



**Technische Universität Ilmenau**

Fakultät für Informatik und Automatisierung

Fachgebiet Neuroinformatik und Kognitive Robotik

# **Simultaneous Localization and Mapping (SLAM) mit hierarchischen Repräsentationsformen für 2D- und 3D-Umgebungsmodelle**

Masterarbeit zur Erlangung des akademischen Grades Master of Science

**Andreas Hollandt**

Betreuer: Dr.-Ing. Christof Schröter

Dipl.-Inf. Erik Einhorn

Verantwortlicher Hochschullehrer:

Prof. Dr. H.-M. Groß, FG Neuroinformatik und Kognitive Robotik

Die Masterarbeit wurde am 11.01.2012 bei der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau eingereicht.



## **Danksagung**

Ich möchte mich hiermit bei all jenen bedanken, die durch ihre Unterstützung auf die eine und auch andere Weise zum Gelingen dieser Arbeit beigetragen haben.

Mein besonderer Dank gilt meinen beiden Betreuern Dr.-Ing. Christof Schröter und Dipl.-Inf. Erik Einhorn, die mir in unseren zahlreichen Sitzungen und Diskussionen trotz konstanter Belastung durch ihre eigenen Projektverpflichtungen und daraus resultierender Zeitknappheit stets mit Rat und Tat zur Seite standen. Ohne ihre Unterstützung in konzeptionellen Fragen oder auch bei der nervenaufreibenden Jagd nach gefühltermaßen nie enden wollenden Fehlerquellen wäre diese Masterarbeit nicht möglich gewesen.

Aber auch meiner Freundin und meinen Eltern gebührt spezieller und herzlicher Dank für ihre Unterstützung, Ermutigung und vor allem für das entgegengebrachte Verständnis in dieser schwierigen Phase.

Erklärung: „Hiermit versichere ich, dass ich diese Masterarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet.“

Ilmenau, 11.01.2012

.....

Andreas Hollandt

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielstellung . . . . .	2
1.3	Gliederung der Arbeit . . . . .	3
<b>2</b>	<b>State of the Art</b>	<b>5</b>
<b>3</b>	<b>Theoretische Grundlagen</b>	<b>9</b>
3.1	Kartenrepräsentation . . . . .	9
3.1.1	Topologische Karten . . . . .	9
3.1.2	Metrische Karten . . . . .	10
3.2	Zustandsschätzung . . . . .	14
3.2.1	Kalmanfilter . . . . .	15
3.2.2	Partikelfilter . . . . .	17
3.3	Distanztransformation . . . . .	19
<b>4</b>	<b>Der SLAM-Algorithmus</b>	<b>21</b>
4.1	Umgebungsrepräsentation . . . . .	21
4.2	Lokalisierung . . . . .	23
4.2.1	Bewegungsmodell . . . . .	23

---

4.3	Resampling . . . . .	25
4.3.1	Kartenübereinstimmung . . . . .	25
4.3.2	Partikelgewichtung . . . . .	32
4.4	Experimentelle Untersuchungen . . . . .	34
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>39</b>
5.1	Zusammenfassung . . . . .	39
5.2	Weiterführende Arbeiten . . . . .	40
5.2.1	Anwendung auf den dreidimensionalen Fall . . . . .	40
5.2.2	Verbesserung der Distanztransformation . . . . .	40
5.2.3	Partikelschwund . . . . .	41
5.2.4	Reduzierung des Speicherbedarfs . . . . .	42
	<b>Literaturverzeichnis</b>	<b>46</b>

# Kapitel 1

## Einleitung

### 1.1 Motivation

Zu den grundlegendsten und am meisten untersuchten Problemen der mobilen Robotik zählt die selbstständige Navigation zu vorgegebenen oder im Rahmen der Umwelterkundung oder anderer Aufgaben selbstgewählten Zielen im Arbeitsbereich des Roboters. Wesentliche Voraussetzung hierfür ist der Aufbau einer möglichst akkuraten Repräsentation der Umgebung und die kontinuierliche Bestimmung des eigenen Standortes.

Diese beiden Teilaspekte sind eng miteinander verbunden. Um die eigene Position im Raum zu bestimmen, muss eine möglichst genaue Karte vorliegen, um sie mit den sensorischen Beobachtungen zu vergleichen. Zu deren korrektem Aufbau muss jedoch stets die aktuelle Position bekannt sein, um Sensordaten korrekt auswerten und in die Karte an der richtigen Stelle einfügen zu können. Diese Problematik wird in der Literatur gemeinhin als *Simultaneous Localization and Mapping* (SLAM) bezeichnet, um zu verdeutlichen, dass beide Aufgaben gleichzeitig gelöst werden müssen.

In der Fachliteratur wurden im Laufe der Zeit verschiedenste Lösungsansätze vorge-

---

stellt und eine Vielfalt an Umgebungsrepräsentationsformen und Verfahren zur Zustandsschätzung entwickelt. Lag der Forschungsschwerpunkt anfangs vor allem auf zweidimensionalen Ansätzen, ist gerade in den letzten Jahren ein steigendes Interesse an Verfahren zu erkennen, welche die zusätzlichen Informationen von dreidimensionalen Daten liefernden Sensoren ausnutzen können. Entscheidende Gründe für diese Entwicklung sind die stetig zunehmende Leistungsfähigkeit der Mikroprozessoren, gestiegene Anforderungen an die Fähigkeiten der Roboter und nicht zuletzt die veränderte Preislage der Messinstrumente. Seit der Einführung des eigentlich nur als Spielsteuerung gedachten Kinect-Systems wird auch dieses als kompakter 3D-Sensor mit insbesondere im Vergleich zu den häufig genutzten 3D-Laser-Rangescannern moderaten Kosten eingesetzt.

Viele in der Literatur vorgestellte Verfahren sind allerdings sehr auf die verwendeten Messgeräte und das Einsatzgebiet des jeweiligen Roboters angepasst. Es ist jedoch wünschenswert, dass SLAM-Verfahren generisch mit verschiedenen Sensoren sowohl im Zwei- als auch im Dreidimensionalen arbeiten können.

## 1.2 Zielstellung

Gegenstand dieser Arbeit ist die Konzeption und Implementierung eines SLAM-Moduls für ein am Fachgebiet Neuroinformatik und Kognitive Robotik als Alternative zum etablierten *Robot Operating System* [ROS, 2011] von Grund auf neu entwickeltes Roboterbetriebssystem, welches in der Lage sein soll je nach vorhandener Sensorik eine global konsistente zwei- oder dreidimensionale Karte seiner Umgebung zu erstellen.

Die Aufgabe baut wesentlich auf der im Rahmen des SerRoKon-Projektes geleisteten Vorarbeit auf, welche ausführlich in [SCHRÖTER, 2008] vorgestellt wurde. Insbesondere

---



---

re soll der darin entwickelte MapMatch-SLAM Algorithmus für das neue System implementiert und erweitert werden. Die Umgebungsrepräsentation wird dabei auf den ebenfalls im Fachgebiet entwickelten und kürzlich veröffentlichten  $N^d$ -Bäumen [EINHORN et al., 2011], einer Verallgemeinerung von Quad- und Octrees zur effizienten Handhabung von Gridkarten, aufbauen. Zur Berechnung der Schätzungsgüte wird das ursprüngliche, einfache Verfahren durch ein kostenintensiveres, aber auch leistungsfähigeres Maß basierend auf dem Prinzip der Distanztransformation ersetzt.

### 1.3 Gliederung der Arbeit

Diese Arbeit ist wie folgt strukturiert. Das zweite Kapitel enthält eine kurze Übersicht über existierende Ansätze zur Lösung des SLAM-Problems im zwei- und dreidimensionalen Raum. Darauf folgend wird auf wichtige theoretische Grundlagen für dieses Gebiet eingegangen. Insbesondere werden einige existierende Verfahren genauer beleuchtet und Vor- und Nachteile diskutiert. Abschließend werden die für diese Arbeit wesentlichen, im Fachgebiet erfolgten Vorarbeiten vorgestellt. Im vierten Kapitel werden sodann die Details des im Rahmen dieser Arbeit implementierten SLAM-Systems erläutert und Ergebnisse experimenteller Auswertungen des Systems mit Hilfe von Sensordaten, die mit einer im Fachgebiet genutzten Robotertestplattform aufgenommen wurden, präsentiert. Abschließend werden die Ergebnisse zusammengefasst und ein Ausblick auf mögliche zukünftige Weiterentwicklungen des vorgestellten Systems gegeben, die im Rahmen dieser Arbeit nicht mehr vollendet werden konnten.

---



# Kapitel 2

## State of the Art

Als eine der Schlüsselherausforderungen der mobilen Robotik erfreut sich das SLAM-Problem in der Fachliteratur außerordentlicher Aufmerksamkeit. Auf Grund der Vielfalt der mittlerweile existierenden Verfahren ist eine eindeutige Kategorisierung schwierig. In dieser Arbeit werden daher die funktionalen Bestandteile des Problems individuell betrachtet.

### **Positionsschätzung**

Eine sehr einfache Möglichkeit der Positionsschätzung bieten Odometriekorrekturnverfahren. In [SCHRÖTER, 2008] wurde ein solches vorgestellt und konkludiert, dass der entstehende Schätzungsfehler trotz relativ guter Ergebnisse nach oben unbeschränkt bleibt.

Eine kontinuierliche Schätzung der aktuellen Position kann theoretisch auch durch Zuhilfenahme einer externen Referenz erfolgen, welche zum Beispiel mittels des *Global Positioning System* (GPS) oder durch spezielle Präparation des Einsatzortes mit künstlichen Landmarken erhalten werden kann.

Diese Vorgehensweise erfordert jedoch einerseits die möglicherweise umständliche oder

---

gar unmögliche Installation solcher Systeme und andererseits wird die Einsatzfähigkeit des Roboters abhängig von deren Funktionstüchtigkeit. So stellt die Nutzung von GPS-Daten in Innenräumen oder Außenarealen mit ungünstigen Reflexionen und Abschirmungen noch immer ein Problem dar. Zudem hängt auch die Genauigkeit der Positionsdaten stark von der Empfangsqualität ab und kann nur mit zusätzlichen Korrekturmaßnahmen (zum Beispiel *Differential GPS*) überhaupt mit einem maximalen Fehler von unter einem Meter dienen.

Daher wird ein Roboter in den meisten Anwendungsfällen zur Wahrung seiner Eigenständigkeit mit Sensoren zur mehr oder weniger komplexen Wahrnehmung der Umgebung ausgestattet. Mit diesen Messwerten kann die Eigenposition verlässlich und dauerhaft geschätzt werden. Lange Zeit dominierten dabei ausgehend von [SMITH et al., 1990] Ansätze, welche sich dem bewährten Kalmanfilter [KALMAN, 1960] bedienen. Mit der Vorstellung des FastSLAM-Verfahrens [MONTEMERLO et al., 2002] und der damit verbundenen Demonstration der hervorragenden Eignung von Partikelfiltern zur Umsetzung des SLAM-Prinzips setzte jedoch eine zunehmende Nutzung dieser Technik ein; insbesondere, da beliebige Wahrscheinlichkeitsverteilungen approximiert werden können.

Dank immer weiter steigenden Anforderungen an die Leistungsfähigkeit und den Funktionsumfang mobiler Roboter und neuen erschwinglichen Sensoren wie dem Kinect-System ist jedoch gerade in den letzten Jahren ist das Interesse an dreidimensionale Daten liefernden Sensoren enorm gestiegen.

Diese Entwicklung verlangte nach der Nutzung dreidimensionaler Umgebungsrepräsentationen um die Sensordaten vollständig repräsentieren und ausnutzen zu können.

---

## Kartierung

Erste Arbeiten nutzten hierfür die naheliegende direkte Erweiterung von Belegtheitskarten auf drei Dimensionen [MORAVEC, 1996], welche den Raum in gleich große Würfel aufteilen, in dieser Arbeit daher im Folgenden auch als Voxelkarten bezeichnet.

Der Vorteil dieser Repräsentationsform liegt vor allem in der konzeptionellen Einfachheit. Nachteilig ist hingegen der enorme Speicherbedarf, da die Karte mindestens so groß sein muss wie die Einhüllende der bereits besuchten Regionen. Insbesondere bei großen Außenarealen oder bei Bedarf an fein aufgelösten Umgebungsinformationen wird dies schnell unhandbar.

Da jene Voxel mit tatsächlichen Umgebungsinformationen in der Regel nicht gleichförmig im Raum verteilt, sondern meist lokal gebündelt sind, bietet sich eine flexible Repräsentationsform mit adaptiver Auflösung an. Besonders beliebt sind hierbei baumbasierte Strukturen wie Quadrees im zweidimensionalen und die entsprechenden Octrees im dreidimensionalen Fall [WURM et al., 2010].

Kartenelemente werden bei diesem Ansatz erst bei tatsächlichen Messungen des Roboters angelegt und müssen nicht von vornherein wie bei regulären Gridkarten initialisiert werden. Bei Bedarf kann der Baum sogar als Multiresolution Repräsentation genutzt werden, indem ab einer bestimmten Tiefe abgeschnitten wird.

Eine vollkommen generische Datenstruktur wie die in dieser Arbeit verwendeten  $N^d$ -Bäume konnte jedoch so in der Literatur nicht gefunden werden.

---



# Kapitel 3

## Theoretische Grundlagen

### 3.1 Kartenrepräsentation

Für viele Probleme der mobilen Robotik ist ein möglichst genaues und reichhaltiges Modell der Umgebung essentiell. Dies schließt Aufgaben wie Kollisionsvermeidung, Pfadplanung und Lokalisierung ein. In der Fachliteratur wurden hierfür bereits verschiedenste Ansätze entwickelt, die oftmals auf die spezifischen Anforderungen ihres jeweiligen Anwendungsgebietes zugeschnitten wurden. Grob lassen sich diese in topologische und metrische Ansätze kategorisieren.

#### 3.1.1 Topologische Karten

Topologische Karten bauen auf einer Graphenstruktur auf, wobei die Knoten bereits besuchte Orte oder auffällige Landmarken repräsentieren. Kanten zwischen diesen Knoten bedeuten, dass es einen für den Roboter befahrbaren Pfad zwischen den dazugehörigen Plätzen gibt. Das Modell kann leicht mit zusätzlichen Informationen angereichert werden. Damit ist ein äußerst kompaktes und abstraktes Modell der Umgebung erstellbar, was ideal für Pfadplanungen und Lokalisierungen in großen Arealen ist. Anderer-

---

seits speichern diese Karten nur spärliche Informationen über die Umwelt und sind daher insbesondere für Einsätze in engen Innenräumen weniger geeignet, da genaue Daten über Hindernisse und Wände fehlen. Allerdings können topologische Karten auch mit anderen Modellarten kombiniert werden, um diese Nachteile auszugleichen.

### 3.1.2 Metrische Karten

Zu den metrischen Repräsentationsformen zählen insbesondere geometrische und auf Raumdekomposition basierte Ansätze. Geometrische Karten modellieren die Umwelt mit Hilfe einer Vielzahl geometrischer Objekte wie Punkten, Linien und Flächen [PATHAK et al., 2010]. Diese müssen in der Regel über eine Merkmalsextraktion aus den rohen Sensordaten gewonnen werden. Die Wahl der Beschreibungsart erlaubt eine flexible Anpassung an die Anwendungsanforderungen. Einfache Merkmale wie Punkte sind mit geringem Aufwand berechenbar, führen allerdings zu einem relativ großen Modell und entsprechend höherem Verwaltungsaufwand. Komplexe Objekte hingegen erlauben eine effiziente, abstrakte Repräsentation und bieten zudem einen hohen Wiedererkennungswert. Die Detektion erfordert jedoch aufwändige Berechnungen.

Ein genereller Nachteil dieser merkmalsbasierten Verfahren liegt darin, dass sie meist auf bestimmte Sensortypen und unter Umständen vorhandenes Umgebungsvorwissen zugeschnitten sind. Bei Wechsel der Sensoren oder Hinzufügen neuer zum Zwecke der Sensorfusion müssen dadurch in der Regel neue Algorithmen implementiert werden. Ferner wird Freiraum nicht modelliert und somit sind stets nur spärliche Informationen über die Umgebung verfügbar.

Auf Raumdekomposition basierende Repräsentationen sind detailgetreue, feinaufgelöste Umgebungsmodelle. Die am meisten verbreitete Variante ist die sogenannte Belegtheitskarte (engl. *occupancy grid map*), welche die Umwelt in gleich große, diskrete Zellen aufteilen und pro Zelle die Wahrscheinlichkeit speichern, ob sich dort ein Hin-

---



dernis befindet [MORAVEC, 1988]. Dieses an sich hochdimensionale Problem wird in der Regel in kleinere Teilprobleme zerlegt, indem der Zustand jeder einzelnen Zelle als unabhängige Zufallsvariable geschätzt wird.

Es werden also keine Annahmen über die Struktur der Umgebung getroffen und der Ansatz ist nicht auf bestimmte Sensorik beschränkt. Ebenso können die Unsicherheitscharakteristika unterschiedlicher Sensoren berücksichtigt werden.

Der größte Nachteil dieser Kartenform ist der immens hohe Speicherverbrauch. Wird eine hohe Auflösung für die jeweilige Anwendung - beispielsweise in engen Innenräumen - benötigt, müssen die Zellen sehr klein gewählt werden, wodurch sich deren Anzahl drastisch erhöht. Dies fällt insbesondere im dreidimensionalen Fall ins Gewicht.

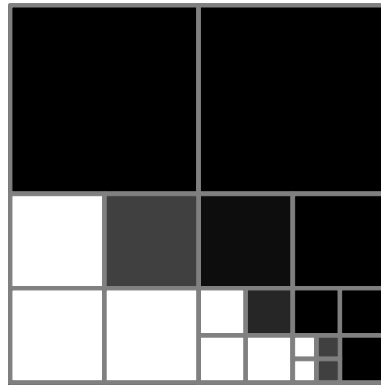
Ein weiterer Kritikpunkt ist die Fixiertheit der Auflösung. Sie muss im Vorfeld des Einsatzes festgelegt werden und kann nicht ohne großen Aufwand verändert werden. Dies kann jedoch durch hierarchische Datenstrukturen behoben werden.

### $N^d$ -Baum

In dieser Arbeit wird auf eine kürzlich vorgestellte hierarchische Repräsentationsform zurückgegriffen. [EINHORN et al., 2011] beschreibt eine „ $N^d$ -Baum“ genannte Verallgemeinerung von Quad- und Octrees, die es erlaubt einen  $d$ -dimensionalen Unterraum rekursiv mit  $N^d$  Kindern pro Knoten zu unterteilen (siehe Bild 3.1).

Der  $N^d$ -Baum ist definiert als ein zusammenhängender azyklischer gerichteter Graph  $T = (N, E)$ , dessen Knoten  $N$  und Kanten  $E$  folgende Eigenschaften aufweisen:

1. Jeder Knoten repräsentiert eine bestimmte Partition des Unterraums  $M \subset \mathbb{R}^d$ .  
Im zweidimensionalen Fall ist dies ein quadratischer Ausschnitt der Umgebung, im Höherdimensionalen ein achsenausgerichteter würfelförmiger Teilraum mit gleicher Ausdehnung entlang jeder Dimension.
-

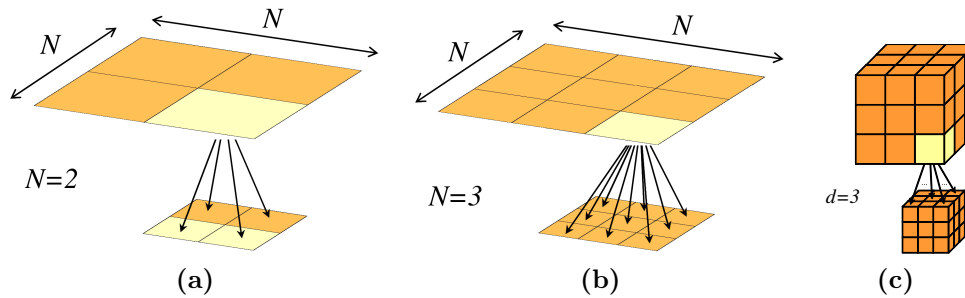


**Abbildung 3.1:** rekursive Unterteilung der Umgebung durch eine Baumstruktur  
 Das Bild zeigt einen typischen Aufbau eines Quadtrees. Dieser Datentyp ist ein Spezialfall des  $N^d$ -Baums mit  $N = 2$  und  $d = 2$ .

2. Der Raum eines jeden inneren Knotens  $n$  wird entlang jeder Dimension in  $N$  Abschnitte unterteilt, wodurch  $N^d$  Teilräume entstehen.
3. Jeder dieser Teilräume wird einem Kindknoten  $c_j$  zugewiesen und eine entsprechende Kante  $(n, c_j)$  in den Graphen eingefügt.
4. Die Teilräume sind abgeschlossen und disjunkt.
5. Es existiert genau ein ausgezeichneteter Knoten  $n_0$  ohne Vaterknoten, der als Wurzel bezeichnet wird.
6. Knoten, welche nicht weiter in Kinder unterteilt sind, werden als Blätter bezeichnet. Diese tragen die eigentlichen Daten in Form der Wahrscheinlichkeit, dass der dazugehörige Raum durch ein Hindernis belegt ist.

Die Parametrisierung ermöglicht eine flexible Anpassung der Eigenschaften der Datenstruktur sowie der Laufzeit der darauf operierenden Algorithmen an die spezifischen Erfordernisse des Anwendungsgebietes. Kleine Werte für  $N$  resultieren in tiefen und kompakten Bäumen, welche bestens für Schnittberechnungen geeignet sind. Größere

Werte begünstigen hingegen Baumdurchläufe, sind allerdings weniger speichereffizient (Bild 3.2).

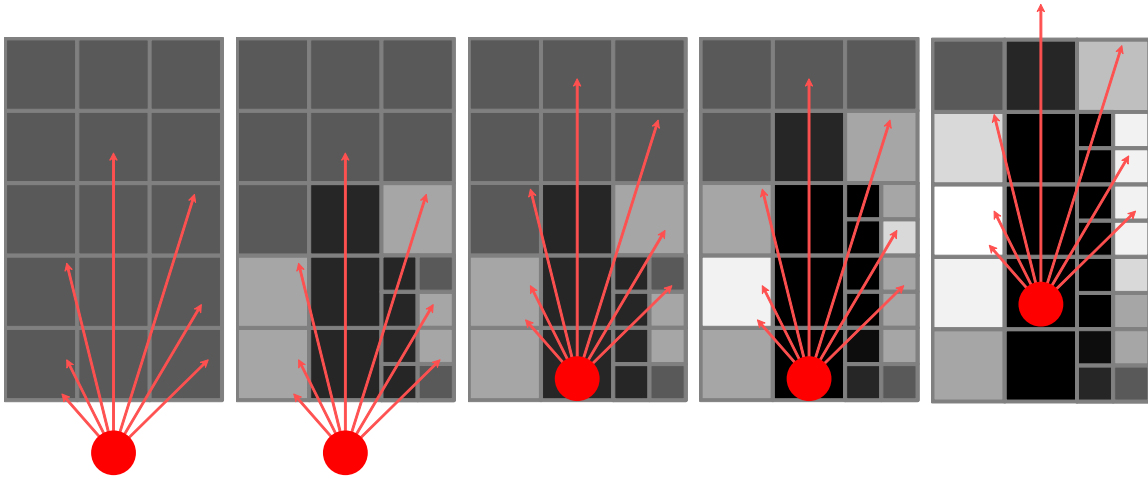


**Abbildung 3.2:** Vergleich verschiedener Werte für die Parameter  $N$  und  $d$   
 (a) Der herkömmliche Quadtree entspricht dem Spezialfall  $N = 2$  und  $d = 2$ . (b) Die Fächerung des Baums kann durch den Parameter  $N$  beeinflusst werden. (c) dreidimensionale Variante von (b)

### Adaptive Resolution Mapping

Mit dieser Datenstruktur ist die Auflösung problemlos dynamisch wählbar. Die wichtigen Hindernisse können zentimetergenau gespeichert werden, während große Flächen gleicher Struktur nur wenige Knoten zur Darstellung erfordern. Dies betrifft ausgedehnte Freiräume und die noch unbekanntenen Gebiete der Karte. Es sind daher keine besonderen Kartentypen wie die in [SCHRÖTER, 2008] diskutierten On-Demand Karten mehr nötig, um spärliche Umgebungen kartieren zu können, die zwar eine sehr große Fläche abdecken, jedoch nur wenige tatsächliche Informationen enthalten wie etwa bei der Umfahrung eines Hauses.

Bei der Kartierung mit  $N^d$ -Bäumen gelten die gleichen Annahmen wie bei herkömmlichen Gridkarten. Insbesondere werden die Zellen als statistisch unabhängig angenommen und können nur vollständig belegt oder frei sein. Ausgehend von einer leeren Karte in Form eines einzelnen Knotens mit Belegtheitswert 0,5 werden einzelne Entfernungsmessungen eingefügt, indem sie als Strahl  $e = o + rd$  aufgefasst und mittels



**Abbildung 3.3:** Kartierung mit adaptiver Auflösung

Strahlenverfolgung verarbeitet werden. Dabei ist  $o$  die Position des Sensors,  $d$  die Richtung der Messung und  $r$  die ermittelte Distanz [EINHORN et al., 2011]. Neue Knoten werden dabei dynamisch erzeugt (siehe Abbildung 3.3).

## 3.2 Zustandsschätzung

Probabilistisch gesehen kann das SLAM-Problem allgemein wie folgt als rekursive Bayessche Filterung formuliert werden [HÄHNEL et al., 2003]:

$$p(m, x_{1:t} | z_{1:t}, u_{1:t}) = \frac{p(z_t | m, x_t) p(x_t | x_{t-1}, u_t) p(m, x_{1:t-1} | z_{1:t-1}, u_{1:t-1})}{p(z_t)} \quad (3.1)$$

Ziel ist die Schätzung einer Umgebungsrepräsentation  $m$  und der Robotertrajektorie  $x_{1:t}$  basierend auf den zur Verfügung stehenden Daten der Odometrie  $u_{1:t}$  und der Sensorbeobachtungen  $z_{1:t}$ . Üblicherweise wird die rechte Seite der Gleichung in der probabilistischen Robotik als Kombination eines Bewegungsmodells  $p(x_t | x_{t-1}, u_t)$  und eines Beobachtungsmodells  $p(z_t | m, x_t)$  interpretiert.

Das Bewegungsmodell beschreibt die Wahrscheinlichkeit, dass der Roboter bei vorheriger Position  $x_{t-1}$  und angenommener Bewegung  $u_t$  nun bei Position  $x_t$  ist und spezifiziert somit wie sich der Zustand über die Zeit ändert, weshalb es in der Literatur auch Zustandsübergangswahrscheinlichkeit genannt wird [THRUN et al.].

Das Beobachtungsmodell hingegen definiert die Wahrscheinlichkeit an Position  $x_t$  die Beobachtung  $z_t$  machen zu können.

Der Nenner ist von den zu schätzenden Größen unabhängig und stellt somit lediglich einen konstanten Normierungsterm dar. In vielen Darstellungen wird er daher auch direkt durch eine Normierungskonstante  $\eta$  ersetzt. Eine explizite Formulierung ist auch nicht notwendig und würde weitere Betrachtungen nur unnötig verkomplizieren.

Eine vollständige Berechnung der zuvor beschriebenen Wahrscheinlichkeitsverteilung ist jedoch in praktischen Problemstellungen nicht möglich. Daher werden in der Regel stochastische Zustandsschätzer zur Annäherung eingesetzt. Auf die wichtigsten Varianten wird im Folgenden eingegangen.

### 3.2.1 Kalmanfilter

Eine wichtige Familie rekursiver Zustandsschätzer sind die sogenannten Gaußfilter. Der Name ist in der Repräsentation der Schätzung mittels multivariater Normalverteilungen begründet. Einer der ersten und zugleich bedeutendsten Vertreter ist der nach seinem Erfinder benannte Kalmanfilter [KALMAN, 1960], welcher eine effiziente Implementierung des Bayesfilters in kontinuierlichen Zustandsräumen ermöglicht, da die Schätzung mittels Mittelwert und Kovarianzmatrix der Verteilung erfolgt.

Daher werden auch heute noch Varianten des Kalmanfilters vielseitig eingesetzt, obwohl einige inhärente Nachteile bestehen. Die Beschränkung auf Gaußverteilungen impliziert eine Beschränkung auf unimodale Verteilungen. Dies ist gut geeignet für Trackingaufgaben, jedoch insbesondere bei globalen Problemen wie der Lokalisierung

---

problematisch, da verschiedenste, räumlich entfernte Hypothesen zur aktuellen Position existieren können.

Die Beschränktheit des ursprünglichen Algorithmus auf lineare Prozesse wurde hingegen seit der Erstveröffentlichung durch verschiedene Erweiterungen behoben. Ein früherer Ansatz war der vielgenutzte Erweiterte Kalmanfilter [SORENSEN, 1970], welcher auf Linearisierung der nichtlinearen Zustandsübergangs- und Beobachtungsfunktionen beruht.

Eine etwas neuere Entwicklung ist der sogenannte *Unscented Kalman-Filter* [JULIER et al., 2004], der im Gegensatz zum EKF nicht die Modelle approximiert, sondern die über Mittelwert und Kovarianzmatrix spezifizierte Verteilung. Dazu werden deterministisch gewählte Stützpunkte gewählt, welche die gegebenen Parameter ausreichend annähern. Auf diesen können die nichtlinearen Transformationen dann direkt angewendet werden. Wesentliche Vorteile dieses Verfahrens sind höhere Genauigkeit insbesondere bei starker Nichtlinearität und die Eliminierung der im EKF benötigten Funktionsableitungen.

Der größte Nachteil dieser Verfahren ist jedoch die durch die Korrelation der Komponenten des Zustandsvektors bedingte Komplexität von  $O(N^2)$ . Auch wenn die Umgebung auf hervorstechende Merkmale, oft Landmarken genannt, reduziert wird und der Zustandsvektor somit relativ kompakt aus der aktuellen Pose des Roboters und den Landmarkenpositionen aufgebaut werden kann, bedeutet dies bei großen Umgebungen einen signifikanten Aufwand. Wählt man Gridkarten zur Repräsentation der Umwelt vergrößert sich der Zustandsvektor erheblich, da in diesem Falle jede einzelne Zelle Teil des Vektors wird. Kalmanfilter sind daher für die Arbeit mit gridbasierten Karten nicht geeignet. Zwar gibt es Ansätze dies zu umgehen, doch arbeiten neuere Publikationen mit mächtigeren Techniken, die diesen Beschränkungen nicht unterliegen.

---

### 3.2.2 Partikelfilter

Ein Partikelfilter ist eine sequentielle Monte Carlo Näherung der Bayesschen Filterung [DELLAERT et al., 1999]. Hierbei wird die zugrundeliegende Wahrscheinlichkeitsverteilung nicht wie beim Kalmanfilter direkt parametrisch beschrieben, sondern durch eine zufällig gezogene Menge von Stichproben repräsentiert.

Dies ermöglicht es nicht normalverteilte Wahrscheinlichkeitsdichten ohne konzeptionellen Mehraufwand beliebig genau zu schätzen. Zudem muss der Zustandsraum nicht diskretisiert werden wie dies bei Histogrammfiltern notwendig wäre. Ferner kann die Systemdynamik nichtlinear sein und die Schätzung ist nicht auf unimodale Verteilungen beschränkt.

Dennoch kann auch der herkömmliche Partikelfilter hochdimensionale Zustandsräume nur unzureichend effizient abbilden, da mit steigender Anzahl der Zustandskomponenten die Anzahl der benötigten Partikel exponentiell ansteigt.

#### Rao-Blackwellization

Zur Reduzierung der Größe des Zustandsvektors wird in dieser Arbeit daher eine Abwandlung namens *Rao-Blackwellized Particle Filter* (RBPF) verwendet, welche seit einigen Jahren effektiv zur Lösung des SLAM-Problems mit Landmarken (FastSLAM [MONTEMERLO et al., 2002]) und auch Gridkarten [HÄHNEL et al., 2003] eingesetzt wird.

Die Schlüsselerkenntnis dieses bewährten Ansatzes besteht in wichtigen bedingten Unabhängigkeiten. Wird der bisherige Pfad  $x_1, \dots, x_t$  und damit die aktuelle Pose als bekannt vorausgesetzt und als korrekt angenommen, sind die einzelnen Beobachtungen unabhängig voneinander. [MURPHY, 1999]

---

Der RBPF nutzt folgende Faktorisierung aus:

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) = p(m | x_{1:t}, z_{1:t}) p(x_{1:t} | z_{1:t}, u_{1:t-1}) \quad (3.2)$$

Diese Gleichung kann effizient gelöst werden, da  $p(m | x_{1:t}, z_{1:t})$  analytisch berechnet werden kann, sobald  $x_{1:t}$  und  $z_{1:t}$  bekannt sind. Um die A-posteriori-Wahrscheinlichkeitsverteilung  $p(x_{1:t} | z_{1:t}, u_{0:t-1})$  über die möglichen Trajektorien zu schätzen, wird ein Partikelfilter benutzt, der jedem Partikel eine eigene Karte zuordnet. Jede dieser Karten wird erstellt mit den Sensorbeobachtungen  $z_{1:t}$  und den Trajektorien, die das jeweilige Partikel repräsentiert und die mit Hilfe des Bewegungsmodells geschätzt werden. Beim Resampling wird das Gewicht eines Partikels proportional zu der Wahrscheinlichkeit  $p(z_t | m, x_t)$  der letzten Beobachtung  $z_t$  gewählt. [STACHNISS et al., 2005]

### Map-Match SLAM

Während viele Verfahren mit Lasersensorik arbeiten, wurde in [SCHRÖTER, 2008] ein Ansatz entwickelt, der unabhängig vom eingesetzten Sensor, insbesondere jedoch auch mit vergleichsweise ungenauen Techniken wie Ultraschall, funktioniert.

Dabei erhält jedes Partikel eine zusätzliche Karte, in welche ausschließlich die neuesten Beobachtungen einfließen. Diese Karte bildet also stets die unmittelbare *lokale* Umgebung ab. Durch die Vereinigung mehrerer Einzelmessungen enthält sie mehr und genauere Informationen über die Umgebung als die Summe der einzelnen Beobachtungen, da Sensorrauschen gefiltert wird und eine höhere räumliche Auflösung möglich ist.

Die Berechnung der Beobachtungswahrscheinlichkeit  $p(z_t | m, x_t)$  aus Gleichung 3.1 kann dann über einen einfachen Vergleich dieser lokalen Karte mit der aus dem her-

---



kömmlichen RBPF bekannten globalen Karte erfolgen.

In dieser Arbeit wird dies durch die Nutzung eines neuen Verfahrens auf Basis einer Distanztransformation verbessert. Daher sollen im folgenden Abschnitt kurz die Grundlagen dieser Technik erläutert werden.

### 3.3 Distanztransformation

Die Distanztransformation eines (zwei- oder dreidimensionalen) binären Bildes  $m$  weist jedem Element die Distanz zum nächstgelegenen Element mit Wert 1 zu.

Für jedes Pixel oder Voxel  $p$  in  $m$  wird also

$$t(p) = \min_k \{d(p, q_k) | m(q_k) = 1\} \quad (3.3)$$

bestimmt, wobei  $d(p, q_k)$  eine Metrik und  $m(q_k)$  den Wert des Elementes  $q_k$  bezeichnet. Der Index  $k$  läuft dabei über alle Elemente des Ausgangsbildes.

Abbildung 3.4 zeigt das Ergebnis einer Distanztransformation auf einem einfachen Beispielfeld. Zur Vereinfachung der Darstellung wird die quadrierte euklidische Distanz als Metrik verwendet.

Diese Technik wird in verschiedensten Anwendungsbereichen wie der digitalen Bildverarbeitung (Skelettierung, Weichzeichnung), Mustererkennung, Bewegungs- und Pfadplanung eingesetzt. Eine für diese Arbeit wichtige Applikation ist jedoch die Berechnung der Übereinstimmung zweier Bilder wie sie vornehmlich im biomedizinischen Bereich zur Registrierung von Aufnahmen unterschiedlicher bildgebender Verfahren, aber auch bereits in der mobilen Robotik verwendet wird [BIRK und CARPIN, 2006]. Üblicherweise werden die 1-Elemente anschaulich als Hindernisse interpretiert und die restlichen als Freiraum. Das Ergebnis stellt dann eine kompakte und vollständige Kodierung der Entfernungen einzelner Punkte des Raums zum nächsten Hindernis dar.

---

0	1	0	0	0	0	0
0	1	0	0	0	1	0
1	1	0	0	0	1	1
0	1	0	0	1	1	1
0	1	0	0	0	1	1
0	1	0	0	0	1	0
0	1	0	0	0	0	0
0	1	0	0	0	0	0

(a) Binäres Ausgangsbild

1	0	1	4	2	1	2
1	0	1	4	1	0	1
0	0	1	2	1	0	0
1	0	1	1	0	0	0
1	0	1	2	1	0	0
1	0	1	4	1	0	1
1	0	1	4	2	1	2
1	0	1	4	5	2	5

(b) Ergebnis

**Abbildung 3.4:** Beispiel einer Distanztransformation

Als Metrik kommt die quadrierte euklidische Distanz zum Einsatz. Es ist leicht nachzuprüfen, dass jedes Element im Ergebnisbild (b) die Entfernung zum nächstgelegenen 1-Element aus dem Ausgangsbild (a) enthält.

# Kapitel 4

## Der SLAM-Algorithmus

Im vorangegangenen Kapitel wurden verschiedene Verfahren zur Umgebungsdarstellung und Positionsschätzung erläutert. Dabei wurden insbesondere auch die im Fachgebiet geleisteten Vorarbeiten beschrieben. Zusätzlich wurde das Prinzip der Distanztransformation vorgestellt. In diesem Kapitel soll nun dargelegt werden, wie diese Ansätze in dem im Rahmen dieser Arbeit implementierten SLAM-Modul kombiniert und implementiert wurden. Zudem wird eine Auswertung der Implementierung anhand von Testdaten vorgenommen, die mit einem Roboter im Konrad-Zuse-Bau der Technischen Universität Ilmenau aufgenommen wurden.

### 4.1 Umgebungsrepräsentation

Zur effizienten Erstellung einer Karte der Umwelt wurde auf das in [EINHORN et al., 2011] entwickelte Verfahren zur Kartierung mit Hilfe einer hierarchischen Repräsentationsform namens *N<sup>d</sup>-Bäumen* zurückgegriffen. Eine genaue Definition dieser Datenstruktur und weitere theoretischen Aspekte wurden bereits in Abschnitt 3.1.2 beschrieben.

---

Die flexible Struktur eines Baumes ermöglicht es die Auflösung der Karte problemlos dynamisch anzupassen, indem bestehende Knoten entweder weiter aufgeteilt oder nicht mehr benötigte Kindknoten zu einem verschmolzen werden. Somit können Gebiete hohen Interesses sehr genau repräsentiert werden während größere Flächen mit wenigen grob aufgelösten Zellen abdeckbar sind. Ferner kann die Karte ohne teure Kopieroperationen in konstanter Rechenzeit vergrößert werden.

Dadurch ist es möglich den Speicherverbrauch insbesondere im dreidimensionalen Raum im Vergleich zu herkömmlichen Gridkarten massiv zu verringern. Zusätzlich werden auch weniger Rechenoperationen benötigt, da weniger Zellen durchlaufen werden müssen.

Insgesamt konnten in Testläufen im Vergleich zu einer herkömmlichen Belegtheitskarte mit fester Auflösung über 60% der benötigten Zellen eingespart und damit auch die Anzahl der anfallenden Zelloperationen bei Kartenaktualisierungen um mindestens 50% verringert werden [EINHORN et al., 2011].

Die Nutzung einer bezüglich der Dimensionalität allgemein definierten Datenstruktur begünstigt ferner eine generische Implementierung des SLAM-Moduls, die leicht sowohl für den zweidimensionalen als auch den dreidimensionalen Fall anwendbar ist.

Zum Aufbau einer solchen Karte können momentan Sensoren verwendet werden, deren Daten sich in einzelne Entfernungsmessungen zerlegen lassen (siehe Abschnitt 3.1.2). In dieser Arbeit wurden vor allem 2D-Laser-Messungen zur Kartierung genutzt. Für die Erstellung dreidimensionaler  $N^d$ -Bäume können Tiefenbilder eines Kinect-ähnlichen Gerätes genutzt werden.

---

## 4.2 Lokalisierung

Das implementierte Modul basiert auf dem in [SCHRÖTER, 2008] beschriebenen *Map-Match-SLAM* Verfahren. Dessen Herzstück ist ein *Rao-Blackwellized Partikelfilter* (siehe Abschnitt 3.2.2).

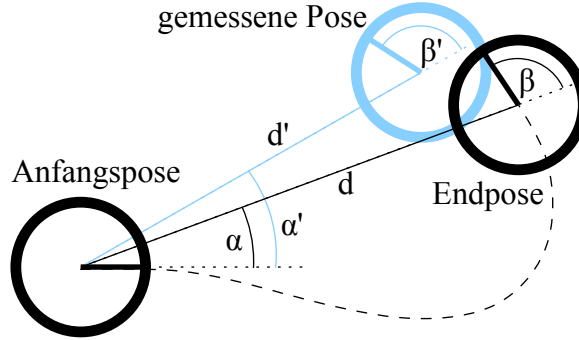
Die Bestandteile des Filters sind bei der Implementierung unmittelbar auf entsprechende Klassen übertragbar und ermöglichen somit eine moderne, objektorientierte Codestruktur. In Folge der berechnungstechnischen Unabhängigkeit der einzelnen Partikel ist der Algorithmus zudem mühelos parallelisierbar, wodurch die immer wichtiger werdenden Mehrkernarchitekturen vollständig ausgelastet werden können. Mit Hilfe einer Schnittstelle wie OpenMP erfordert dies nicht mehr als eine einzige Zeile Code, welche zu den berechnungsintensiven Schleifen über allen Partikeln hinzugefügt wird.

### 4.2.1 Bewegungsmodell

Zur Berechnung des Bewegungsmodells  $p(x_t|x_{t-1}, u_t)$  kommt eine leicht abgewandelte Form des in [SCHRÖTER, 2008] auf Basis von [HÄHNEL et al., 2003] entwickelten Bewegungsmodells zum Einsatz, da es sowohl intuitiv verständlich ist als auch genügend Raum zur Parametrisierung bietet. Die Bewegung des Roboters wird zerlegt in eine anfängliche reine Drehbewegung  $\alpha$  relativ zur Anfangsausrichtung, eine darauffolgende translatorische Bewegung mit Länge  $d$  und eine abschließende Drehung  $\beta$  in die finale Pose (siehe Abbildung 4.1).

Dabei wird die Unsicherheit der einzelnen Teilbewegungen mittels drei Varianzparametern bezüglich der aus den Odometriedaten gewonnenen Werten  $d'$ ,  $\alpha'$  und  $\beta'$  modelliert:

---



**Abbildung 4.1:** Schematische Darstellung der Komponenten des Bewegungsmodells

Die tatsächliche Bewegung des Roboters ist gestrichelt dargestellt und wird durch das schwarz gezeichnete Modell angenähert. Durch Odometriefehler wurde jedoch eine andere Position mit entsprechend veränderten Parametern  $\alpha'$ ,  $d'$  und  $\beta'$  gemessen. Angepasste Version des Schemas aus [HÄHNEL et al., 2003].

$$d \sim d' \cdot \mathcal{N}(1, \sigma_d) \quad (4.1)$$

$$\alpha \sim \alpha' + (\text{abs}(\alpha') + \text{abs}(\beta')) \cdot \mathcal{N}(0, \sigma_{\text{turn}}) + d' * \mathcal{N}(0, \sigma_{d,\text{turn}}) \quad (4.2)$$

$$\beta \sim \beta' + (\text{abs}(\alpha') + \text{abs}(\beta')) \cdot \mathcal{N}(0, \sigma_{\text{turn}}) + d' * \mathcal{N}(0, \sigma_{d,\text{turn}}) \quad (4.3)$$

Sowohl Richtungsänderung als auch abschließende Drehung fließen in die Unsicherheit beider Drehungen ein, da diese in der Praxis nicht gänzlich unabhängig voneinander sind. Zusätzlich wirkt sich die Länge der gefahrenen Strecke unsicherheitsvergrößernd aus.

Die Parameter sind in Abhängigkeit vom verwendeten Robotersystem und dem Einsatzgebiet zu bestimmen. Da sich eine genaue Bestimmung jedoch äußerst aufwendig gestalten kann, wurden diese empirisch gewählt. Dabei muss nur darauf geachtet werden die Varianzen nicht zu unterschätzen, da anderenfalls die Positionsschätzung auf Grund von zu großem Vertrauen in die Genauigkeit der Odometriemessungen so weit vom tatsächlichen Ort abweichen kann, dass im Falle eines Kreisschlusses dieser nicht

mehr erkannt werden kann. Zu hohe Werte führen lediglich zu erhöhter Posenunsicherheit, wodurch mehr Partikel benötigt werden, aber dennoch korrekte Karten entstehen. Beim Bewegungsupdate entnimmt dann jedes Partikel unabhängig eine Stichprobe aus der oben beschriebenen Verteilung. Auf Basis dieser geschätzten Bewegung des Roboters wird sodann die jeweilige Pose aktualisiert:

$$x_{t+1} = x_t + d \cdot \cos(\phi_t + \alpha) \quad (4.4)$$

$$y_{t+1} = y_t + d \cdot \sin(\phi_t + \alpha) \quad (4.5)$$

$$\phi_{t+1} = \phi_t + \alpha + \beta \quad (4.6)$$

## 4.3 Resampling

Das Resampling ist der wichtigste Teil eines jeden partikelfilterbasierten SLAM-Algorithmus, da es dessen funktionale und nichtfunktionale Eigenschaften maßgeblich beeinflusst. In diesem Abschnitt werden daher die hierfür genutzten, im Rahmen dieser Arbeit entwickelten Methoden vorgestellt.

### 4.3.1 Kartenübereinstimmung

Die Qualität der durch das SLAM-Verfahren erstellten Karten hängt wesentlich von dem verwendeten Ansatz zur Berechnung der Übereinstimmung von lokaler und globaler Karte eines Partikels ab.

In [SCHRÖTER, 2008] wurde ein ausschließlich mit den reinen Belegtheitswerten arbeitender Ansatz gewählt, welcher über alle Zellen der lokalen Karte iteriert und den jeweiligen Inhalt mit der korrespondierenden Zelle in der globalen Karte nach folgender

---

Formel vergleicht:

$$match = \sum_{cell \in map} \left\{ \begin{array}{l} 1, \quad falls(cell_{loc} > 0.6) \wedge (cell_{glob} > 0.6) \\ -1, \quad falls(cell_{loc} > 0.6) \wedge (cell_{glob} < 0.4) \\ \quad \quad oder(cell_{loc} < 0.4) \wedge (cell_{glob} > 0.6) \\ 0, \quad \quad \quad \quad \quad \quad \quad \quad sonst \end{array} \right\} \quad (4.7)$$

Dies ist äußerst effizient berechenbar und wurde bereits erfolgreich in großen Umgebungen von mehreren tausend Quadratmetern eingesetzt. Zunächst wurde dieses Verfahren an die neue Form der Kartenrepräsentation mittels  $N^d$ -Baum angepasst und implementiert. Als problematisch erwies sich allerdings, dass sich Knoten der beiden Karten beliebig überlappen könnten und für jedes Blatt der lokalen Karte somit der komplette globale Baum nach all jenen Blättern durchsucht werden müsste, deren zugehöriger Teilraum den des lokalen Knotens schneidet. Dieser naive Ansatz besitzt also eine Laufzeit von  $O(M * N)$ , wobei  $M$  die Anzahl der Blätter in der lokalen und  $N$  die Anzahl der Knoten in der globalen Karte bezeichnet. Besonders teuer ist dies, da sehr viele Rekursionen auftreten und somit für den Compiler wenig Raum zur Optimierung bleibt.

Daher wurde zur Optimierung dafür gesorgt, dass beide Karten eines jeden Partikels gleichermaßen mit einem einzelnen Knoten so initialisiert werden, dass sie den gleichen Versatz zum Referenzkoordinatensystem aufweisen. Da im Zuge der darauffolgenden Kartenaktualisierungen neue Knoten ausschließlich durch Aufteilung der vorhandenen Blätter oder Erweiterung des aktuellen Wurzelknotens durch Geschwister und eine zusätzliche Elternschicht erstellt werden, ändert sich dieser Versatz im Verlaufe der Zeit nicht. Somit existiert zu jedem Knoten in der lokalen Karte ein zugehöriger in der globalen, welcher entweder den gleichen Teil des Raums abdeckt oder diesen vollständig enthält. Diese Annahme ermöglicht eine effiziente Implementierung durch gleichzeitiges

---



Absteigen in den zueinander gehörenden Teilbäumen beider Karten.

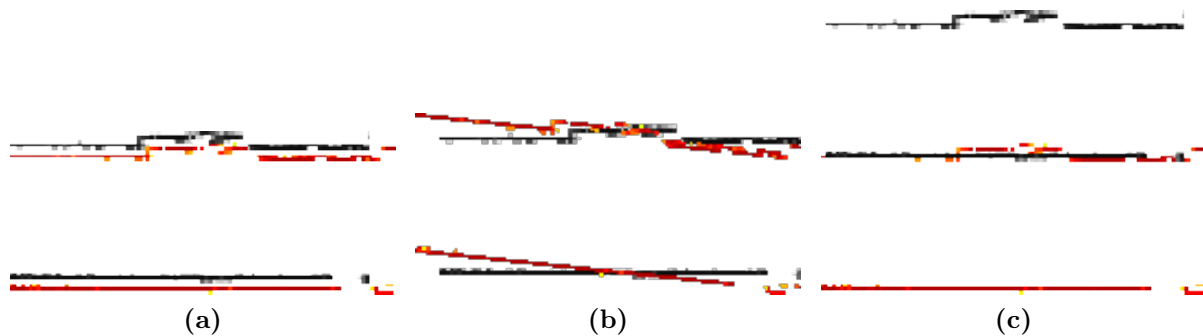
Als zusätzliche Optimierung wird in einem Vorverarbeitungsschritt die globale Karte auf den tiefsten Teilbaum reduziert, welcher die von der lokalen Karte abgedeckte Fläche noch vollständig einschließt. Dies kann äußerst effizient als Schleife verwirklicht werden, da lediglich ein einzelner Pfad vom Wurzelknoten ausgehend zum Zielknoten hin beschritten und ein Vergleich der zugehörigen Teilräume durchgeführt werden muss. Existiert ein Kindknoten, welcher den gesuchten Raum vollständig enthält, wird dieser Knoten für die nächste Iteration ausgewählt. Anderenfalls ist der aktuelle Knoten bereits der gesuchte und die Schleife wird abgebrochen.

Der größte Nachteil dieser ausschließlich mit den Belegtheitswerten arbeitenden Methode ist jedoch, dass bei der Berechnung keinerlei Informationen über den eigentlichen räumlichen Versatz der lokalen und globalen Karte einbezogen werden. Das führt insbesondere dazu, dass bereits bei kleinen Abweichungen große Unterschiede des Kartenübereinstimmungsmaßes und andererseits bei einem großem Fehler unter Umständen falsche Werte auftreten können, siehe Abbildung 4.2.

Dies wird zusätzlich erschwert durch die mit dem  $N^d$ -Baum mögliche, lokal erheblich höhere Auflösung. Wände können wesentlich dünner repräsentiert sein als bei einer herkömmlichen Belegtheitskarte.

Infolgedessen wurde daher ein alternatives Maß entwickelt, welches die räumliche Abweichung von korrekter und geschätzter Position berücksichtigt und stetig mit steigendem Abstand abfällt. Für die Ermittlung dieses Versatzes als Ausgangswert wurde auf das aus der Bildverarbeitung bekannte Verfahren der Distanztransformation zurückgegriffen, da hierfür bewährte Algorithmen existieren (siehe Abschnitt 3.3). Der Nachteil dieser Ansätze ist allerdings, dass sie eine gleichmäßige Zellstruktur, wie sie in Bildern vorhanden ist, voraussetzen. Es müsste also zunächst aus dem  $N^d$ -Baum eine herkömmliche Gridkarte erzeugt werden, um ein Distanzbild berechnen zu können. Da

---



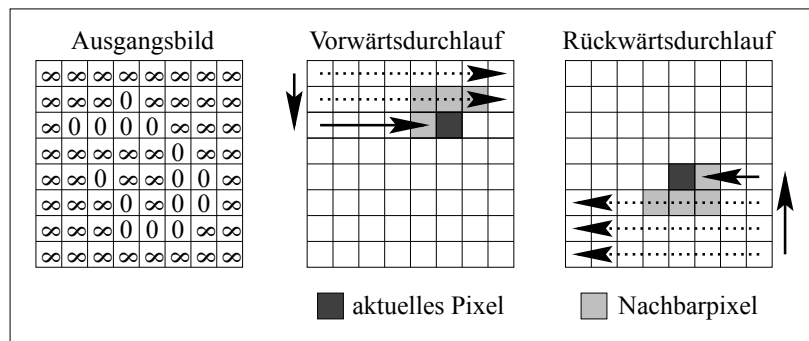
**Abbildung 4.2:** Typische Beispielsituationen, in denen eine ausschließliche Betrachtung von Belegtheitswerten irreführende Werte liefert  
 (a) Obwohl der Roboter seine Ausrichtung fehlerfrei geschätzt und nur einen geringen Versatz zur korrekten Position hat, ergibt ein einfacher Vergleich einen stark verminderten Wert, da alle vier Wände negativ in die Berechnung eingehen. (b) Dieser Fall kann anhand des Übereinstimmungswertes kaum von Fall a unterschieden werden. (c) Diese ebenfalls sehr häufig auftretende Situation wird je nach Größe der lokalen Karte sogar noch besser als die ersten beiden Varianten bewertet, obwohl sie eindeutig einen größeren Fehler darstellt.

die lokale Karte in ihren räumlichen Ausmaßen jedoch stets beschränkt ist, bietet sich an, für die lokale Karte von vornherein auf normale Gridkarten zurückzugreifen.

Es existiert mittlerweile eine Vielzahl äußerst effizienter Algorithmen zur exakten oder annäherungsweisen Berechnung der Distanztransformation in linearer Zeit [BORGEFORS, 1986, FELZENSZWALB und HUTTENLOCHER, 2004]. Die Grundidee der approximativen Ansätze ist die Ersetzung der kostenintensiven globalen Bestimmung des nächstgelegenen 1-Pixels durch sukzessive Propagierung von Entfernungen einzelner Pixel in einer lokal begrenzten Nachbarschaft.

Der letztlich implementierte Algorithmus basiert auf dem in [BORGEFORS, 1986] vorgestellten Verfahren, welches auch in der OpenCV-Bibliothek [OPENCV, 2011] eingesetzt wird. Dieses bietet eine hinreichend genaue Annäherung der tatsächlichen euklidischen Entfernung bei vertretbarem Berechnungsaufwand.

Vor der Transformation wird die Karte über eine Schwellwertoperation in ein binäres Format überführt. Infolgedessen sind Hindernisse bereits korrekt mit dem Wert 0 in-



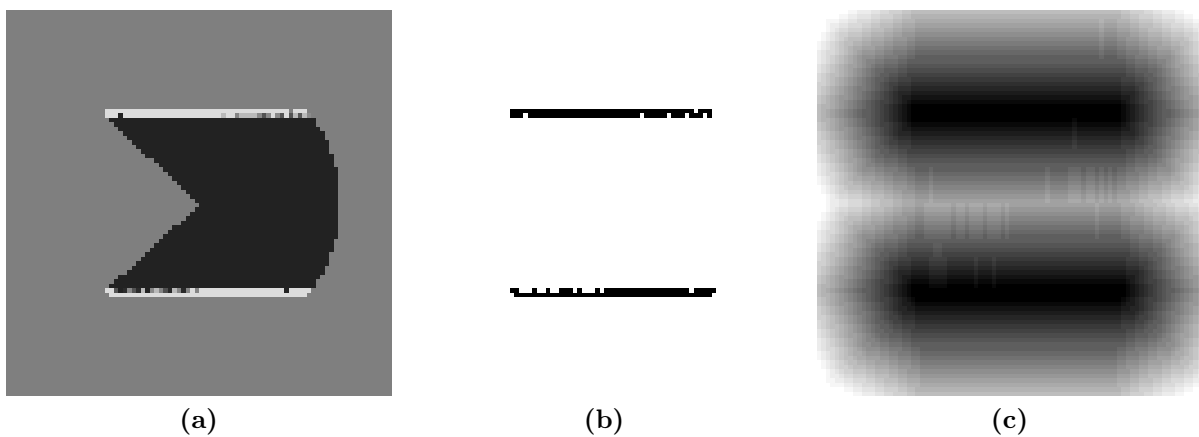
**Abbildung 4.3:** Schema zur Distanztransformation

Der neue Wert des „aktuellen Pixels“ nach jedem Lauf ist das Minimum der Summen von Entfernung und momentanem Wert der Nachbarpixel und des Pixels selbst. Modifizierte Version einer Abbildung aus [BIRK und CARPIN, 2006].

initialisiert. Unbekanntes Gebiet und Freiraum wird auf  $\infty$  gesetzt. Sodann werden auf der Karte zwei Berechnungsdurchläufe sequentiell durchgeführt (siehe Abbildung 4.3).

Ein ähnlicher Ansatz wurde bereits in [BIRK und CARPIN, 2006] zur Vereinigung mehrerer Belegtheitskarten derselben Umgebung, welche in diesem Falle durch einen Roboterschwarm erzeugt wurden, verwendet. Da der Ansatz offensichtlich für den Offlineeinsatz konzipiert wurde, sind allerdings Transformationen auf globaler und lokaler Karte vorgesehen. Durch die benötigte hohe Anzahl an Partikeln ist dies jedoch nicht in Echtzeit berechenbar. In dieser Arbeit wurde die Transformation daher lediglich auf der lokalen und damit stets in ihrer Größe stark begrenzten Karte berechnet. Als Ergebnis dieses Prozesses ergibt sich eine neue Karte, die alle für nachfolgenden Berechnungen erforderlichen Informationen beinhaltet (siehe Abbildung 4.4).

Da die Distanztransformation reine Abstandswerte liefert, die zudem in Gridzellen bemessen sind, müssen diese vor der weiteren Verwendung in Meter umgerechnet und so umgeformt werden, dass die für die spätere Partikelgewichtung bestimmten Endwerte mit zunehmender Distanz abnehmen. Auf Grund der einfachen Berechenbarkeit und



**Abbildung 4.4:** Beispiel einer einfachen Umgebungskarte und der errechneten Distanztransformation

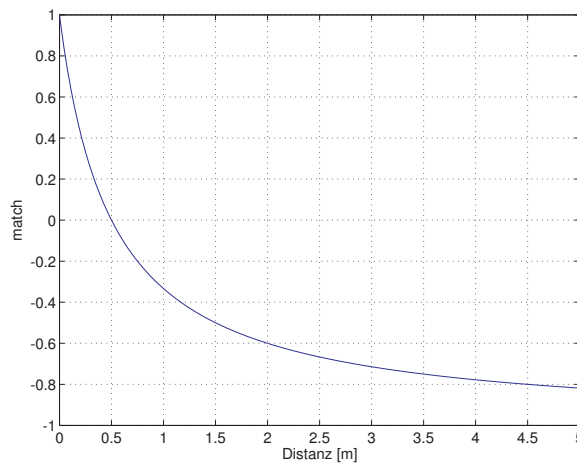
(a) Zur Demonstration wurde mit einem synthetischen Laserscan eine einfache Belegtheitskarte erstellt. Die weißen Pixel stellen Hindernisse dar und die schwarzen Freiraum. Dem Roboter noch unbekannte Bereiche sind grau eingefärbt. (b) Nach der initialen Schwellwertoperation sind Hindernisse auf den Wert 0 (schwarz) und der restliche Raum auf  $\infty$  (weiß) gesetzt. Auf diesem Bild operiert sodann der Algorithmus zur Berechnung der Distanztransformation. (c) Ergebnis der Distanztransformation.

Parametrisierbarkeit wurde die Funktion

$$match = \frac{1}{d + 0.5} - 1 \quad (4.8)$$

gewählt.

Damit ergibt sich bei exakter Übereinstimmung zweier Zellen ein Matchwert von genau 1. Bis zu einem Versatz von 0,5 Metern fällt das Maß stetig auf 0 ab und geht sodann ins Negative über (siehe Abbildung 4.5).

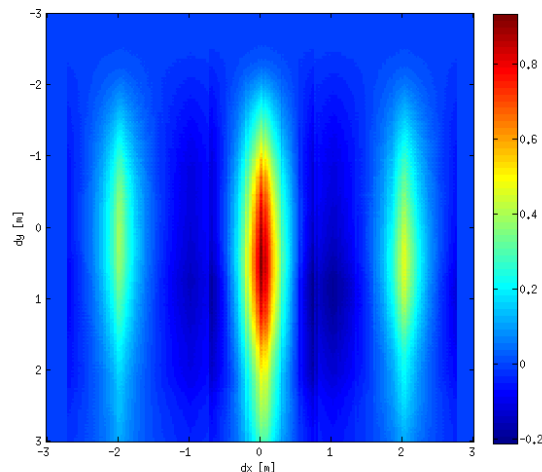


**Abbildung 4.5:** Umrechnung von Distanz in Matchwert

Zur Berechnung des Gesamtübereinstimmungswertes  $match^p$  wird die globale Karte rekursiv nach Blättern durchsucht. Repräsentiert ein Blatt ein Hindernis und liegt es in dem von der lokalen Karte abgedeckten Raum, so werden aus der zuvor berechneten Distanzkarte alle im Bereich des Blattes befindlichen Elemente extrahiert, mittels obiger Formel in Matchwerte umgerechnet und aufsummiert. Abschließend wird diese Summe noch mittels der Anzahl gefundener globaler Hindernis-Blätter normiert.

Zur Veranschaulichung der Wirkungsweise der Distanztransformation in den Berechnungen zur Kartenübereinstimmung wurde eine synthetische Resamplingsituation er-

zeugt, indem der einfache Korridor aus Bild 4.4 links direkt als globale Karte und mit unterschiedlich starken räumlichen Versatz als lokale Karte benutzt wurde. Abbildung 4.6 zeigt die sich ergebenden Matchwerte farblich kodiert.



**Abbildung 4.6:** Ergebnis der Kartenübereinstimmung bei unterschiedlich starkem räumlichem Versatz

*Als Basis wurde der künstliche Gang aus Abbildung 4.4 verwendet. Die vertikale Achse stellt den Versatz in Richtung des Korridors dar während die x-Achse orthogonal dazu verläuft. Entsprechend sind Bewegungen entlang der y-Achse weniger problematisch, da ein Großteil der Wände noch immer aufeinander liegen.*

### 4.3.2 Partikelgewichtung

Das eigentliche Gewicht eines Partikels berechnet sich aus dem Übereinstimmungswert seiner globalen und lokalen Karte über eine einfache Exponentialfunktion:

$$w_{match}^p = e^{c \cdot match^p} \quad (4.9)$$

Der Grund für diese Berechnung ist eine zusätzlich eingeführte Streuung der Gewichtsverteilung aller Partikel, die auf einfache und intuitive Weise über den Parameter  $c$  beeinflusst werden kann. Je höher dieser Wert gewählt wird, desto stärker unterschei-

den sich die relativen Gewichte der einzelnen Partikel. Das abschließende Resampling erfolgt nicht stochastisch, sondern es wird dazu zunächst das jeweilige relative Gewicht bezüglich der Summe aller Partikelgewichte und sodann dynamisch ein Multiplikator  $M_p$  für jedes Partikel bestimmt, der angibt, wie oft dieses Partikel für die neue Verteilung zu kopieren ist.

Da diese Multiplikatoren logischerweise nur ganzzahlig sein können und somit ein gewisser Anteil an der Verteilung verloren gehen könnte, wird zusätzlich ein persistentes Gewicht berechnet, welches diese Informationen bewahrt [SCHRÖTER, 2008]:

$$w_t^p = \frac{w_{match}^p}{\sum_{p=1}^P w_{match}^p} \frac{P}{M_p} w_{t-1}^p \quad (4.10)$$

Diese Gewichte sind sodann Ausgangswerte für den im folgenden Pseudocode beschriebenen Algorithmus zum Durchführen des Resamplings einschließlich Bestimmung der benötigten Mindestanzahl an Partikeln.

**Eingaben**

1  $P = \{(x_1, w_1), \dots, (x_N, w_N)\}$  //  $N$  Partikel mit Zustand  $x_i$  und Gewicht  $w_i$

**Initialisierung**

2  $P_{neu} \leftarrow \emptyset$ ; // neue Partikelmenge  
 3  $M_{1:N} \leftarrow 0$ ; // Partikelmultiplikatoren  
 4  $N_{neu} \leftarrow 0$ ; // Anzahl neuer Partikel  
 5  $W \leftarrow \sum_{j=1:N} w_j$ ; // Summe der Partikelgewichte  
 6  $b_{1:k} \leftarrow 0$ ; // Binbelegung  
 7  $B \leftarrow 0$ ; // Anzahl belegter Bins

**Algorithmus**

/\* Multiplikatoren und  $N_{neu}$  bestimmen \*/  
 8 Solange  $N_{neu} < N_{min}$  oder  $N_{neu} \leq 2 * B$  oder  $N_{neu} < 0.5 * N$   
 9 Für  $i = 1 : N$   
 10  $d_i = \frac{w_i}{W} - \frac{M_i}{N_{neu}}$  // Differenz zwischen erwünschtem und  
 11 // tatsächlichem Anteil an der Verteilung  
 12  $i' = \operatorname{argmax}(d_i)$   
 13  $M_{i'} = M_{i'} + 1$   
 14  $N_{neu} = N_{neu} + 1$   
 15  $l = \operatorname{bin}(x_{i'})$  // Bin-Index für Partikel  $i'$   
 16 Wenn  $b_l = 0$ , dann  
 17  $b_l = 1$  // belege Bin  
 18  $B = B + 1$   
 /\* Aktualisierung der Gewichte und Erstellung der neuen Partikelmenge \*/  
 19 Für  $i = 1 : N$   
 20  $w_i = \frac{w_i}{W} \cdot \frac{N_{neu}}{M_i}$  // berechne persistentes Gewicht  
 21 Für  $j = 1 : M_i$  // Kopiere Partikel  $M_i$  mal  
 22  $P_{neu} = P_{neu} \cup P_i$

**Rückgabe**  
 23  $P_{neu} = \{(x_1, w_1), \dots, (x_{N_{neu}}, w_{N_{neu}})\}$  // neue Partikelmenge

---

**Abbildung 4.7:** Pseudocode für das Resampling der Partikel einschließlich dynamischer Bestimmung der benötigten Anzahl basierend auf [SCHRÖTER, 2008]

---

## 4.4 Experimentelle Untersuchungen

Zur Überprüfung der korrekten Funktionsweise des Verfahrens wurde das implementierte Modul auf einem im Fachgebiet eingesetzten Testroboter installiert und Teile der Innenräume des Konrad-Zuse-Baus der Technischen Universität Ilmenau kartiert.

---



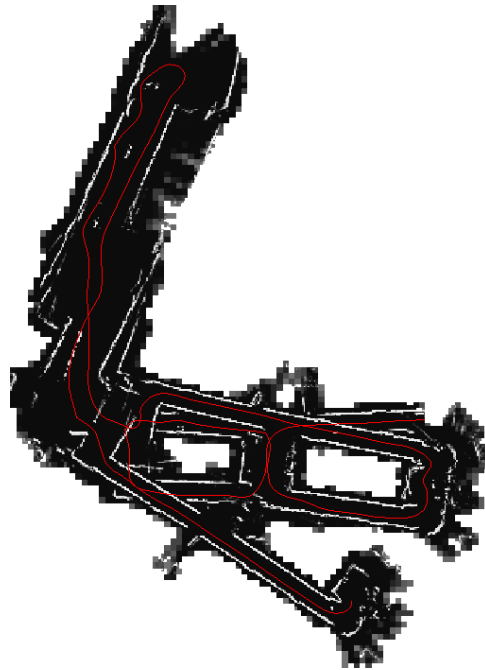
Abbildung 4.8 zeigt das Ergebnis bei ausschließlicher Nutzung der vom Roboter gelieferten Odometriedaten. Schon bei einer so geringen Fahrtstrecke von etwa 160 Metern ergeben sich massive Abweichungen in der Karte durch die stetige Akkumulation von Messfehlern.

Beim Vergleich mit dem Ergebnis des SLAM-Systems wird dies besonders deutlich. Abbildung 4.9 zeigt, dass eine konsistent Karte der Umgebung aufgebaut werden konnte. Es wird korrekt wiedergegeben, dass Anfangs- und Endpunkt der Bewegung sehr nahe beieinander liegen.

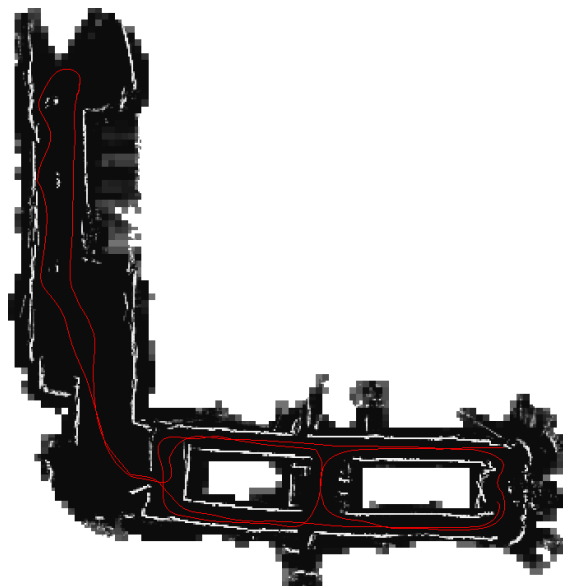
In Abbildung 4.10 ist die dazugehörige Entwicklung der Anzahl der benötigten Partikel und des Speicherverbrauchs über den Verlauf des Versuchs abgetragen. Dabei wurde eine untere Schranke von 50 und eine obere Schranke von 500 Partikeln festgelegt. Es ist erkennbar wie die Kurve anfangs schnell auf den Minimalwert abfällt, da noch keine Unsicherheit vorhanden ist. Die beiden darauffolgenden Phasen veranschaulichen die vom Roboter durchgeführten Explorationsphasen, welche die Unsicherheit der Positionsschätzung deutlich erhöhen und mehr Partikel zur Repräsentation der Wahrscheinlichkeitsverteilung nötig machen, sowie die jeweils zur Entspannung führenden Kreisschlüsse.

Ein großes Problem ist der immense Speicherverbrauch des Systems. Es konnte im Rahmen dieser Arbeit nicht abschließend geklärt werden, worin die Ursache der Diskrepanz zwischen rechnerischer Größe der globalen Karten und tatsächlichem Speicherverbrauch der Testumgebung liegt. Eine mögliche Ursache könnte Fragmentierung des Hauptspeichers sein, da Knoten in den  $N^d$ -Bäumen sehr kleine Speichereinheiten darstellen. Aber auch die Nutzung eines Pool-Allokators mit der auch in Boost zum Einsatz kommenden *Simple Segregated Storage* konnte das Problem nicht vollständig beheben.

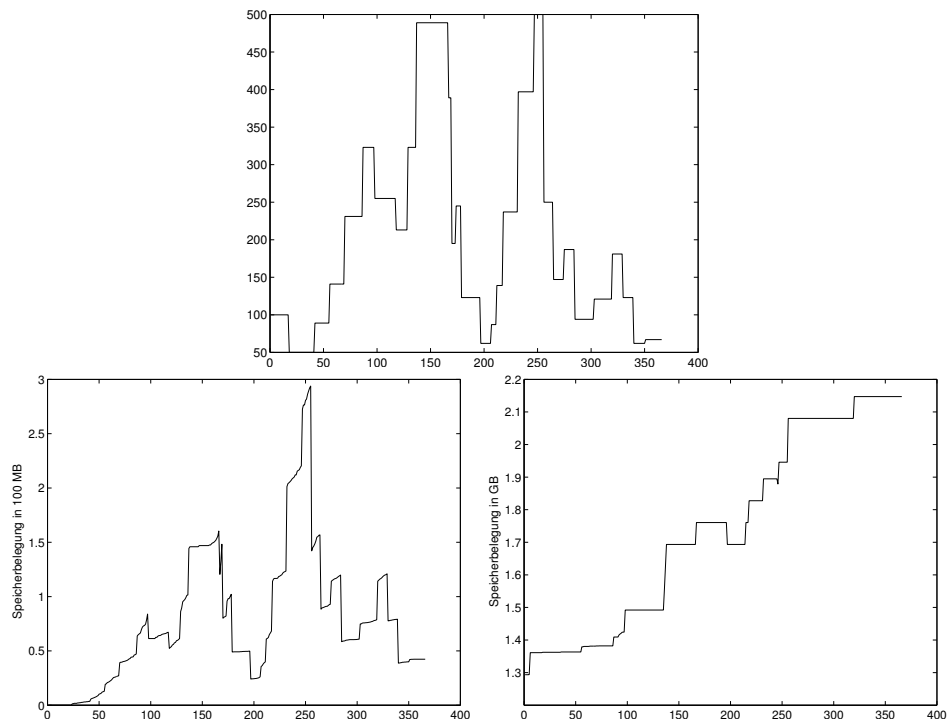
---



**Abbildung 4.8:** Umgebungskarte bei ausschließlicher Nutzung der Odometriedaten  
*Die Trajektorie des Roboters ist rot (bzw. grau) eingezeichnet.*



**Abbildung 4.9:** Ausgabe des SLAM-Algorithmus



**Abbildung 4.10:** Anzahl der benötigten Partikel und Speicherverbrauch im Verlauf der Umgebungserkundung

Die abgetragenen Werte wurden in regelmäßigen Zeitabständen von ungefähr einer Sekunde gemessen. Daher entspricht die horizontale Achse in etwa dem zeitlichen Verlauf der Untersuchung. Im oberen Diagramm ist die jeweilige Anzahl der Partikel dargestellt. Die beiden unteren Abbildungen zeigen den rechnerischen Speicherverbrauch der globalen Karten aller Partikel in Form von  $N^d$ -Bäumen und den wesentlich höheren tatsächlichen Gesamtverbrauch der Testumgebung.



# Kapitel 5

## Zusammenfassung und Ausblick

### 5.1 Zusammenfassung

In dieser Arbeit wurde ein Verfahren zur global konsistenten Modellierung der Umwelt mittels einer generischen, hierarchischen Repräsentationsform von Belegtheitskarten und gleichzeitiger verlässlicher Lokalisierung im Raum präsentiert. Dieses stellt eine konsequente Weiterentwicklung der im Fachgebiet Neuroinformatik und Kognitive Robotik geleisteten Vorarbeiten dar. Das bewährte Map-Match-SLAM-Verfahren wurde mit den sogenannten  $N^d$ -Bäumen als effektive Umgebungsrepräsentationsform verknüpft und aufbauend auf dem Prinzip der Distanztransformation der Resampling-Schritt verbessert.

Der Algorithmus wurde als Modul für das ebenfalls am Fachgebiet in Entwicklung befindliche RSI-Framework implementiert und auf einem mobilen Roboter in Innenraumszenarios getestet. Es wurde demonstriert, dass das System in der Lage ist, global konsistente Karten der Umgebung zu erstellen.

---

## 5.2 Weiterführende Arbeiten

Dieses Kapitel soll einen Ausblick auf zukünftige Weiterentwicklungsmöglichkeiten des Systems geben. Es werden einige Kritikpunkte am momentanen System besprochen, deren Bearbeitung zwar bereits begonnen hat, welche jedoch noch nicht vollständig implementiert und getestet werden konnten.

### 5.2.1 Anwendung auf den dreidimensionalen Fall

Das vorgestellte Verfahren wurde von Beginn so konzipiert und implementiert, dass es relativ einfach auf die Nutzung dreidimensionaler Entfernungsmesser umgestellt werden kann. Notwendig für eine dreidimensionale Umgebungskartierung ist die Erweiterung der Distanztransformation um eine Dimension und die Nutzung einer 3D-Belegtheitskarte, auch Voxelkarte genannt. Die tatsächliche Umsetzung konnte aus Zeitgründen jedoch nicht mehr fertig gestellt werden.

### 5.2.2 Verbesserung der Distanztransformation

Insbesondere im einfacheren zweidimensionalen Fall ist die Nutzung einer herkömmlichen Belegtheitskarte für die lokalen Beobachtungen relativ unproblematisch und ermöglichte die Verwendung eines simplen, aber effektiven Algorithmus zur Berechnung der Distanztransformation. Dies steht jedoch im Widerspruch zu dem gesetzten Ziel Umgebungen unabhängig von der gewünschten Kartendimensionalität speicher- und berechnungseffizient behandeln zu können.

Ideal wäre ein Verfahren, welches direkt auf einem  $N^d$ -Baum operiert und Daten für die Berechnung der Kartenübereinstimmung liefert.

---

### 5.2.3 Partikelschwund

Der Resampling-Schritt hat wesentlichen Einfluss auf die Leistungsfähigkeit des Partikelfilters. Partikel mit geringem relativen Gewicht werden entfernt und jene mit hohem Gewicht vermehrt. Dabei ist die Implementierung bestrebt, stets nur so viele Partikel beizubehalten wie zur Repräsentation der Wahrscheinlichkeitsverteilung notwendig sind. Bei zu aggressivem Verhalten kann dies jedoch schnell dazu führen, dass eigentlich gute Schätzungen gelöscht werden. Ebenso führen wiederholt gefahrene lokale Schleifen zu einer Unterschätzung der globalen Positionsunsicherheit. Diese Problematik wird in der Literatur oftmals Partikelschwund genannt (englisch *Particle Depletion* oder *Particle Impoverishment*).

Besonders bei den in dieser Arbeit verwendeten Belegtheitskarten muss dies berücksichtigt werden, da eine nachträgliche Korrektur bestehender Karten im Gegensatz zu topologischen Karten eher schwierig ist und sich einmal eingeführte Fehler in der Karte fortpflanzen. Es ist daher zu prüfen, inwiefern das Resampling in dieser Hinsicht verbessert werden kann. Eine detaillierte Analyse hierzu bezüglich des FastSLAM-Algorithmus kann in [KWAK et al., 2007] nachgelesen werden. Lösungsansätze für Rao-Blackwellized Partikelfilter in Kombination mit Gridkarten wurden beispielsweise in [STACHNISS et al., 2005] und [GRISSETTI et al., 2007] vorgestellt. Erstgenanntes Verfahren führt eine zusätzliche topologische Karte ein, um aktiv Schleifen zu schließen und achtet beim Abschluss darauf, die Partikelvielfalt aufrecht zu erhalten. Letzteres hingegen stellt ein Kriterium vor, um den Zeitpunkt des Resamplings adaptiv zu wählen.

---

### 5.2.4 Reduzierung des Speicherbedarfs

Der Rao-Blackwellized Partikelfilter ist ein exzellentes Mittel zur Lösung des SLAM-Problems, insbesondere, da es ein recht einfaches und leicht zu implementierendes Konzept ist. Diese Vorteile werden jedoch damit erkauft, dass jedes Partikel eine eigene Karte der Umgebung aufbauen muss, was besonders im dreidimensionalen Fall zu einem erheblichem Speicherbedarf führt.

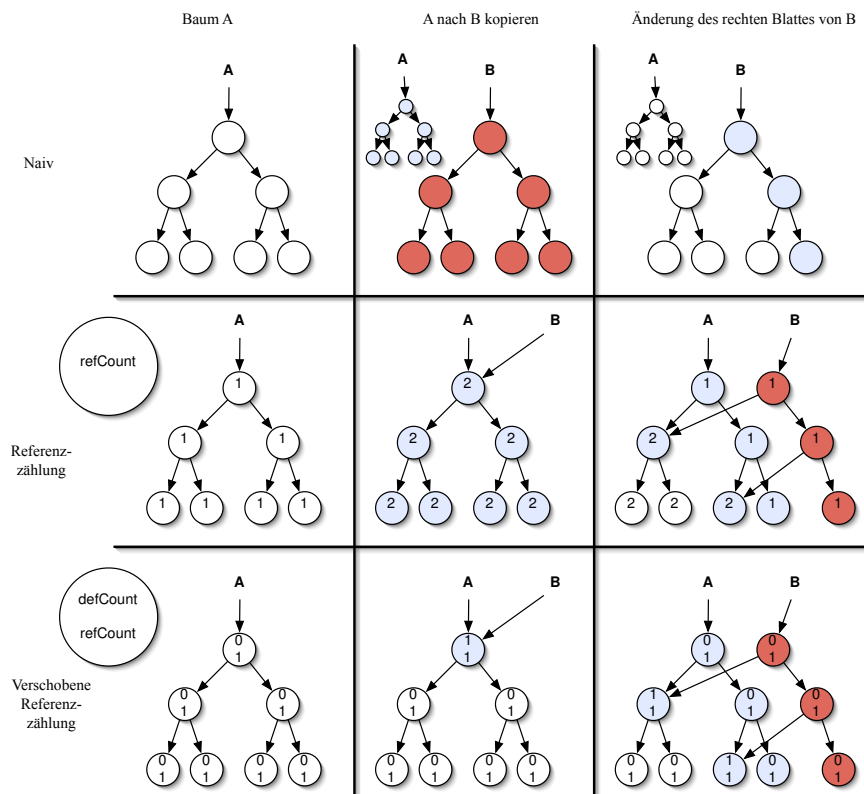
Beim Resampling werden zudem jene Partikel mit den höchsten Gewichten entsprechend ihres Multiplikators vervielfältigt. Allein diese Kopieroperation kann bei großen Karten einen signifikanten Zeitaufwand bedeuten. Zusätzlich besitzen alle kopierten Partikel die gleiche globale Karte und auch im weiteren Verlauf werden größere Teile davon in den jeweiligen Geschwisterpartikeln unverändert bleiben. Es liegt also nahe, diese Bereiche nur einmal im Speicher zu halten. Bereits in [SCHRÖTER, 2008] wurden daher Methoden untersucht, um dies bei regulären Belegtheitskarten zu erreichen.

Während bei diesen jedoch besondere Ansätze vonnöten sind, eignen sich die in dieser Arbeit genutzten  $N^d$ -Bäume schon auf Grund ihrer Struktur vorzüglich für die geteilte Nutzung von Ressourcen, da jeder Knoten des Baumes ohnehin einen gewissen Teilausschnitt der Umgebung repräsentiert. Indem jene Knoten, die in mehreren Partikeln vorhandene und gleiche Regionen repräsentieren, von diesen gemeinsam genutzt werden, kann der Speicherbedarf erheblich eingedämmt werden. Um dies zu gewährleisten ist die Implementierung einer einfachen Referenzzählung durch eine Erweiterung der Knotenobjekte naheliegend, sodass sich jeder Knoten die Anzahl der Referenzen auf sich selbst merkt. Damit kann sodann ein Copy-on-Write Verhalten erzielt werden, indem bei Kopieroperationen lediglich dieser Referenzwert erhöht wird und bei modifizierenden Aktionen nur jene Knoten kopiert werden, deren Wert höher als 1 ist. Zur Reduzierung des Berechnungsaufwandes stellten [FAIRFIELD et al., 2007] eine interessante Erweiterung dieses Konzeptes namens *Deferred Reference Counting Octree*

---



vor. Hierbei werden die Kosten der Propagierung veränderter Referenzwerte in den gesamten Teilbaum adressiert, indem diese aus der Kopieroperation in die Einfügemethode verschoben werden. Dafür wird eine zusätzliche Referenzvariable eingeführt, die beim Kopieren als einziger Wert erhöht wird. Vor modifizierenden Operationen wird dieser sodann nach unten in die eigentlichen Referenzwerte des Teilbaums propagiert.



**Abbildung 5.1:** Vergleich der Kopier- und Aktualisierungsoperationen für unterschiedliche Implementierungen eines Baumes

Blau (bzw. hellgrau) dargestellte Knoten sind jene, auf die zugegriffen werden muss und die rot (bzw. dunkelgrau) gezeichneten sind neue Knoten. Angepasste Version der Veranschaulichung aus [FAIRFIELD, 2009].



---

# Abbildungsverzeichnis

3.1	rekursive Unterteilung der Umgebung durch eine Baumstruktur . . . . .	12
3.2	Vergleich verschiedener Werte für die Parameter $N$ und $d$ . . . . .	13
3.3	Kartierung mit adaptiver Auflösung . . . . .	14
3.4	Beispiel einer Distanztransformation . . . . .	20
4.1	Schematische Darstellung der Komponenten des Bewegungsmodells . . . . .	24
4.2	Typische Beispielsituationen, in denen eine ausschließliche Betrachtung von Belegtheitswerten irreführende Werte liefert . . . . .	28
4.3	Schema zur Distanztransformation . . . . .	29
4.4	Beispiel einer einfachen Umgebungskarte und der errechneten Distanz- transformation . . . . .	30
4.5	Umrechnung von Distanz in Matchwert . . . . .	31
4.6	Ergebnis der Kartenübereinstimmung bei unterschiedlich starkem räum- lichem Versatz . . . . .	32
4.7	Pseudocode für das Resampling der Partikel einschließlich dynamischer Bestimmung der benötigten Anzahl basierend auf [SCHRÖTER, 2008] . . . . .	34
4.8	Umgebungskarte bei ausschließlicher Nutzung der Odometriedaten . . . . .	36
4.9	Ausgabe des SLAM-Algorithmus . . . . .	36

---

4.10 Anzahl der benötigten Partikel und Speicherverbrauch im Verlauf der Umgebungserkundung . . . . .	37
5.1 Vergleich der Kopier- und Aktualisierungsoperationen für unterschied- liche Implementierungen eines Baumes . . . . .	43

---

---

# Literaturverzeichnis

- [ROS, 2011] (2011). *Robot Operating System (ROS)*. <http://www.ros.org>. 1.2
- [BIRK und CARPIN, 2006] BIRK, ANDREAS und S. CARPIN (2006). *Merging occupancy grid maps from multiple robots*. IEEE Proceedings, special issue on Multi-Robot Systems, 94(7):1384–1397. 3.3, 4.3, 4.3.1
- [BORGEFORS, 1986] BORGEFORS, GUNILLA (1986). *Distance transformations in digital images*. Computer Vision, Graphics, and Image Processing, 34(3):344 – 371. 4.3.1
- [DELLAERT et al., 1999] DELLAERT, FRANK, D. FOX, W. BURGARD und S. THRUN (1999). *Monte Carlo Localization for Mobile Robots*. In: *IEEE International Conference on Robotics and Automation (ICRA99)*. 3.2.2
- [EINHORN et al., 2011] EINHORN, ERIK, C. SCHRÖTER und H.-M. GROSS (2011). *Finding the adequate resolution for grid mapping - Cell sizes locally adapting on-the-fly*. In: *ICRA*, S. 1843–1848. IEEE. 1.2, 3.1.2, 3.1.2, 4.1
- [FAIRFIELD, 2009] FAIRFIELD, NATHANIEL (2009). *Localization, Mapping, and Planning in 3D Environments*. Doktorarbeit, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. 5.1
-

- [FAIRFIELD et al., 2007] FAIRFIELD, NATHANIEL, G. KANTOR und D. WETTERGREEN (2007). *Real-time slam with octree evidence grids for exploration in underwater tunnels*. Journal of Field Robotics, 24:3–21. 5.2.4
- [FELZENSZWALB und HUTTENLOCHER, 2004] FELZENSZWALB, PEDRO F. und D. P. HUTTENLOCHER (2004). *Distance transforms of sampled functions*. Technischer Bericht, Cornell Computing and Information Science. 4.3.1
- [GRISSETTI et al., 2007] GRISSETTI, G., C. STACHNISS und W. BURGARD (2007). *Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters*. IEEE Transactions on Robotics, 23(1):34–46. 5.2.3
- [HÄHNEL et al., 2003] HÄHNEL, D, W. BURGARD, D. FOX und S. THRUN (2003). *An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements*. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2003*, Bd. 1, S. 206–211. 3.2, 3.2.2, 4.2.1, 4.1
- [JULIER et al., 2004] JULIER, SIMON J., JEFFREY und K. UHLMANN (2004). *Unscented Filtering and Nonlinear Estimation*. In: *Proceedings of the IEEE*, S. 401–422. 3.2.1
- [KALMAN, 1960] KALMAN, RUDOLPH EMIL (1960). *A New Approach to Linear Filtering and Prediction Problems*. Transactions of the ASME–Journal of Basic Engineering, 82(Series D):35–45. 2, 3.2.1
- [KWAK et al., 2007] KWAK, NOSAN, I. K. KIM, H. C. LEE und B. H. LEE (2007). *Analysis of Resampling Process for the Particle Depletion Problem in FastSLAM*. Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on, S. 200–205. 5.2.3
-

- 
- [MONTEMERLO et al., 2002] MONTEMERLO, M., S. THRUN, D. KOLLER und B. WEGBREIT (2002). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*. In: *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada. AAAI. 2, 3.2.2
- [MORAVEC, 1988] MORAVEC, HANS (1988). *Sensor fusion in certainty grids for mobile robots*. AI Mag., 9:61–74. 3.1.2
- [MORAVEC, 1996] MORAVEC, HANS P. (1996). *Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids*. Technischer Bericht 2
- [MURPHY, 1999] MURPHY, KEVIN (1999). *Bayesian Map Learning in Dynamic Environments*. In: *In Neural Info. Proc. Systems (NIPS)*, S. 1015–1021. MIT Press. 3.2.2
- [OPENCV, 2011] OPENCV (2011). *OpenCV*. <http://opencv.willowgarage.com>. 4.3.1
- [PATHAK et al., 2010] PATHAK, KAUSTUBH, A. BIRK, N. VASKEVICIUS, M. PFINGSTHORN, S. SCHWERTFEGER und J. POPPINGA (2010). *Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation*. J. Field Robot., 27:52–84. 3.1.2
- [SCHRÖTER, 2008] SCHRÖTER, CHRISTOF (2008). *Probabilistische Methoden für die Roboter-Navigation am Beispiel eines autonomen Shopping-Assistenten*. Doktorarbeit, TU Ilmenau. 1.2, 2, 3.1.2, 3.2.2, 4.2, 4.2.1, 4.3.1, 4.3.2, 4.7, 5.2.4, 5.2.4
- [SMITH et al., 1990] SMITH, RANDALL, M. SELF und P. CHEESEMAN (1990). *Estimating uncertain spatial relationships in robotics*, Bd. 4, S. 167–193. 2
-

- [SORENSEN, 1970] SORENSON, H. W. (1970). *Least-squares estimation: from Gauss to Kalman*. IEEE Spectrum, 7:63–68. 3.2.1
- [STACHNISS et al., 2005] STACHNISS, CYRILL, G. GRISETTI und W. BURGARD (2005). *Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM after Actively Closing Loops*. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, S. 667–672. 3.2.2, 5.2.3
- [THRUN et al.] THRUN, SEBASTIAN, W. BURGARD und D. FOX. *Probabilistic Robotics*. Intelligent robotics and autonomous agents. The MIT Press. 3.2
- [WURM et al., 2010] WURM, K. M., A. HORNING, M. BENNEWITZ, C. STACHNISS und W. BURGARD (2010). *OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems*. In: *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA. Software available at <http://octomap.sf.net/>. 2
-



# Thesen

- Zweidimensionale Umgebungsrepräsentationen sind für moderne Anwendungen auf Grund des beschränkten Informationsgehalts nicht mehr ausreichend.
- Existierende Verfahren zur dreidimensionalen Modellierung der Umgebung sind oft sehr auf den verwendeten Sensor und den Einsatzort abgestimmt und können nicht ohne Weiteres auf andere Anwendungen übertragen werden.
- Der Map-Match-SLAM-Algorithmus wurde für den Einsatz mit verschiedenster Sensorik konzipiert und kann durch die Verwendung von Gridkarten beliebige Umgebungen ohne a-priori Wissen kartieren.
- $N^d$ -Bäume sind ideal dafür geeignet die an sich sehr speicherlastigen Belegtheitskarten sowohl generisch und speichereffizient zu repräsentieren als auch die Kosten für deren Erstellung drastisch zu reduzieren.
- Eine Kombination dieser beiden Verfahren ist dafür geeignet SLAM unabhängig von Sensorik und gewünschter Dimensionalität zu implementieren.
- Das Resampling wird durch die Nutzung eines Kartenübereinstimmungsmaßes auf Basis einer Distanztransformation wesentlich robuster.

Ilmenau, 11.01.2012

.....

Andreas Hollandt