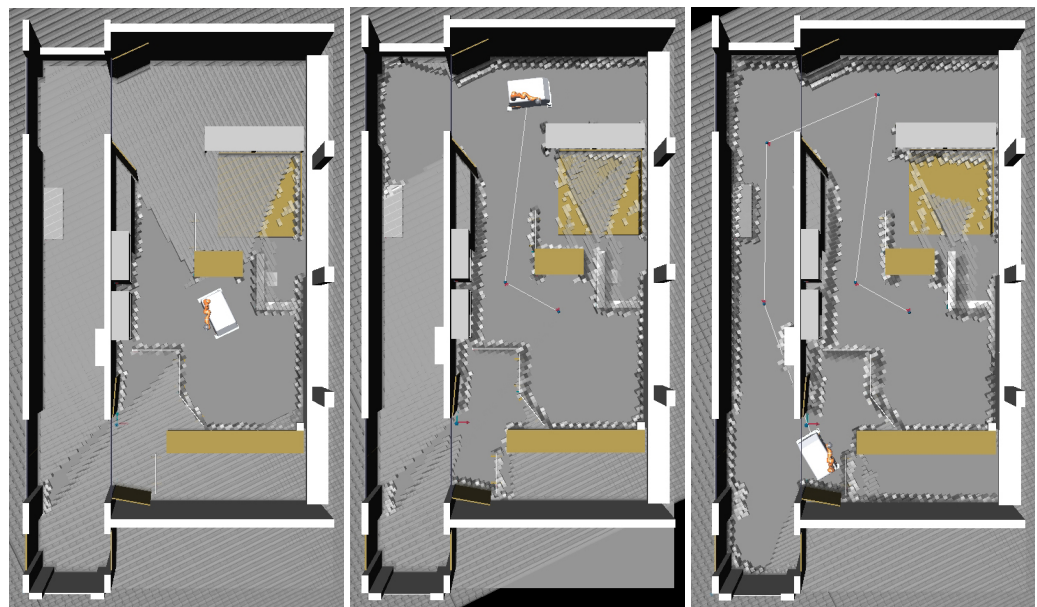
Verfahren	Erkundungszeit in min:s			Anzahl NBV-Positionen		
	min.	avg.	max.	min.	avg.	max.
Neu	9:48	10:07	10:53	5	5,2	6
Bisher	Abbruch nach 30:00			25	27,3	29

Startbereich wurde dabei nie verlassen. Es werden unbegrenzt NBV-Positionen berechnet, bis das Programm vom Benutzer manuell abgebrochen wird. Der manuelle Abbruch erfolgt stets nach dreißig Minuten. Die Erkundungsfahrt des Roboters unter Verwendung des neuen Verfahrens verläuft bei unterschiedlichen Startpunkten sehr ähnlich zu der Fahrt bei fixem Startpunkt, wie in Abb. 5.5 zu sehen ist. In zwei, der zehn Durchläufe war eine zusätzliche NBV-Position nötig, um die Exploration abzuschließen, ansonsten waren fünf NBV-Positionen ausreichend.

5 Experimentelle Erprobung und Vergleich mit bisherigem Verfahren



(a) bisheriges Verfahren 1:30 min (b) bisheriges Verfahren 3:30 min (c) bisheriges Verfahren 9:00 min



(d) neues Verfahren 1:30 min (e) neues Verfahren 3:30 min (f) neues Verfahren 9:00 min

Abbildung 5.4: Zeitlicher Verlauf der beiden Verfahren

5.2 Performancevergleich des neuen Verfahrens mit dem Bisherigen

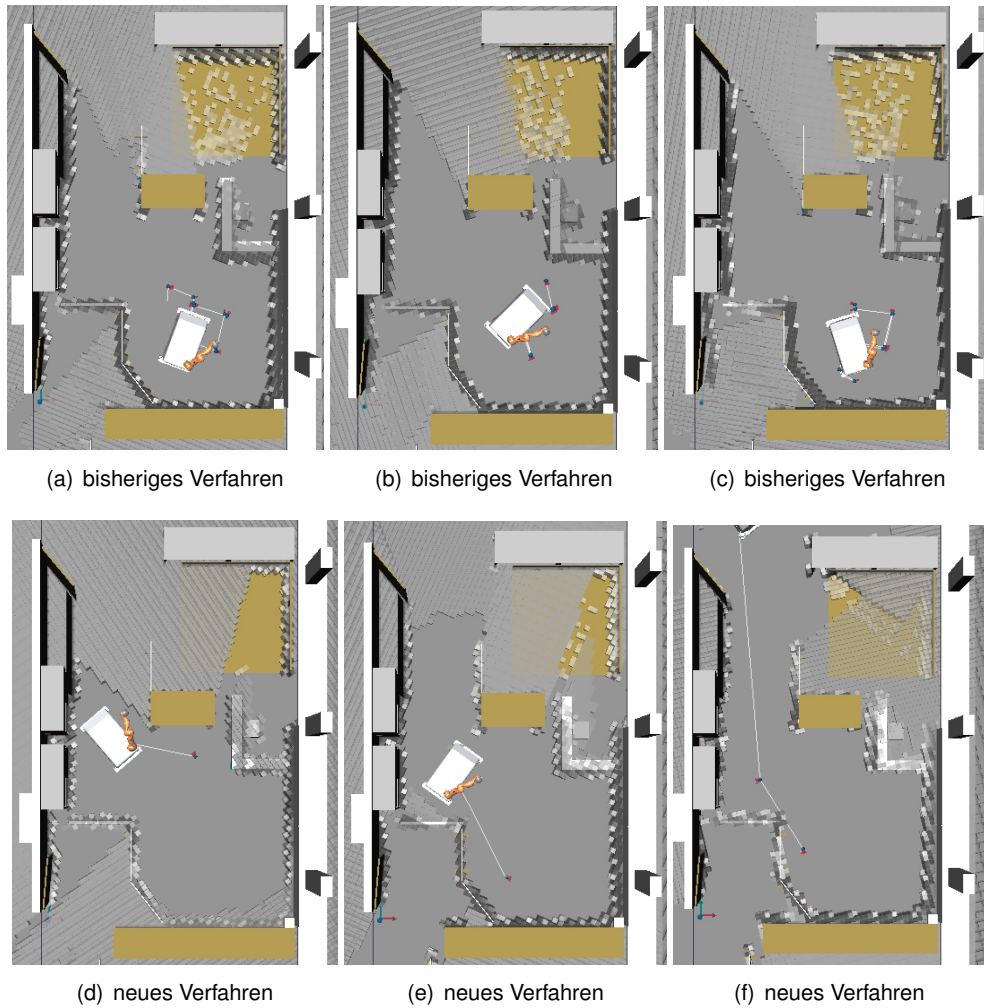


Abbildung 5.5: Verlauf der beiden Verfahren mit variablen Startpunkten

6 Zusammenfassung und Ausblick

Im Rahmen dieser Bachelorarbeit wurde eine Explorationskomponente für das Robotik-Software-Framework des Instituts für Robotik und Mechatronik des DLR entwickelt. Diese Komponente ist nötig, um einem mobilen Roboter die Möglichkeit zu eröffnen, ein SLAM-Verfahren autonom durchzuführen. Ein bereits am Institut vorhandenes Verfahren, welches bei der Lokalisierung des Roboters Anfahrpunkte berechnet, hat sich zu diesem Verwendungszweck als ungeeignet herausgestellt.

Es wurden zunächst diverse bereits existierende Next-Best-View-Verfahren analysiert und auf Adaptierbarkeit für diese Aufgabe untersucht. Aus den Erkenntnissen der Recherche wurde ein neues Verfahren entwickelt.

Es hat sich gezeigt, dass durch eine optimierte Auswahl an Next-Best-View-Observationsposition Räume auf kurzem Weg sehr schnell zu erkunden sind. Im Gegensatz zu dem sehr einfachen Verfahren, welches bei der Lokalisierung Anwendung findet, ist es mit dem neuen Verfahren möglich, Gebiete vollständig zu erkunden, sofern es die Abmessungen des Roboters zulassen. Das bisherige Verfahren hatte es dem Roboter nicht ermöglicht, den Startbereich zu verlassen, um weiteres Gebiet zu vermessen. Auch in Umgebungen, wo das einfache Verfahren evtl. mehr Raum explorieren könnte, wird der Fahrweg darin sehr lang und ineffizient sein. Der Rechenaufwand zur Bestimmung von nächsten Anfahrpunkten ist dabei zwar sehr gering, jedoch ist davon auszugehen, dass die Explorationsdauer aufgrund des langen Fahrtweges deutlich länger sein wird. Bei dem hier entwickelten Verfahren wird durch entsprechenden Rechenaufwand der Fahrtweg und somit die gesamte Explorationsdauer verringert.

Der Energieverbrauch konnte hier nicht analysiert werden, da der Roboter lediglich simuliert wurde. Aufgrund der Kenndaten von Motoren, Messsystem und Steuerungshardware ist aber davon auszugehen, dass der Energieverbrauch durch den geringeren Fahrweg, die geringere Anzahl an Messungen, sowie die kürzere Betriebszeit deutlich niedriger ist, als bei Verfahren welche kein kostenoptimiertes Verfahren zur Erkundung verwenden.

Das neue Verfahren steht jetzt als Explorationskomponente für das sich noch in Entwicklung befindliche SLAM-Verfahren bereit. Diese Verfahren sind im Weiteren auf den realen Roboter zu übertragen und zu testen. Eine noch fehlende Komponente ist die Pfadplanung, welche zu adaptieren ist. In der Simulation wurde der Roboter bisher nur auf den nächsten Viewpoint gesetzt, ohne dass der Pfad dorthin geplant wurde.

Potential für eine weitere Optimierung liegt darin, die Voxelkarte dynamisch zu erweitern und nicht schon zu Beginn mit der maximalen Größe zu initialisieren. Zwar benötigt eine große Karte zu Beginn aufgrund der Homogenität mit der Verwendung des Dynamic

6 Zusammenfassung und Ausblick

Octree Space wenig Speicher, jedoch ist die Flexibilität eingeschränkt. Wird ein Gebiet exploriert, dessen Größe a priori nicht bekannt ist, wird dies bei einer statischen Karte, welche zu klein initialisiert wurde, zu Fehlern führen. Da die Adaptierbarkeit des Verfahrens eines der allgemeinen Anforderungskriterien ist, ist eine dynamische Karte zu bevorzugen.

Die Rechenleistung ist seit der Entwicklung der ersten Halbleiter stets sehr schnell gestiegen und günstiger geworden [19], wie auch die letzten Jahre weiterhin zu beobachten war[15]. Auch der Energieverbrauch wurde dabei immer effizienter, was vor allem dem Einsatz von Mehrkernprozessoren ab dem Jahr 2004 zu verdanken ist [27]. Die Entwicklung läuft mit einem Tempo ab, welches deutlich über dem der Entwicklung von effizienten Antrieben und Energiespeichern liegt. In Zukunft wird also der Rechenaufwand, im Verhältnis zu Stromverbrauch von Antrieben und Verschleiß von mechanisch beanspruchten Komponenten, immer unbedeutender. Somit werden die Vorteile von Algorithmen, wie diesem, welche durch zusätzlichen Rechenaufwand den Fahrweg minimieren können, noch stärker zum Tragen kommen.

Aus diesem Grund wird gerade auch in Zukunft eine akkurate Next-Best-View-Planung wichtiger Bestandteil von Explorationsverfahren sein und es sich stets lohnen, nach weiteren Optimierungen zu forschen.

Literaturverzeichnis

1. F. Amigoni und A. Gallo. A multi-objective exploration strategy for mobile robots. In *International Conference on Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE*, S. 3850–3855. IEEE, 2005.
2. N. Basilico, F. Amigoni, L. Tanca und B. Pernici. *Navigation Strategies for Exploration and Patrolling with Autonomous Mobile Robots*. Dissertation, Politecnico di Milano, 2010.
3. T. Bodenmüller. *Streaming surface reconstruction from real time 3D measurements*. Dissertation, Technische Universität München, 2009.
4. E. Bourque und G. Dudek. Viewpoint selection - an autonomous robotic system for virtual environment creation. In *International Conference on Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ*, Band 1, S. 526–532. 1998. doi:10.1109/IROS.1998.724672.
5. L. Freda und G. Oriolo. Frontier-based probabilistic strategies for sensor-based exploration. In *International Conference on Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE*, S. 3881–3887. IEEE, 2005.
6. H.H. Gonzalez-Banos und J.C. Latombe. Navigation strategies for exploring indoor environments. In *The International Journal of Robotics Research*, 21(10-11), S. 829–848, 2002.
7. D.B. Henry L. Jones, Stephen M. Rock und S. Morris. Autonomous Robots in SWAT Applications: Research, Design, and Operations Challenges. In *Proceedings of the 2002 Symposium for the Association of Unmanned Vehicle Systems International (AUVSI '02)*. AUVSI, AUVSI, Orlando, FL, July 2002.
8. A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss und W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. In *Autonomous Robots*, 2013. doi:10.1007/s10514-012-9321-0. URL <http://octomap.github.com>.
9. International Hellenic University Paraskevopoulos Konstantinos. The analytic hierarchy process (AHP) lesson 1. URL http://rad.ihu.edu.gr/fileadmin/labsfiles/decision_support_systems/lessons/ahp/AHP_Lesson_1.pdf. Aufgerufen am 25.06.2013.

Literaturverzeichnis

10. T. Kohonen. The self-organizing map. In *Proceedings of the IEEE*, Band 78, S. 1464 – 1480. IEEE, September 1990.
11. T. Kohonen, M.R. Schroeder und T.S. Huang (Hrsg.). *Self-Organizing Maps*. 3. Auflage. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001. ISBN 3540679219.
12. S. Kriegel, M. Brucker, Z.C. Marton, T. Bodenmüller und M. Suppa. Combining object modeling and recognition for active scene exploration. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*. Accepted for publication, 2013.
13. S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa und G. Hirzinger. Next-best-scan planning for autonomous 3D modeling. In *IROS*, S. 2850–2856. IEEE, 2012. ISBN 978-1-4673-1737-5.
14. J.J. Leonard und H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D. Koditschek und J. Hollerbach (Hrsg.), *Robotics Research: The Ninth International Symposium*, S. 169–176. Springer Verlag, Snowbird, Utah, 2000.
15. M.S. Li. Keeping up with moore's law. Mai 2013. URL <http://dujs.dartmouth.edu/spring-2013-15th-anniversary-edition/keeping-up-with-moores-law>. Aufgerufen am 05.07.2013.
16. K.L. Low. *View planning for range acquisition of indoor environments*. Dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 2006. AAI3207396.
17. G. Oriolo, M. Vendittelli, L. Freda und G. Troso. The SRT method: Randomized strategies for exploration. In *International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE*, Band 5, S. 4688–4694. IEEE, 2004.
18. T.L. Saaty. *Multicriteria decision making - the analytic hierarchy process. Planning, priority setting, resource allocation*. RWS Publishing, 1990. ISBN 0-9620317-2-0.
19. R.R. Schaller. Moore's law: past, present, and future. In *IEEE Spectr.*, 34(6), S. 52–59, Juni 1997. ISSN 0018-9235. doi:10.1109/6.591665.
20. W.R. Scott, G. Roth und J.F. Rivest. View planning for automated three-dimensional object reconstruction and inspection. In *ACM Comput. Surv.*, 35(1), S. 64–96, März 2003. ISSN 0360-0300. doi:10.1145/641865.641868. URL <http://doi.acm.org/10.1145/641865.641868>.
21. W.B. Sebastian Thrun und D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

22. D. Seth. *Statische 3D-Szenen am Beispiel des DLR-Roboters „Rollin’ Justin“ (Arbeitspaket: Volumenbasierte Umweltmodellierung)*. Entwicklungs- und Lizenzvertrag über eine strategische Kooperation zum Know-how-Transfer im Bereich Service- und Industrierobotik, Institut für Robotik und Mechatronik, Deutsches Zentrum für Luft- und Raumfahrt, 2010.
23. D. Seth. *Effizientes Lesen und Schreiben statischer 3D-Karten und Raytracing-Verbesserungen (Arbeitspaket: Volumenbasierte Umweltmodellierung)*. Entwicklungs- und Lizenzvertrag über eine strategische Kooperation zum Know-how-Transfer im Bereich Service- und Industrierobotik, Institut für Robotik und Mechatronik, Deutsches Zentrum für Luft- und Raumfahrt, 2011.
24. C.E. Shannon und W. Weaver. A mathematical theory of communication. 1948.
25. M. Strauch. Topologische Umweltmodellierung und Pfadplanung bei mobilen Robotern. Fakultät für Informatik Prof. Dr. Andreas Zell Uni Tübingen, 5 2002.
26. M. Suppa. *Autonomous robot work cell exploration using multisensory eye-in-hand systems*. Dissertation, Universität Hannover, 2008.
27. N. Thompson. Moore’s law goes multicore: The economic and strategic consequences of a fundamental change in how computers work. August 2012. URL http://faculty.haas.berkeley.edu/neil_thompson/Innovation_Seminar/Papers/Neil_Thompson%27s_Moore%27s_Law_Paper_-_Summary.pdf. Aufgerufen am 05.07.2013.
28. Univ.-Prof Dr. Stephan Zelewski, Malte L. Peters. AHP - dargestellt am Beispiel der Auswahl von Projektmanagement-Software zum Multiprojektmanagement. 2002. URL http://www.pim.wiwi.uni-due.de/uploads/tx_itochairt3/publications/bericht14.pdf. Aufgerufen am 25.06.2013.