

ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
FAKULTÄT FÜR ANGEWANDTE WISSENSCHAFTEN

Institut für Informatik
Arbeitsgruppe Autonome Intelligente Systeme
Prof. Dr. Wolfram Burgard

Untersuchungen zur Genauigkeit von SLAM-Verfahren mit Partikel-Filtern

Diplomarbeit

Verfasser: Slawomir Grzonka
Abgabedatum: 27. Juni 2006
Erstgutachter: Prof. Dr. Wolfram Burgard
Zweitgutachter: Prof. Dr. Bernhard Nebel
Betreuer: Dipl.-Inf. Christian Plagemann
Kobetreuer: Dr. Giorgio Grisetti

Auch der längste Weg beginnt mit dem ersten Schritt.

Chinesische Weisheit

Kurzfassung

Die Fähigkeit, eine Karte seiner Arbeitsumgebung eigenständig zu erstellen, ist eine der Grundvoraussetzungen für ein wirklich autonom agierendes System. Im Kontext der Kartographierung und zeitgleichen Lokalisierung (SLAM) mit autonomen Robotern hat sich in den letzten Jahren der *Rao-Blackwellized Particle Filter* als ein effizientes und genaues Verfahren erwiesen. Für unendlich viele Kartenhypothesen (Partikel) lässt sich die Konvergenz des Verfahrens beweisen. Bei der in der Praxis verwendeten geringen Anzahl von Partikeln kann die Unsicherheit und das Rauschen durch die verwendeten Modelle (Zustandsübergangsfunktion, Observationsfunktion) nur zu einem gewissen Maß beschrieben werden. Dadurch kann die Integration von Messungen an falschen bzw. ungünstigen Positionen nicht verhindert werden. Dies kann zu einer unpräzisen Karte und im Extremfall zu deren Divergenz führen. Deshalb werden in dieser Arbeit insgesamt drei Verfahren entwickelt, die die korrekte Kartengenerierung auch in schwierigen Umgebungen gewährleisten bzw. begünstigen sollen. Der SLAM-Kontext wird dazu in drei zeitlichen Abschnitten auf mögliche Verbesserungen des *Rao-Blackwellized Particle Filters* hin untersucht. Der Zeitpunkt vor, während und nach der Integration der aktuellen Messung. Für jeden dieser Zeitpunkte wird ein Verfahren entwickelt, das die „Verschmierung“ einer Karte verhindern bzw. minimieren soll. In Experimenten mit echten Robotern wird die Wirkung der Verfahren sowohl im Innen- als auch im Außenbereich demonstriert. Dabei wird gezeigt, dass die entwickelten Verfahren zu einer präziseren Karte führen und im Innenbereich nahe an die Leistungsfähigkeit des *scanmatchingbasierten FastSLAM* herankommen. Im Außenbereich werden auch im Vergleich zum *scanmatchingbasierten FastSLAM* deutlich bessere Ergebnisse erzielt.

Abstract

The ability to construct an accurate map of the environment is one of the vital properties of a truly autonomous agent. In the context of simultaneous localization and mapping (SLAM), the *Rao-Blackwellized Particle Filter* (RBPF) has emerged as an efficient approach to this problem. The convergence of this technique can be guaranteed for an infinite number of map hypotheses. In practice, however, only a limited number of hypotheses can be maintained, which leads to an upper limit on the level of noise the system can deal with. Thus, the integration of laser scans at false or unfavourable map positions cannot be avoided completely, which may lead to inconsistent or even diverged maps. In this work, three extensions to the standard RBPF algorithm are introduced that yield accurate maps even in low-structured environments and with higher noise levels. The three extensions work complementarily at the three main parts of the mapping algorithm respectively: the point of time before, during, and after the integration of the current observation. Experiments with real robots in highly structured indoor environments as well as in low-structured outdoor environments demonstrate that the developed techniques lead to substantially improved maps. It is furthermore shown that the proposed system is nearly as accurate as the *scanmatching-based FastSLAM* algorithm in the highly structured environment and clearly outperforms it in the low-structured case.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Außerdem erkläre ich, dass die Diplomarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung verwendet wurde.

(Slawomir Grzonka)
Freiburg, 27. Juni 2006

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Erstellung dieser Diplomarbeit unterstützt haben. Ich danke Herrn Prof. Dr. Wolfram Burgard für die Heranführung an die wissenschaftliche Arbeitsweise und dafür, dass er jederzeit ein offenes Ohr für Fragen hatte. Weiterhin danke ich Herrn Prof. Dr. Bernhard Nebel dafür, dass er sich als Zweitgutachter zur Verfügung gestellt hat. In besonderem Maße möchte ich Christian Plagemann für die großartige Betreuung, Unterstützung und Hilfe danken, die ich von ihm während meiner Studien- und Diplomarbeit erfahren habe. Weiterhin möchte ich Giorgio Grisetti für seine sehr gute Betreuung, hilfreiche und anregenden Gespräche danken. Auch bei Axel Rottmann und Wolfgang Kluge möchte ich mich bedanken.

Ich danke Martin Wehrle, Bernd Gutmann, Robert Mattmüller und Thomas Wimmer. Ich bedanke mich bei meinen Eltern, die es mir ermöglicht haben, dieses Studium einzuschlagen und die mich während der gesamten Zeit unterstützt haben. Schließlich bedanke ich mich bei Martin Grzonka und Beate Garbers für ihr Verständnis und ihre Unterstützung, gerade in der heißen Endphase.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zielsetzung	4
1.2	Aufbau dieser Arbeit	4
2	Verwandte Arbeiten	5
3	Sequentielle Zustandsschätzung	9
3.1	Notation	9
3.2	Bayes-Filter	10
3.3	Kalman-Filter	12
3.4	Informations-Filter	17
3.5	Partikel-Filter	19
4	Grundlagen der Lokalisierung und Kartengenerierung	23
4.1	Kartenrepräsentation	23
4.2	Lokalisierung mit dem Partikel-Filter bei bekannter Karte	25
4.3	Kartengenerierung bei bekannter Trajektorie	29
4.4	Rao-Blackwellized-Partikel-Filter und das SLAM-Problem	31
5	Entwickelte Erweiterungen zum RBPF	33
5.1	Verzögerte Integration von Messungen zur verbesserten Positionsschätzung	35
5.2	Entropie-Differenz als Gütemaß zur Umschaltung von SLAM auf reine Lokalisierung	42
5.3	Erlernen einer korrelationsbasierten Netzstruktur zur nachträglichen Korrektur von Messintegrationsfehlern	44
5.4	Implementierung des Gesamtsystems	48
6	Evaluierung	51
6.1	Experiment innerhalb eines Gebäudes	52
6.2	Experiment im Außenbereich	58
6.3	Evaluation der einzelnen Verfahren	65
6.3.1	Verzögerte Integration	65

6.3.2	Adaptives Umschalten von SLAM auf reine Lokalisierung	66
6.3.3	Erlernen einer korrelationsbasierten Netzstruktur	66
7	Zusammenfassung und Ausblick	69
A	Technische Daten	71
A.1	ActivMedia Robotics Pioneer 3-DX	71
A.2	SICK-LMS-Laserscanner	72
	Abbildungsverzeichnis	73
	Literaturverzeichnis	75

1 Einleitung

Die mobile Robotik gewinnt, getrieben durch entstehende Anwendungen und wissenschaftlichen Fortschritt, zunehmend an Bedeutung im alltäglichen Leben. Schon in einigen Jahren wird es für uns selbstverständlich sein, in vielen Bereichen durch autonome Helfer unterstützt zu werden. Nach Aussagen der *United Nations Economic Commission for Europe* [UNECE 2005] werden bereits jetzt mehr als eine Million Roboter im häuslichen Gebrauch verwendet; Tendenz stark steigend. Die Einsatzgebiete solcher Helfer reichen vom autonomen Rasenmäher bis zum Transportieren von Gegenständen in Kliniken und Fabriken. Unabhängig vom Einsatzgebiet ist die genaue Kenntnis der eigenen Position und der Umgebung für eine verlässliche und effiziente Arbeitsweise von entscheidender Bedeutung. Dazu ist es nötig, dass der Roboter eine akkurate Karte seiner Arbeitsumgebung besitzt. Fehlerhafte Karten und der dadurch verursachte Verlust der eigenen Positionsinformation können zu schwerwiegenden Folgen führen. Während sich womöglich der Nachbar freut, dass der autonome Rasenmäher auch einen Teil seines Gartens mäht, können die Folgen für einen Patienten fatal sein, falls sich der Transportroboter im Zimmer geirrt hat und ihm somit falsche Medikamente liefert.

Sind vor Inbetriebnahme des Roboters bereits Konstruktionspläne des Arbeitsumfeldes vorhanden, können diese in eine Karte umgewandelt und dem Roboter vor Beginn seiner Tätigkeit mitgegeben werden. Dies hat jedoch mehrere Nachteile:

- Das Inventar des Gebäudes, wie z. B. Tische, ist in den Konstruktionsplänen meist nicht vorhanden.
- Kleine Umbauten werden in den Plänen nicht erfasst.
- Der Roboter ist nur für eine Arbeitsumgebung konzipiert. Bei einem Wechsel (z. B. Umzug) muss die Karte des Roboters aktualisiert werden. Dies ist oftmals mit hohen Kosten verbunden.
- Jeder Roboter muss individuell an seinen Kunden angepasst werden, was sich in hohen Anschaffungskosten widerspiegelt.
- Die Konstruktionspläne weichen oftmals vom echten Gebäude ab.
- Für ältere Gebäude sind Konstruktionspläne teils nicht vorhanden.

Besser als die manuelle Konstruktion von Karten durch den Menschen ist, dass der Roboter die Karte seiner Umgebung selbstständig bei Aufnahme seiner Arbeit mit Hilfe seiner Sensoren

erstellt. Dieses Verfahren, bekannt unter dem Namen SLAM (engl. für *simultaneous localization and mapping*, also „zeitgleiche Lokalisierung und Kartengenerierung“) hat in den letzten Jahren eine immer größere Bedeutung erlangt. Dabei soll der Roboter eine Karte seiner Arbeitsumgebung erstellen und sich zeitgleich in dieser lokalisieren. Diese Eigenschaft ist eine der Grundvoraussetzungen für ein wirklich autonom agierendes System. In dieser Arbeit wird das SLAM-Verfahren mit dem *Rao-Blackwellized Particle Filter* (kurz: RBPF) basierend auf Rasterkarten verwendet (siehe Kapitel 4.4). Das Verfahren kann folgendermaßen skizziert werden: Aufgrund der eigenen Positionsunsicherheit baut der Roboter zeitgleich mehrere Karten der Umgebung auf und versucht, sich innerhalb jeder Karte bezüglich dieser zu lokalisieren. Eine Karte besteht dabei aus Zellen gleicher Größe (z. B. $5\text{cm} \times 5\text{cm}$). Die Karten weichen oftmals nur geringfügig voneinander ab. Die Unterschiede werden besonders dann deutlich, wenn beispielsweise eine Schleife geschlossen wird. Einige Karten können dann z. B. überlappende Wände besitzen, während in anderen Karten die zusammengesetzten Korridore passen (siehe Abbildung 1.2). Die grundlegenden Probleme, welche bei der Konstruktion einer Karte nach diesem Verfahren auftreten, sind die Folgenden. Theoretisch sind unendlich viele Karten nötig, um die Konvergenz des sequentiellen Verfahrens bei den existierenden Unsicherheiten zu garantieren. In der Praxis wird nur eine geringe Anzahl an Karten (< 1000) generiert. Deshalb müssen die verwendeten Modelle (Zustandsübergangsfunktion, Observationsfunktion) möglichst genau sein. Dennoch weicht im Allgemeinen die geschätzte eigene Position von der wahren, wenn meist auch nur gering, ab. Somit werden Messungen an falschen Positionen integriert. Da die Sensoren eines Roboters nicht perfekt sind, können fehlerhafte Messungen auftreten. Durch die Diskretisierung der Karte können Objekte evtl. nur zum Teil in einer Zelle liegen. All dies führt dazu, dass sich innerhalb der Karten Unsicherheiten bezüglich einzelner Zellen bilden. Im fortlaufenden Betrieb führt dies zu einer „Verschmierung“ der Karte. Diese Kartenunsicherheit führt zu einer erhöhten Unsicherheit bezüglich der eigenen Position, z. B. durch unpräzise Merkmale, an denen sich der Roboter gut hätte lokalisieren können, einer möglichen Divergenz der Karte und damit zu den am Anfang beschriebenen Problemen (siehe Abbildung 1.1). Darüber hinaus ist es nicht immer notwendig, alle Messungen zu integrieren. Hat der Roboter bereits einen Bereich kartographiert und würde die aktuelle Messung dem Roboter keine neuen Informationen über den Bereich liefern, so ist es sinnvoller, die Messung zu verwerfen, als diese zu integrieren und dadurch eine eventuelle Vergrößerung der Kartenunsicherheit zu riskieren.

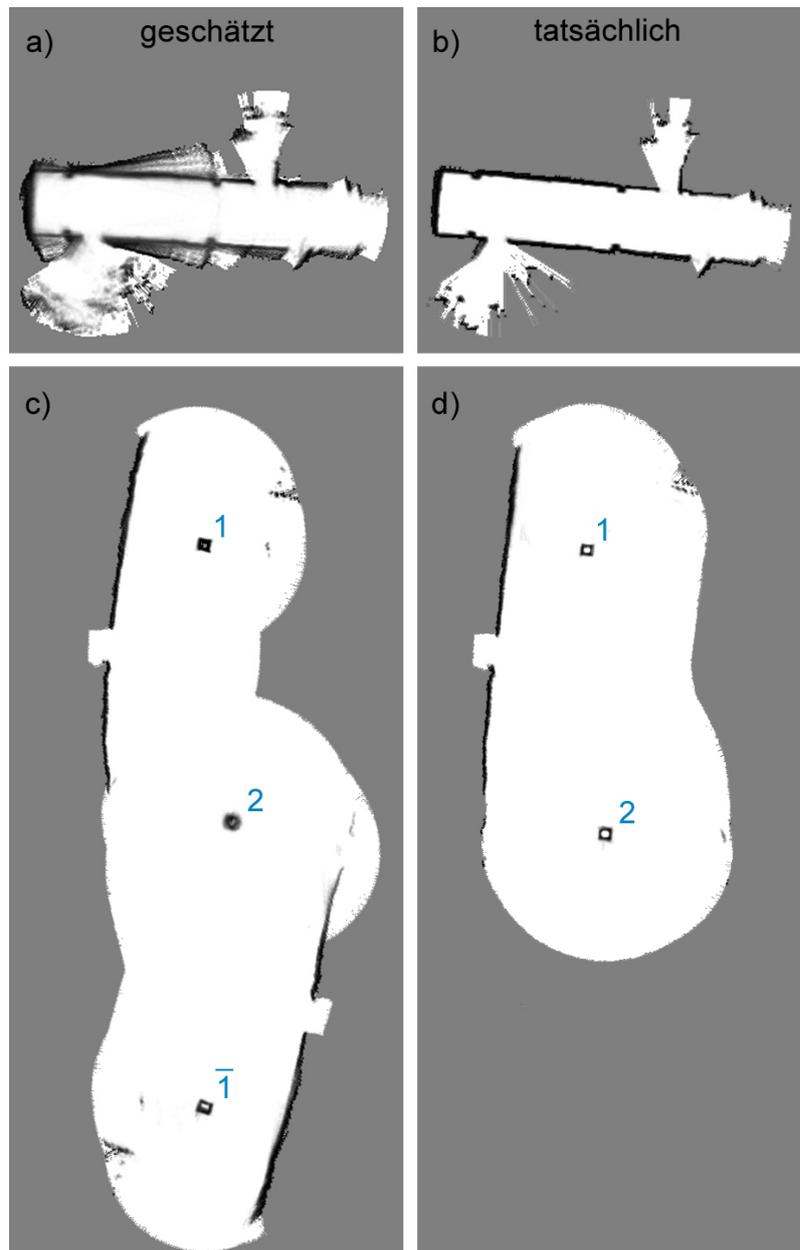


ABBILDUNG 1.1: Probleme der rasterkartenbasierten Kartographierung im Innen- und Außenbereich. Karten aus Experimenten mit einem echten Roboter. Zustand der besten Kartenhypothese nach b) einer und a) mehreren Runden innerhalb eines Gebäudes. c) und d) Experiment mit zwei Kisten im Außenbereich. Nachdem der Roboter von Kiste 1 zu Kiste 2 gefahren ist, hat er Kiste 2 mehrmals umrundet, um anschließend wieder zu Kiste 1 zu fahren. Bei der Kartographierung mit dem Standardverfahren ist Kiste 2 nach mehreren Umrundungen immer mehr zu einem Kreis geworden. Die Lokalisierung an ihr wurde dadurch erschwert. Dies führte schließlich dazu, dass der Roboter innerhalb seiner Karte nicht, wie real (d), zu Kiste 1 zurückgefahren ist und deshalb beim erneuten Beobachten derselben sie als neue Kiste $\bar{1}$ kartographiert hat. Der Fehler, der durch die erschwerte Lokalisierung an Kiste 2 erzeugt wurde, führte schließlich zu einer Divergenz der Karte.

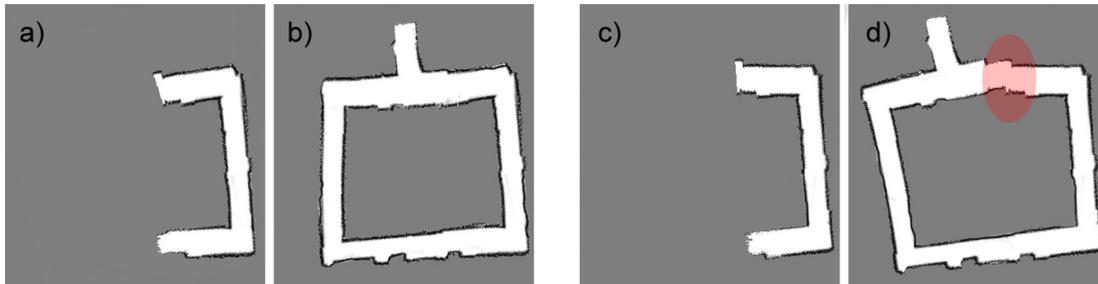


ABBILDUNG 1.2: Beispiel für den *Rao-Blackwellized Particle Filter* in Kombination mit Rasterkarten. Die Kartenhypothesen a) und c) weichen zu Beginn nur gering voneinander ab. Der Unterschied wird jedoch deutlich, wenn Schleifen geschlossen werden. Die Hypothese a) ist in der Lage, die Schleife korrekt zu schließen (b), während Hypothese c) inkonsistent bezüglich der Wände (rot unterlegt) ist (d).

1.1 Zielsetzung

Ziel dieser Arbeit ist die Verbesserung des SLAM-Verfahrens nach [MURPHY 1999], um auch in schwierigen Umgebungen die oben beschriebene Kartenunsicherheit zu reduzieren. Weiterhin soll ein Verfahren entwickelt werden, welches die Güte von Messungen bewerten und somit gegebenenfalls deren Integration verhindern kann. Dies ist gleichbedeutend mit einem Verfahren, welches adaptiv zwischen Kartographierung und Lokalisierung umschalten kann.

1.2 Aufbau dieser Arbeit

Diese Arbeit ist wie folgt gegliedert: In Kapitel 2 werden bisherige Arbeiten vorgestellt, die ähnliche Ziele mit anderen Ansätzen verfolgen. Anschließend werden in Kapitel 3 die Grundlagen der sequentiellen Zustandsschätzung erläutert. Dazu werden basierend auf dem Bayes-Filter der in dieser Arbeit verwendete Partikel-Filter und aus Gründen der Vollständigkeit der Kalman-Filter, der Informations-Filter und ihre nichtlinearen Erweiterungen beschrieben. In Kapitel 4 wird auf verschiedene Kartenrepräsentationen eingegangen. Dabei bildet die Rasterkarte (engl. *Occupancy Grid Map*) die Grundlage für die in dieser Arbeit verwendeten Verfahren. Aufbauend auf der Rasterkarte werden dazu Verfahren zur Lokalisierung und Kartengenerierung erläutert und der *Rao-Blackwellized Particle Filter*, das für diese Arbeit zugrundeliegende Verfahren, vorgestellt. In Kapitel 5 werden die in dieser Arbeit entwickelten Erweiterungen zum *Rao-Blackwellized Particle Filter* beschrieben. In Kapitel 6 wird auf die durchgeführten Experimente, ihre Ergebnisse und Deutungen eingegangen. Abschließend werden in Kapitel 7 die erzielten Ergebnisse zusammengefasst und mögliche Erweiterungen und Verbesserungen des Systems diskutiert.

2 Verwandte Arbeiten

Die existierenden Verfahren zur simultanen Lokalisierung und Kartengenerierung (SLAM) lassen sich grob nach zwei Kriterien gliedern:

- die Art ihrer Kartenrepräsentation und
- das Verfahren zur Positionsschätzung.

Bei der Kartenrepräsentation unterscheidet man im Wesentlichen zwischen vier Kategorien.

Topologische Karten repräsentieren ihre Umgebung durch einen Graphen. Dabei stehen Knoten für Raumbereiche und Kanten für eine Verbindung zwischen diesen. Topologische Karten beschreiben die grobe Struktur der Umgebung und beinhalten im Allgemeinen weder metrische noch detaillierte Informationen über den konkreten Aufbau eines Bereiches. Topologische Karten werden beispielsweise in [NEHMZOW et al. 1993, HAHNEL et al. 2003] verwendet.

Landmarkenbasierte Karten repräsentieren ihre Umgebung durch eine Menge von Orientierungspunkten. Sie enthalten metrische Informationen über die Orientierungspunkte und erlauben genauere Navigation und Kontrolle als bei topologischen Karten [THRUN et al. 2005].

Rasterkarten (engl. *Occupancy Grid Maps*) diskretisieren ihre Umgebung durch eine Menge von gleichgroßen Zellen (z. B. $5\text{cm} \times 5\text{cm}$). Eine Zelle beinhaltet dabei die Wahrscheinlichkeit, dass diese belegt und somit für den Roboter unpassierbar ist. Sie liefern eine detaillierte Repräsentation ihrer Umgebung auf Kosten eines erhöhten Speicherplatzbedarfs [STACHNISS et al. 2004b]. **Hybride Ansätze** verwenden mehrere der oben genannten Repräsentationen zugleich. Oft handelt es sich dabei um eine Kombination aus topologischen und diskretisierten Karten [GRISSETTI et al. 2005]. Eine topologische Karte kann beispielsweise semantische Informationen über die Umgebung enthalten, was eine bessere Lokalisierung und Navigation auf höherem Abstraktionsniveau erlaubt [ROTTMANN et al. 2005].

Verfahren zur Positionsschätzung können im Groben in drei Kategorien eingeteilt werden.

Parametrisierte Verfahren zur Positionsschätzung repräsentieren die Position durch eine feste Form der Wahrscheinlichkeitsverteilung und deren Parametern, z. B. den Mittelwert μ und die Kovarianz Σ einer mehrdimensionalen Normalverteilung. Das älteste und bekannteste Verfahren dieser Kategorie ist der Kalman-Filter (kurz: KF), der erstmals von Swerling und Kalman beschrieben wurde [KALMAN 1960]. Kalman-Filter, ebenso die nichtlineare Erweiterung des Kalman-Filters (kurz: EKF) werden oftmals in Verbindung mit landmarkenbasierten Karten verwendet.

Dabei besteht der Zustandsvektor aus dem Mittelwert der Positionen des Roboters, der Merkmale und ihren Unsicherheiten. Die Kovarianzmatrix besteht entsprechend aus deren Korrelationen. Wie der Kalman-Filter repräsentieren der Informations-Filter (kurz: IF) und seine nichtlineare Erweiterung (kurz: EIF) den Zustand durch eine mehrdimensionale Normalverteilung. Der Unterschied zwischen den Verfahren besteht in der Darstellung des Zustandsvektors. Im Kalman-Filter wird die Normalverteilung durch den Mittelwert und die Kovarianzmatrix repräsentiert, während der Informations-Filter die Normalverteilung durch die Informationsmatrix $\Omega = \Sigma^{-1}$ und den Informationsvektor $\xi = \Sigma^{-1}\mu$ darstellt. Ein Vorteil des EIF gegenüber dem EKF im SLAM-Kontext ist die Art der Auswirkung einer Messintegration auf die Kovarianzmatrix Σ bzw. die Informationsmatrix Ω . Während sich die Kovarianzmatrix in fast allen Einträgen signifikant ändert, besitzt die Informationsmatrix nur wenige Einträge, die verändert werden. [THRUN et al. 2004a] erweitern den EIF zum *sparse extended information filter* durch Approximation der Informationsmatrix. Dies wird durch die Festlegung einer maximalen Anzahl n von möglichen Änderungen erreicht. Dadurch kann eine Iteration in amortisiert konstanten Kosten erreicht werden, wohingegen der EKF quadratische Laufzeit besitzt.

Stichprobenbasierte Verfahren repräsentieren die Verteilung des Zustands durch eine gewichtete Menge an Stichproben. Diese Verfahren, in der Literatur als Monte-Carlo-Verfahren beziehungsweise Partikel-Filter bekannt, besitzen gegenüber parametrisierten Verfahren den Vorteil, dass sie beliebige und insbesondere multimodale Verteilungen approximieren können. Dies wird in der Praxis allerdings dadurch eingeschränkt, dass nur eine endliche Anzahl von Stichproben (Partikeln) zur Verfügung steht. Ein Verfahren zur adaptiven Wahl der Partikel-Anzahl wird von [FOX 2003] vorgestellt. [MURPHY 1999] erweitert den Partikel-Filter zum *Rao-Blackwellized Particle Filter*, bei dem jedes Partikel eine Trajektorie des Roboters inklusive zugehöriger Rasterkarte repräsentiert. Dieses Verfahren bildet die Grundlage für diese Arbeit.

Maximum-Likelihood-Verfahren schätzen, basierend auf der gesamten Trajektorie und den Messungen die daraus resultierende wahrscheinlichste Karte. Diese Verfahren finden ihre Verwendung insbesondere bei der Kombination verschiedener Sensoren, bzw. wenn Sensoren sich in ihren Messbereichen überlappen. Dies ist beispielsweise bei Ultraschallsensoren der Fall. [THRUN 2002] verwendet den EM-Algorithmus um eine Karte zu erstellen. Der Nachteil dieses Verfahrens ist, dass dazu die gesamte Trajektorie und alle Messungen bekannt sein müssen, was einen sequentiellen Einsatz zur Laufzeit ausschließt. Durch Weglassen des *Expectation*-Schrittes, in dem die Trajektorie basierend auf der aktuell besten Karte geschätzt wird, wird durch Thrun et al. das Verfahren dahingehend erweitert, dass es zur Laufzeit angewendet werden kann. Dazu wird eine Menge von möglichen Trajektorien und die aktuell beste Karte geschätzt. Dies kann als Vorläufer des *Rao-Blackwellized Particle Filters* angesehen werden, dessen Erweiterung zu diesem Verfahren die Verwendung einer eigenen Karte für jede Trajektorie repräsentiert durch einen Partikel-Filter darstellt.

Verfahren zur Reduzierung von Kartenunsicherheiten und zur Umschaltung von SLAM auf reine Lokalisierung können in drei zeitlichen Abschnitten betrachtet werden:

Vor der Integration der aktuellen Messung führt die Berechnung einer akkuraten Position zu

einer Reduzierung von Unsicherheiten. Dies wird z. B. bei [STACHNISS 2004] durch den *Scanmatcher* erreicht. Dabei bildet die aktuelle Messung basierend auf einem Laserscanner, im Gegensatz zur Odometrie beim Standardverfahren, die Grundlage für die Schätzung einer Zustandsübergangsfunktion. Dieses Verfahren, bekannt als *scanmatchingbasiertes FastSLAM* (kurz: SM-FastSLAM), hat sich in vielen Anwendungen bewährt und benötigt im Vergleich zum Standardverfahren um ca. eine Größenordnung geringere Anzahl an Partikeln.

Zum Zeitpunkt der Integration kann die Verhinderung der Integration der aktuellen Messung und der damit einhergehenden Umschaltung von SLAM auf reine Lokalisierung zu einer Verringerung von Messintegrationsfehlern und somit zu einer geringeren Kartenunsicherheit führen. [GRISSETTI et al. 2005] verwenden dazu eine hybride Kartenstruktur bestehend aus einer topologischen Karte und, für jeden Knoten darin, einer Rasterkarte. Dabei repräsentieren die Knoten einzelne Bereiche, die durch eine kleine Rasterkarte (engl. patch) detailliert dargestellt werden. Der Roboter bewegt sich hierbei innerhalb der Rasterkarten von Knoten zu Knoten. Eine Messung wird nur dann integriert, wenn ein neuer Knoten erstellt und somit unbekanntes Gebiet kartographiert wird. Im Kontext des EKF-SLAM verwenden [ZHANG et al. 2005] den Informationsgewinn (basierend auf der Entropie), der durch das Hinzunehmen der in der aktuellen Messung erfassten neuen Landmarke zustandekommt und entscheiden basierend darauf, ob diese in den aktuellen Zustandsvektor aufgenommen wird oder nicht. Dies führt zu einer geringeren Anzahl an verwendeten Orientierungspunkten und somit zu einer verbesserten Laufzeit gegenüber dem Standard EKF-SLAM.

Nach der Integration werden im wesentlichen *Maximum-Likelihood*-Verfahren verwendet, um Messintegrationen nachträglich zu korrigieren. Dies kann beispielsweise mit dem *Expectation-Maximization*-Algorithmus erreicht werden [THRUN 2002]. [TEN HAGEN und KROSE 2002] verwenden den Kalman-Filter, um die bestehende Trajektorie nachträglich zu glätten und dadurch eine genauere Karte zu erhalten.

Der Beitrag dieser Arbeit ist die Entwicklung von drei Verfahren. Dazu wird, aufbauend auf dem *Rao-Blackwellized Particle Filter*, für jeden der oben genannten Zeitpunkte ein Verfahren entwickelt, das die über die Zeit eintretende Kartenunsicherheit minimieren soll. Das Verfahren, das vor der Integration der aktuellen Messung eingesetzt wird, dient im Vergleich zur Odometrie der akkurateren Schätzung der Zustandsübergangsfunktion. Dieses Verfahren kann als stichprobenbasierte Alternative zum SM-FastSLAM angesehen werden. Dadurch können Multimodalitäten aufgelöst werden, während der auf Gradientenabstieg und der aktuellen Messung basierende SM-FastSLAM zu einer suboptimalen Lösung führen kann. Das Verfahren, welches zum Zeitpunkt der Integration zum Einsatz kommt dient der Umschaltung von SLAM auf reine Lokalisierung. Dies wird durch eine Berechnung des Informationsgewinns der aktuellen Messung, basierend auf der Entropie, erreicht. Abschließend wird fortlaufend eine korrelationsbasierte Netzstruktur der Karte erlernt, um Messintegrationsfehler mit zukünftigen Messungen zu korrigieren. Der Vorteil dieses Verfahrens – beispielsweise gegenüber dem EM-Ansatz – ist, dass es zur Laufzeit verwendet werden kann.

2 Verwandte Arbeiten

3 Sequentielle Zustandsschätzung

Bevor Algorithmen zur Lokalisierung und Kartengenerierung besprochen werden, sollen in diesem Kapitel die grundlegenden Verfahren zur sequentiellen Zustandsschätzung im allgemeinen Fall erläutert werden. Das theoretische Grundgerüst zur rekursiven Schätzung der Wahrscheinlichkeitsverteilung eines Zustandsvektors \mathbf{x} (im Folgenden *Belief*, oder kurz $bel(\mathbf{x})$ genannt) bildet der Bayes-Filter. Dieser wird in Kapitel 3.2 eingeführt. Anschließend werden fünf Implementierungen des Bayes-Filters vorgestellt und deren Stärken und Schwächen abhängig vom Einsatzgebiet diskutiert: der Kalman-Filter und der Informations-Filter, zwei parametrisierte Implementierungen des Bayes-Filters, ihre nichtlinearen Approximationen und der Partikel-Filter, eine nichtparametrisierte Approximation des Bayes-Filters. Die formalen Grundlagen sind weitestgehend an [THRUN et al. 2005] und [RISTIC et al. 2004] angelehnt. Die Notation deckt sich mit der in [GRZONKA 2005] verwendeten.

3.1 Notation

Für den formalen Umgang mit den in diesem Kapitel beschriebenen Verfahren wird die folgende Notation verwendet:

$p(A)$	Wahrscheinlichkeit für Ereignis A
$p(A, B)$	Wahrscheinlichkeit für Ereignis A und B
$p(A B)$	Bedingte Wahrscheinlichkeit von Ereignis A , gegeben Ereignis B
\mathbf{x}_t	Zustand des Systems zum Zeitpunkt t
\mathbf{z}_t	Sensormessung zum Zeitpunkt t
\mathbf{u}_t	Kontrollkommando im Zeitintervall $(t - 1, t]$
$\mathbf{x}_{1:t}$	$:= \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t$
$\mathbf{z}_{1:t}$	$:= \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t$
$\mathbf{u}_{1:t}$	$:= \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_t$
$\overline{bel(\mathbf{x}_t)}$	$= p(\mathbf{x}_t \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$, a-priori-Verteilung des Zustands zum Zeitpunkt t
$bel(\mathbf{x}_t)$	$= p(\mathbf{x}_t \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$, a-posteriori-Verteilung des Zustands zum Zeitpunkt t
μ_t	Mittelwert zum Zeitpunkt t
Σ_t	Kovarianz zum Zeitpunkt t
η	Normalisierungskonstante

3.2 Bayes-Filter

Das theoretische Grundgerüst zur sequentiellen Zustandsschätzung ist durch den Bayes-Filter gegeben. Dieser rekursive Filter berechnet die Wahrscheinlichkeitsverteilung über den Zustand $bel(\mathbf{x}_t)$ basierend auf den Messdaten $\mathbf{z}_{1:t}$ und den Kommandos $\mathbf{u}_{1:t}$. Dazu wird angenommen, dass der Zustand \mathbf{x}_t vollständig (Markow) ist. Vollständigkeit bedeutet hierbei, dass in \mathbf{x}_t alle Informationen bis zum Zeitpunkt t vorhanden sind. Daraus folgt, dass \mathbf{x}_t nur von \mathbf{x}_{t-1} und \mathbf{u}_t abhängig ist. Vorherige Zustände $\mathbf{x}_{0:t-2}$, Kommandos $\mathbf{u}_{1:t-1}$ und Messungen $\mathbf{z}_{1:t}$ liefern keine zusätzlichen Informationen. Weiterhin ist \mathbf{z}_t nur von \mathbf{x}_t abhängig. Diese Annahme wird durch Abbildung 3.1 visualisiert. Per Induktion kann man zeigen, dass sich $bel(\mathbf{x}_t)$ aus $bel(\mathbf{x}_{t-1})$ berechnen lässt. Mit Induktionsanfang $bel(\mathbf{x}_0)$ und der Markow-Annahme gilt

$$\begin{aligned}
 p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) & \stackrel{\text{Bayes}}{=} \frac{p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})} \\
 & = \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 & \stackrel{\text{Markow}}{=} \eta p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}). \tag{3.1}
 \end{aligned}$$

Nun gilt mit $bel(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ und $\overline{bel}(\mathbf{x}_t) = p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t})$

$$bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \overline{bel}(\mathbf{x}_t). \tag{3.2}$$

Mit dem Satz von der totalen Wahrscheinlichkeit¹ lässt sich $\overline{bel}(\mathbf{x}_t)$ umformen zu

$$\begin{aligned}
 \overline{bel}(\mathbf{x}_t) & = p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) \\
 & \stackrel{\text{S.v.d.t.W.}}{=} \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t}) d\mathbf{x}_{t-1} \\
 & \stackrel{\text{Markow}}{=} \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}) d\mathbf{x}_{t-1} \\
 & = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}. \tag{3.3}
 \end{aligned}$$

Einsetzen von (3.3) in (3.2) führt zu:

$$bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}. \tag{3.4}$$

In (3.4) wird die rekursive Natur des Bayes-Filters deutlich, denn $bel(\mathbf{x}_t)$ kann aus $bel(\mathbf{x}_{t-1})$, dem Kontrollkommando \mathbf{u}_t und der Messung \mathbf{z}_t berechnet werden. Der Korrektheitsbeweis des Bayes-Filters basiert auf der Annahme, dass \mathbf{x}_t vollständig (Markow) ist. In der Praxis ist Vollständigkeit meist nicht gegeben, da z. B. Sensoren nur einen Teil der aktuellen Welt erfassen

¹Für $A, B \subseteq \Omega$, mit Grundraum Ω gilt: $p(A) = \int p(A, B) dB = \int p(A|B)p(B) dB$

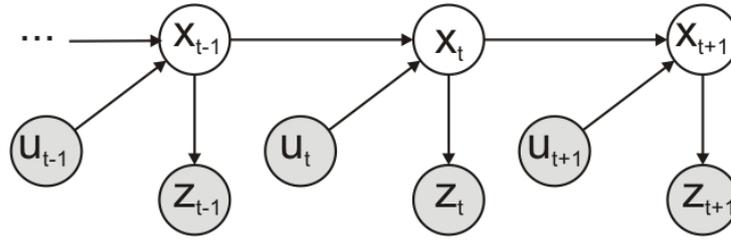


ABBILDUNG 3.1: Abhängigkeiten zwischen den Zuständen, Messungen und Kommandos unter der Markow-Annahme. Die Werte der grau unterlegten Variablen sind bekannt. Abhängigkeiten werden durch Pfeile dargestellt.

```

1:   Algorithmus Bayesfilter( $bel(\mathbf{x}_0), \mathbf{z}_{1:k}, \mathbf{u}_{1:k}$ ) :
2:     for  $i = 0, \dots, k - 1$  do
3:        $bel(\mathbf{x}_{i+1}) = \text{Bayesfilter\_update}(bel(\mathbf{x}_i), \mathbf{z}_{i+1}, \mathbf{u}_{i+1})$ 
4:     endfor
5:     return  $bel(\mathbf{x}_k)$ 

6:   Algorithmus Bayesfilter\_update( $bel(\mathbf{x}_{t-1}), \mathbf{z}_t, \mathbf{u}_t$ ) :
7:     for all  $\mathbf{x}_t$  do
8:        $\overline{bel}(\mathbf{x}_t) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$ 
9:        $bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \overline{bel}(\mathbf{x}_t)$ 
10:    endfor
11:    return  $bel(\mathbf{x}_t)$ 

```

TABELLE 3.1: Der Bayes-Filter-Algorithmus

können. Der Bayes-Filter hat sich dennoch als robust gegenüber Verletzungen der Markow-Annahme erwiesen [THRUN et al. 2005].

Allgemein formuliert lässt sich der Bayes-Filter-Algorithmus wie in Tabelle 3.1 beschreiben. Die rekursive Gleichung (3.4) wird durch die Zeilen 8-9 implementiert. In Zeile 8 wird die Vorhersage (engl. *prediction step*) bezüglich des Beliefs errechnet. Zeile 9 korrigiert (engl. *correction step*) diese mit Hilfe der aktuellen Messung. In der Praxis ist die Wahl der Repräsentation des Zustandsraumes von entscheidender Bedeutung, da Zeile 8 aufgrund des Dimensionsraumes von \mathbf{x} im Allgemeinen nicht mit vertretbarem Rechenaufwand berechenbar ist. In den nächsten Abschnitten werden deshalb die wichtigsten Implementierungen und Approximationen vorgestellt: der Kalman-Filter und der Informations-Filter, zwei parametrisierte Implementierungen für den Spezialfall eines normalverteilten Zustandsvektors, ihre nichtlinearen Approximationen, und der Partikel-Filter aus der Familie der stichprobenbasierten Schätzverfahren.

3.3 Kalman-Filter

Eine der bekanntesten und oft verwendeten Implementierungen des Bayes-Filters ist der Kalman-Filter. Es handelt sich dabei um einen parametrisierten Filter, der erstmals von Swerling (1958) und Kalman (1960) beschrieben wurde [KALMAN 1960]. Der Grundgedanke ist, $bel(\mathbf{x})$ durch eine (mehrdimensionale) Normalverteilung darzustellen. Dazu werden, zusätzlich zur Markow-Annahme, die folgenden Annahmen gemacht:

- Die Startverteilung $bel(\mathbf{x}_0)$ ist eine (mehrdimensionale) Normalverteilung $\mathcal{N}(\mathbf{x}_0; \mu_0, \Sigma_0)$.
- Die Zustandsübergangsfunktion $(\mathbf{x}_{t-1}, \mathbf{u}_t) \rightarrow \mathbf{x}_t$ mit $\mathbf{x}_t \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ ist eine in $\mathbf{x}_{t-1}, \mathbf{u}_t$ lineare Funktion mit additivem gaußschem Rauschen.
- Die Observationsfunktion $p(\mathbf{z}_t | \mathbf{x}_t)$ ist eine in \mathbf{x}_t lineare Funktion mit additivem gaußschem Rauschen.

Sind diese Voraussetzungen erfüllt, so ist $bel(\mathbf{x}_t)$ für alle Zeitpunkte t normalverteilt. Im Folgenden wird detailliert auf die Voraussetzungen eingegangen. Der Korrektheitsbeweis kann z. B. in [THRUN et al. 2005] nachgelesen werden.

Sei $bel(\mathbf{x}_0)$ eine Normalverteilung $\mathcal{N}(\mathbf{x}_0; \mu_0, \Sigma_0)$. Dann lässt sich $bel(\mathbf{x}_0)$ schreiben als

$$bel(\mathbf{x}_0) = p(\mathbf{x}_0) = \frac{1}{\sqrt{\det(2\pi\Sigma_0)}} e^{-\frac{1}{2}(\mathbf{x}_0 - \mu_0)^T \Sigma_0^{-1} (\mathbf{x}_0 - \mu_0)}. \quad (3.5)$$

Weiterhin seien die Funktionen $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ und $p(\mathbf{z}_t | \mathbf{x}_t)$ linear in $\mathbf{x}_{t-1}, \mathbf{u}_t$ bzw. \mathbf{x}_t mit additivem gaußschem Rauschen. Dann lässt sich \mathbf{x}_t als

$$\mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \epsilon_t \quad (3.6)$$

schreiben. Hierbei sind \mathbf{A}_t und \mathbf{B}_t Matrizen. Die Zufallsvariable ϵ_t ist normalverteilt mit Mittelwert $\mathbf{0}$ und Kovarianz \mathbf{R}_t . Die Verteilung $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ ist demnach definiert durch

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) = \frac{1}{\sqrt{\det(2\pi\mathbf{R}_t)}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{A}_t \mathbf{x}_{t-1} - \mathbf{B}_t \mathbf{u}_t)^T \mathbf{R}_t^{-1} (\mathbf{x}_t - \mathbf{A}_t \mathbf{x}_{t-1} - \mathbf{B}_t \mathbf{u}_t)}. \quad (3.7)$$

Ebenso lässt sich \mathbf{z}_t durch

$$\mathbf{z}_t = \mathbf{C}_t \mathbf{x}_t + \delta_t \quad (3.8)$$

mit einer Matrix \mathbf{C}_t darstellen. Die Zufallsvariable δ_t ist normalverteilt mit Mittelwert $\mathbf{0}$ und Kovarianz \mathbf{Q}_t . Die Verteilung $p(\mathbf{z}_t | \mathbf{x}_t)$ ist demnach definiert durch

$$p(\mathbf{z}_t | \mathbf{x}_t) = \frac{1}{\sqrt{\det(2\pi\mathbf{Q}_t)}} e^{-\frac{1}{2}(\mathbf{z}_t - \mathbf{C}_t \mathbf{x}_t)^T \mathbf{Q}_t^{-1} (\mathbf{z}_t - \mathbf{C}_t \mathbf{x}_t)}. \quad (3.9)$$

```

1:   Algorithmus Kalmanfilter( $\mu_0, \Sigma_0, \mathbf{z}_{1:k}, \mathbf{u}_{1:k}$ ) :
2:     for  $i = 0, \dots, k - 1$  do
3:        $\mu_{i+1}, \Sigma_{i+1} = \mathbf{Kalmanfilter\_update}(\mu_i, \Sigma_i, \mathbf{z}_{i+1}, \mathbf{u}_{i+1})$ 
4:     endfor
5:     return  $\mu_k, \Sigma_k$ 

6:   Algorithmus Kalmanfilter_update( $\mu_{t-1}, \Sigma_{t-1}, \mathbf{z}_t, \mathbf{u}_t$ ) :
7:      $\bar{\mu}_t = \mathbf{A}_t \mu_{t-1} + \mathbf{B}_t \mathbf{u}_t$ 
8:      $\bar{\Sigma}_t = \mathbf{A}_t \Sigma_{t-1} \mathbf{A}_t^T + \mathbf{R}_t$ 
9:      $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\Sigma}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1}$ 
10:     $\mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C}_t \bar{\mu}_t)$ 
11:     $\Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\Sigma}_t$ 
12:    return  $\mu_t, \Sigma_t$ 

```

TABELLE 3.2: Der Kalman-Filter-Algorithmus

Tabelle 3.2 beschreibt den Algorithmus in Pseudocode. Da der Zustand \mathbf{x}_t normalverteilt ist, kann dieser durch seine ersten beiden Momente μ_t und Σ_t repräsentiert werden. Zeilen 7-8 implementieren die Vorhersage der Momente entsprechend zur Zeile 8 in Tabelle 3.1. In Zeile 9 wird der *Kalman gain* \mathbf{K}_t berechnet. Dieser gibt an, in welchem Maße die aktuelle Messung \mathbf{z}_t die vorhergesagte Zustandsverteilung $\overline{bel}(\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_t; \bar{\mu}_t, \bar{\Sigma}_t)$ korrigiert. Zeilen 10-11 korrigieren die Vorhersage $\bar{\mu}_t, \bar{\Sigma}_t$ analog zur Zeile 9 in Tabelle 3.1. Abbildung 3.2 visualisiert das Verfahren anhand eines Beispiels mit einem Roboter. In der eindimensionalen Beispielwelt ist der zu schätzende Zustand die x -Position des Roboters. Abbildung 3.2a zeigt die Position des Roboters zum Zeitpunkt $t - 1$. Der Roboter bewegt sich im Intervall $(t - 1, t]$ entlang der x -Achse. Abbildung 3.2b stellt die Vorhersage bezüglich der neuen Position $\bar{\mathbf{x}}_t = \{\bar{\mu}_t, \bar{\Sigma}_t\}$ dar. Die Unsicherheit ϵ_t in Gleichung (3.6) führt einer erhöhten Varianz $\bar{\Sigma}_t$. Abbildung 3.2c: Der Roboter erhält eine neue Messung (z. B. GPS). Die Integration der Messung korrigiert den vorhergesagten Zustand. Dieser ist in Abbildung 3.2d abgebildet.

Eine wichtige Voraussetzung für den Kalman-Filter sind die Annahmen über die Linearität der Funktionen $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ und $p(\mathbf{z}_t | \mathbf{x}_t)$ in $\mathbf{x}_{t-1}, \mathbf{u}_t$ bzw. \mathbf{x}_t , da eine normalverteilte Zufallsvariable nach Transformation durch eine lineare Funktion wieder normalverteilt ist [THRUN et al. 2005]. In der Praxis ist Linearität meist nicht gegeben. So kann z. B. die Fahrt einer gekrümmten Trajektorie nicht durch eine lineare Funktion beschrieben werden. Daher existiert eine nichtlineare Erweiterung des Kalman-Filters (engl. *extended Kalman filter*, kurz: EKF).

3 Sequentielle Zustandsschätzung

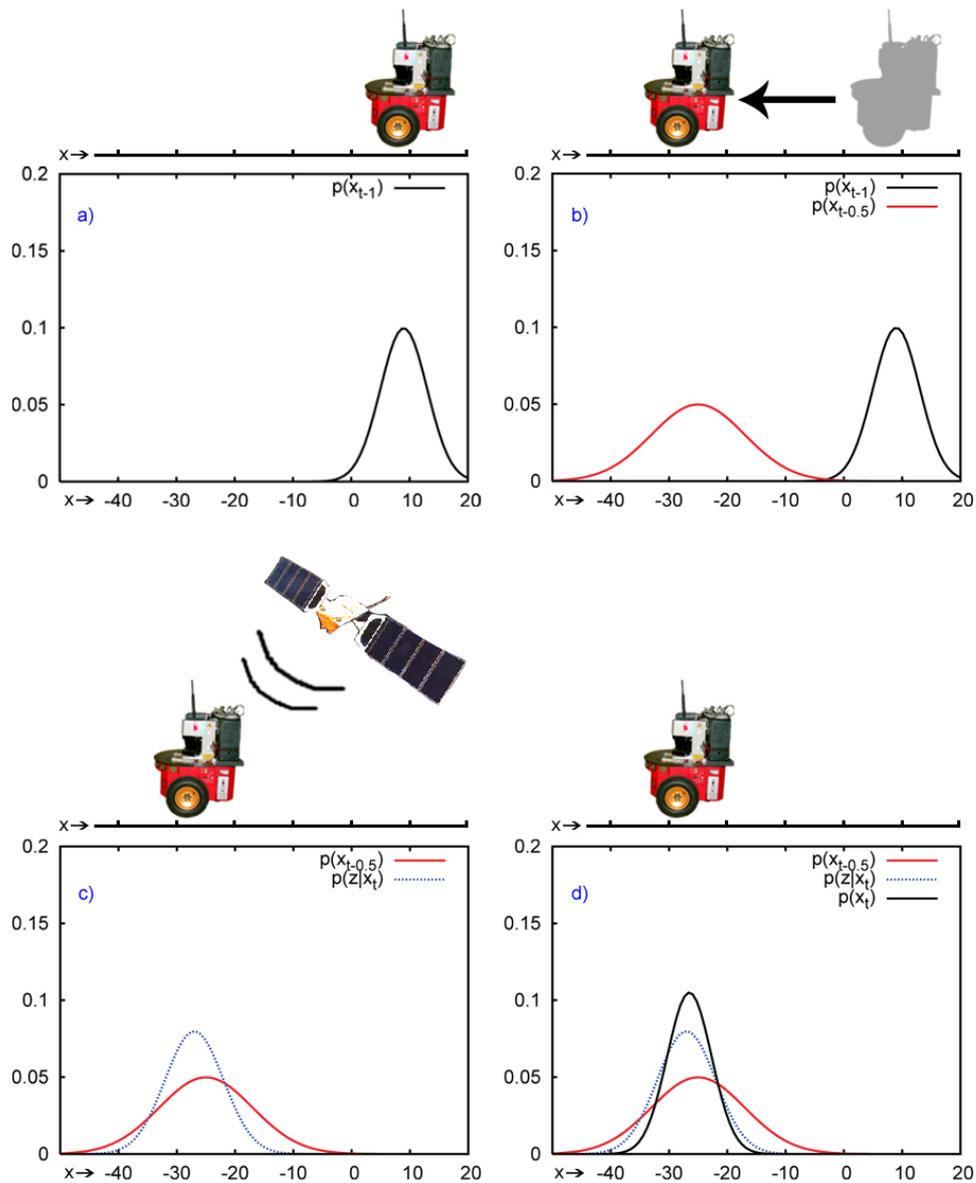


ABBILDUNG 3.2: Beispiel für die sequentielle Zustandsschätzung mit dem Kalman-Filter im eindimensionalen Fall. Der geschätzte Zustand ist die Position des Roboters. a) Die Position des Roboters zum Zeitpunkt $t - 1$. b) Der Roboter hat sich entlang der x -Achse bewegt. Die rote Kurve visualisiert die Vorhersage seiner neuen Position. c) Der Roboter erhält eine neue Messung. d) Integration der Messung führt zur neuen Positionsschätzung.

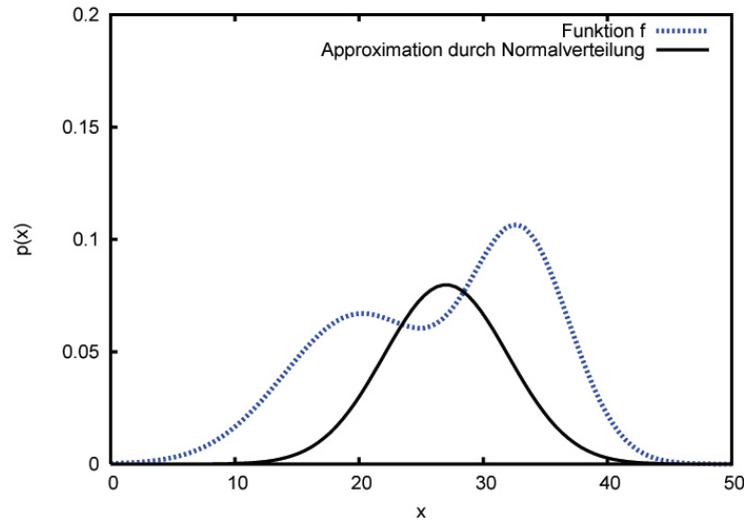


ABBILDUNG 3.3: Eine nichtlineare Funktion (blau, unterbrochen) und ihre gaußsche Approximation

Die nichtlineare Erweiterung des Kalman-Filters ersetzt die Voraussetzungen über die Funktionen $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$ und $p(\mathbf{z}_t|\mathbf{x}_t)$ durch:

- Die Zustandsübergangsfunktion $(\mathbf{x}_{t-1}, \mathbf{u}_t) \rightarrow \mathbf{x}_t$ mit $\mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$ ist eine nichtlineare Funktion mit additivem gaußschem Rauschen.
- Die Observationsfunktion $p(\mathbf{z}_t|\mathbf{x}_t)$ ist eine nichtlineare Funktion mit additivem gaußschem Rauschen.

Dadurch lässt sich \mathbf{x}_t durch

$$\mathbf{x}_t = g(\mathbf{u}_t, \mathbf{x}_{t-1}) + \epsilon_t \quad (3.10)$$

und entsprechend \mathbf{z}_t durch

$$\mathbf{z}_t = h(\mathbf{x}_t) + \delta_t \quad (3.11)$$

mit nichtlinearen Funktionen g und h beschreiben. Die resultierende Zustandsverteilung $bel(\mathbf{x})$ wird beim EKF durch eine Normalverteilung (siehe Abbildung 3.3) approximiert. Dazu werden die Funktionen g und h durch ihre Taylorentwicklung ersten Grades linear angenähert. Dadurch lässt sich \mathbf{x}_t durch

$$\mathbf{x}_t \approx g(\mathbf{u}_t, \mu_{t-1}) + \underbrace{g'(\mathbf{u}_t, \mu_{t-1})}_{=: \mathbf{G}_t} (\mathbf{x}_{t-1} - \mu_{t-1}) + \epsilon_t \quad (3.12)$$

```

1:   Algorithmus Extended_Kalmanfilter( $\mu_0, \Sigma_0, \mathbf{z}_{1:k}, \mathbf{u}_{1:k}$ ) :
2:     for  $i = 0, \dots, k - 1$  do
3:        $\mu_{i+1}, \Sigma_{i+1} = \text{Extended\_Kalmanfilter\_update}(\mu_i, \Sigma_i, \mathbf{z}_{i+1}, \mathbf{u}_{i+1})$ 
4:     endfor
5:     return  $\mu_k, \Sigma_k$ 

6:   Algorithmus Extended_Kalmanfilter_update( $\mu_{t-1}, \Sigma_{t-1}, \mathbf{z}_t, \mathbf{u}_t$ ) :
7:      $\bar{\mu}_t = g(\mathbf{u}_t, \mu_{t-1})$ 
8:      $\bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t$ 
9:      $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1}$ 
10:     $\mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t))$ 
11:     $\Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t$ 
12:    return  $\mu_t, \Sigma_t$ 

```

TABELLE 3.3: Der *Extended-Kalman-Filter*-Algorithmus

mit

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) \approx \frac{1}{\sqrt{\det(2\pi \mathbf{R}_t)}} \cdot e^{-\frac{1}{2} [\mathbf{x}_t - g(\mathbf{u}_t, \mu_{t-1}) - \mathbf{G}_t (\mathbf{x}_{t-1} - \mu_{t-1})]^T \mathbf{R}_t^{-1} [\mathbf{x}_t - g(\mathbf{u}_t, \mu_{t-1}) - \mathbf{G}_t (\mathbf{x}_{t-1} - \mu_{t-1})]} \quad (3.13)$$

und entsprechend \mathbf{z}_t durch

$$\mathbf{z}_t \approx h(\bar{\mu}_t) + \underbrace{h'(\bar{\mu}_t)}_{=: \mathbf{H}_t} (\mathbf{x}_t - \bar{\mu}_t) + \delta_t \quad (3.14)$$

mit

$$p(\mathbf{z}_t | \mathbf{x}_t) \approx \frac{1}{\sqrt{\det(2\pi \mathbf{Q}_t)}} e^{-\frac{1}{2} [\mathbf{z}_t - h(\bar{\mu}_t) - \mathbf{H}_t (\mathbf{x}_t - \bar{\mu}_t)]^T \mathbf{Q}_t^{-1} [\mathbf{z}_t - h(\bar{\mu}_t) - \mathbf{H}_t (\mathbf{x}_t - \bar{\mu}_t)]} \quad (3.15)$$

beschreiben. Tabelle 3.3 beschreibt den EKF in Pseudocode. Eine mathematische Herleitung der nichtlinearen Erweiterung des Kalman-Filters kann z. B. in [THRUN et al. 2005] nachgelesen werden. Neben der Approximation durch die Taylorentwicklung bis zum Grad eins existieren weitere nichtlineare Erweiterungen des Kalman-Filters. Diese werden beispielsweise in [RISTIC et al. 2004] beschrieben.

3.4 Informations-Filter

Der Informations-Filter (engl. *information filter*, kurz: IF) repräsentiert, genauso wie der Kalman-Filter bzw. seine nichtlineare Erweiterung, den *Belief* durch eine (mehrdimensionale) Normalverteilung. Der Unterschied zum Kalman-Filter besteht in der Repräsentation der Normalverteilung. Der Kalman-Filter repräsentiert die Normalverteilung durch die ersten beiden Momente μ , Σ . Im Informations-Filter wird die Normalverteilung durch den Informationsvektor ξ und die Informationsmatrix Ω mit

$$\xi = \Sigma^{-1}\mu \quad (3.16)$$

$$\Omega = \Sigma^{-1} \quad (3.17)$$

repräsentiert. Die Wahl von ξ und Ω lässt sich aus der Normalverteilung herleiten. Eine Normalverteilung mit Mittelwert μ und Kovarianz Σ ist definiert durch

$$p(x) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)}. \quad (3.18)$$

Ausmultiplizieren von (3.18) führt zu:

$$\begin{aligned} p(x) &= \frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x} + \mathbf{x}^T \Sigma^{-1} \mu - \frac{1}{2}\mu^T \Sigma^{-1} \mu} \\ &= \underbrace{\frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{1}{2}\mu^T \Sigma^{-1} \mu}}_{\textit{konstant}} e^{-\frac{1}{2}\mathbf{x}^T \Sigma^{-1} \mathbf{x} + \mathbf{x}^T \Sigma^{-1} \mu}. \end{aligned} \quad (3.19)$$

Mit $\xi = \Sigma^{-1}\mu$ und $\Omega = \Sigma^{-1}$ wird (3.19) zu

$$p(x) = \eta e^{-\frac{1}{2}\mathbf{x}^T \Omega \mathbf{x} + \mathbf{x}^T \xi}. \quad (3.20)$$

Unter Annahme derselben Voraussetzungen wie beim Kalman-Filter ($\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \mu_0, \Sigma_0)$, Zustandsübergangsfunktion (3.6) und Observationsfunktion (3.8) sind linear) lässt sich der Informationsfilter wie in Tabelle 3.4 beschreiben. Analog zur nichtlinearen Erweiterung des Kalman-Filters existiert eine nichtlineare Erweiterung des Informations-Filters (engl. *extended information filter*, kurz: EIF) basierend auf der Taylorapproximation der Funktionen g und h (siehe (3.12) und (3.14)). Tabelle 3.5 beschreibt diesen nach [THRUN et al. 2005].

Gerade im Kontext der zeitgleichen Lokalisierung und Kartengenerierung (siehe Kapitel 4.4) hat der EIF Vorteile gegenüber dem EKF. Während sich zwischen zwei Iterationsschritten die Kovarianzmatrix Σ signifikant ändert, werden in der Informationsmatrix Ω nur wenige Einträge geändert. Zudem hat dort die Informationsmatrix eine quasi-diagonale Form. [THRUN et al. 2004a] erweitern den EIF zum *sparse extended information filter* (kurz: SEIF) durch Approximation der Informationsmatrix (pro Iterationsschritt werden maximal n Einträge geändert).

```

1:   Algorithmus Informationfilter( $\xi_0, \Omega_0, \mathbf{z}_{1:k}, \mathbf{u}_{1:k}$ ) :
2:     for  $i = 0, \dots, k - 1$  do
3:        $\xi_{i+1}, \Omega_{i+1} = \mathbf{Informationfilter\_update}(\xi_i, \Omega_i, \mathbf{z}_{i+1}, \mathbf{u}_{i+1})$ 
4:     endfor
5:     return  $\xi_k, \Omega_k$ 

6:   Algorithmus Informationfilter_update( $\xi_{t-1}, \Omega_{t-1}, \mathbf{z}_t, \mathbf{u}_t$ ) :
7:      $\bar{\Omega}_t = (\mathbf{A}_t \Omega_{t-1}^{-1} \mathbf{A}_t^T + \mathbf{R}_t)^{-1}$ 
8:      $\bar{\xi}_t = \bar{\Omega}_t (\mathbf{A}_t \Omega_{t-1}^{-1} \xi_{t-1} + \mathbf{B}_t \mathbf{u}_t)$ 
9:      $\Omega_t = \bar{\Omega}_t + \mathbf{C}_t^T \mathbf{Q}_t^{-1} \mathbf{C}_t$ 
10:     $\xi_t = \bar{\xi}_t + \mathbf{C}_t^T \mathbf{Q}_t^{-1} \mathbf{z}_t$ 
11:    return  $\xi_t, \Omega_t$ 

```

TABELLE 3.4: Der *Information-Filter-Algorithmus*

```

1:   Algorithmus Extended_Informationfilter( $\xi_0, \Omega_0, \mathbf{z}_{1:k}, \mathbf{u}_{1:k}$ ) :
2:     for  $i = 0, \dots, k - 1$  do
3:        $\xi_{i+1}, \Omega_{i+1} = \mathbf{Extended\_Informationfilter\_update}(\xi_i, \Omega_i, \mathbf{z}_{i+1}, \mathbf{u}_{i+1})$ 
4:     endfor
5:     return  $\xi_k, \Omega_k$ 

6:   Algorithmus Extended_Informationfilter_update( $\xi_{t-1}, \Omega_{t-1}, \mathbf{z}_t, \mathbf{u}_t$ ) :
7:      $\mu_{t-1} = \Omega_{t-1}^{-1} \xi_{t-1}$ 
8:      $\bar{\Omega}_t = (\mathbf{G}_t \Omega_{t-1}^{-1} \mathbf{G}_t^T + \mathbf{R}_t)^{-1}$ 
9:      $\bar{\xi}_t = \bar{\Omega}_t g(\mathbf{u}_t, \mu_{t-1})$ 
10:     $\bar{\mu}_t = g(\mathbf{u}_t, \mu_{t-1})$ 
11:     $\Omega_t = \bar{\Omega}_t + \mathbf{H}_t^T \mathbf{Q}_t^{-1} \mathbf{H}_t$ 
12:     $\xi_t = \bar{\xi}_t + \mathbf{H}_t^T \mathbf{Q}_t^{-1} [\mathbf{z}_t - h(\bar{\mu}_t) - \mathbf{H}_t \bar{\mu}_t]$ 
13:    return  $\xi_t, \Omega_t$ 

```

TABELLE 3.5: Der *Extended-Information-Filter-Algorithmus*

Durch diese Approximation und geeignete Berechnung des Mittelwertes μ kann der SEIF mit amortisiert konstanten Kosten implementiert werden. [EUSTICE et al. 2005] zeigen, dass im *delayed-state SLAM-framework*² die Informationsmatrix Ω dünn besetzt (engl. *sparse*) ist, wodurch dort keine Approximation der Informationsmatrix notwendig ist.

3.5 Partikel-Filter

In diesem Abschnitt wird eine nichtparametrisierte Approximation des Bayes-Filters vorgestellt, die sich in vielen Anwendungsgebieten als vorteilhaft herausgestellt hat und die Grundlage dieser Arbeit darstellt: der Partikel-Filter. Der Partikel-Filter approximiert die a-posteriori-Verteilung $bel(\mathbf{x})$ eines Systems durch eine endliche Menge von Stichproben. Die Grundidee ist, eine Wahrscheinlichkeitsverteilung durch eine gewichtete Stichprobenmenge (die Partikel) zu approximieren, wie in Abbildung 3.5b dargestellt. Im Gegensatz zu alternativen Ansätzen wie z. B. dem Kalman-Filter können Partikel-Filter durch ihre nichtparametrisierte Form beliebige, insbesondere multimodale, Verteilungen approximieren. In der Praxis stellt die Endlichkeit der Partikelmenge jedoch eine wichtige Einschränkung dar. Diese Einschränkung kann teils durch adaptives Anpassen der Partikelanzahl, wie in [FOX 2003] beschrieben, ausgeglichen werden. Ein weiterer Vorteil dieser Repräsentation ist die Möglichkeit, nichtlineare Transformationen direkt anzuwenden zu können. Für den formalen Umgang mit dem Partikel-Filter wird die Notation aus Abschnitt 3.1 folgendermaßen erweitert:

$\mathbf{x}_t^{[i]}$	Durch Partikel i repräsentierter Zustand zum Zeitpunkt t
$w_t^{[i]}$	Gewicht des Partikels i zum Zeitpunkt t
χ_t	$:= \{ \langle \mathbf{x}_t^{[1]}, w_t^{[1]} \rangle, \langle \mathbf{x}_t^{[2]}, w_t^{[2]} \rangle, \dots, \langle \mathbf{x}_t^{[M]}, w_t^{[M]} \rangle \}$
$\mathbf{x}_{1:t}^{[i]}$	$:= \mathbf{x}_1^{[i]}, \mathbf{x}_2^{[i]}, \dots, \mathbf{x}_t^{[i]}$

Tabelle 3.6 beschreibt die Grundversion des Partikel-Filter-Algorithmus in Pseudocode.

Im Detail lässt sich der Algorithmus wie folgt erklären:

Zeile 4 generiert den Folgezustand $\mathbf{x}_t^{[i]}$ für Zeitpunkt t basierend auf dem bisherigen Zustand $\mathbf{x}_{t-1}^{[i]}$ und dem Kommando \mathbf{u}_t (z. B. Bewegung). Da die Zustandsübergangsfunktion in nicht-deterministischer Form gegeben ist, wird aus der Verteilung $p(\mathbf{x}_t^{[i]} | \mathbf{u}_t, \mathbf{x}_{t-1}^{[i]})$ gezogen (analog zu Zeile 8 in 3.1). Zeile 5 berechnet für jedes Partikel das Gewicht (engl. *importance factor*) $w_t^{[i]}$. Da das verwendete Zustandsübergangsmodell vom wahren unbekanntem Übergangsmodell

²Im *delayed-state SLAM-framework* besteht der Zustandsvektor aus der aktuellen Position und „Schlüsselpositionen“ aus der Vergangenheit

```

1:   Algorithmus Partikelfilter( $\chi_{t-1}, \mathbf{u}_t, \mathbf{z}_t$ ) :
2:    $\bar{\chi}_t = \chi_t = \emptyset$ 
3:   for  $i = 1$  to  $M$  do
4:     ziehe  $\mathbf{x}_t^{[i]}$  aus  $p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}^{[i]})$ 
5:      $w_t^{[i]} = \eta p(\mathbf{z}_t | \mathbf{x}_t^{[i]}) w_{t-1}^{[i]}$ 
6:      $\bar{\chi}_t = \bar{\chi}_t \cup \langle \mathbf{x}_t^{[i]}, w_t^{[i]} \rangle$ 
7:   endfor
8:   for  $i = 1$  to  $M$  do
9:     ziehe  $j$  mit Wahrscheinlichkeit proportional zu  $w_t^{[j]}$ 
10:     $\chi_t = \chi_t \cup \langle \mathbf{x}_t^{[j]}, \frac{1}{M} \rangle$ 
11:  endfor
12:  return  $\chi_t$ 

```

TABELLE 3.6: Grundversion des Partikel-Filter-Algorithmus

abweicht, müssen die Partikel durch Anpassung ihrer Gewichte mit

$$w_t^{[i]} = \frac{\overbrace{p(\mathbf{x}_{1:t}^{[i]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}}{\text{wahre Übergangsfunktion}}}{\underbrace{\pi(\mathbf{x}_{1:t}^{[i]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}_{\text{verwendete Übergangsfunktion}}} \quad (3.21)$$

korrigiert werden. Unterliegt die Zustandsübergangsfunktion π der Markow-Annahme und besitzt die Eigenschaft

$$\pi(\mathbf{x}_{1:t}^{[i]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = \pi(\mathbf{x}_t^{[i]} | \mathbf{x}_{1:t-1}^{[i]}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \pi(\mathbf{x}_{1:t-1}^{[i]} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1}), \quad (3.22)$$

kann (3.21) zu

$$w_t^{[i]} = \frac{p(\mathbf{x}_{1:t}^{[i]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}{\pi(\mathbf{x}_{1:t}^{[i]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})} = \eta \frac{p(\mathbf{z}_t | \mathbf{x}_t^{[i]}) p(\mathbf{x}_t^{[i]} | \mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t) p(\mathbf{x}_{1:t-1}^{[i]} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})}{\pi(\mathbf{x}_t^{[i]} | \mathbf{x}_{1:t-1}^{[i]}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \pi(\mathbf{x}_{1:t-1}^{[i]} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})} \quad (3.23)$$

$$= \eta w_{t-1}^{[i]} \frac{p(\mathbf{z}_t | \mathbf{x}_t^{[i]}) p(\mathbf{x}_t^{[i]} | \mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t)}{\pi(\mathbf{x}_t^{[i]} | \mathbf{x}_{1:t-1}^{[i]}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})} \quad (3.24)$$

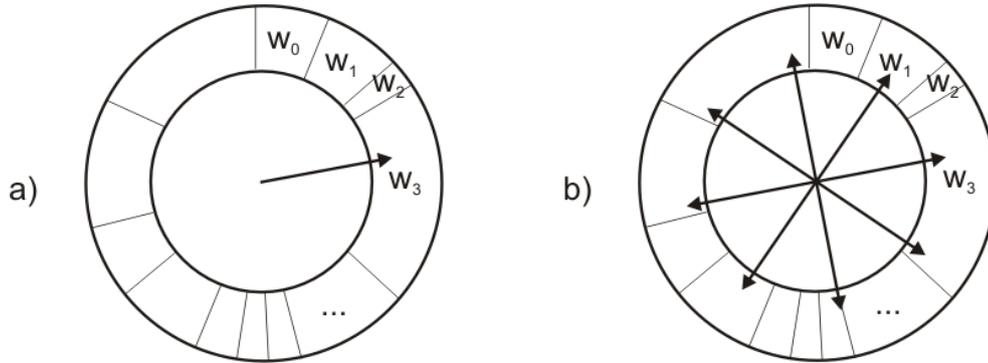


ABBILDUNG 3.4: Resampling-Schritt: a) einfaches Resampling proportional zum Gewicht (Roulette-Resampling). b) Resampling nach dem *Low-Variance-Sampling*-Verfahren.

umgeformt werden. Wird dabei (wie in Tabelle 3.6 implementiert) $p(\mathbf{x}_t^{[i]} | \mathbf{u}_t, \mathbf{x}_{t-1}^{[i]})$ als Zustansübergangsmodell verwendet, vereinfacht sich (3.24) zu

$$w_t^{[i]} = \eta w_{t-1}^{[i]} \frac{p(\mathbf{z}_t | \mathbf{x}_t^{[i]}) p(\mathbf{x}_t^{[i]} | \mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t)}{p(\mathbf{x}_t^{[i]} | \mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t)} \quad (3.25)$$

$$= \eta w_{t-1}^{[i]} p(\mathbf{z}_t | \mathbf{x}_t^{[i]}). \quad (3.26)$$

Die Menge der gewichteten Partikel repräsentiert die a-posteriori-Verteilung.

Zeilen 8-11 implementieren den Resampling-Schritt (*importance resampling*). Dieser Schritt bewirkt eine Konzentration der Partikelmenge auf die wahrscheinlichsten Bereiche des Zustandsraumes. Es wird eine Auswahl aus der Partikelmenge gemäß ihrer Gewichte getroffen. Das entspricht anschaulich dem Prinzip „der Stärkste überlebt“, wie es auch bei Genetischen Algorithmen angewendet wird. Ein Partikel überlebt (d. h. es kommt in die nächste Iteration) mit einer Wahrscheinlichkeit proportional zu seinem Gewicht. Nach einem Resampling-Schritt sind die Partikel gleichverteilt. In dieser Arbeit wird für den Resampling-Schritt das in [THRUN et al. 2005] vorgestellte *low-variance-sampling*-Verfahren verwendet. Hier wird ein Partikel nicht einfach unabhängig von den anderen Partikeln proportional zu seinem Gewicht gewählt, welches einem Rouletteziehen gleichkommt (siehe Abbildung 3.4a). Das verwendete Verfahren besteht aus drei Schritten: Zuerst werden die Partikel gemäß ihrer Gewichte aneinandergereiht. Anschließend wird zufällig ein Startpartikel gewählt und von diesem aus die restliche Anzahl an Partikeln zu gleichen Abständen entlang der Gewichtsachse gezogen. Die Arbeitsweise wird durch Abbildung 3.4b veranschaulicht.

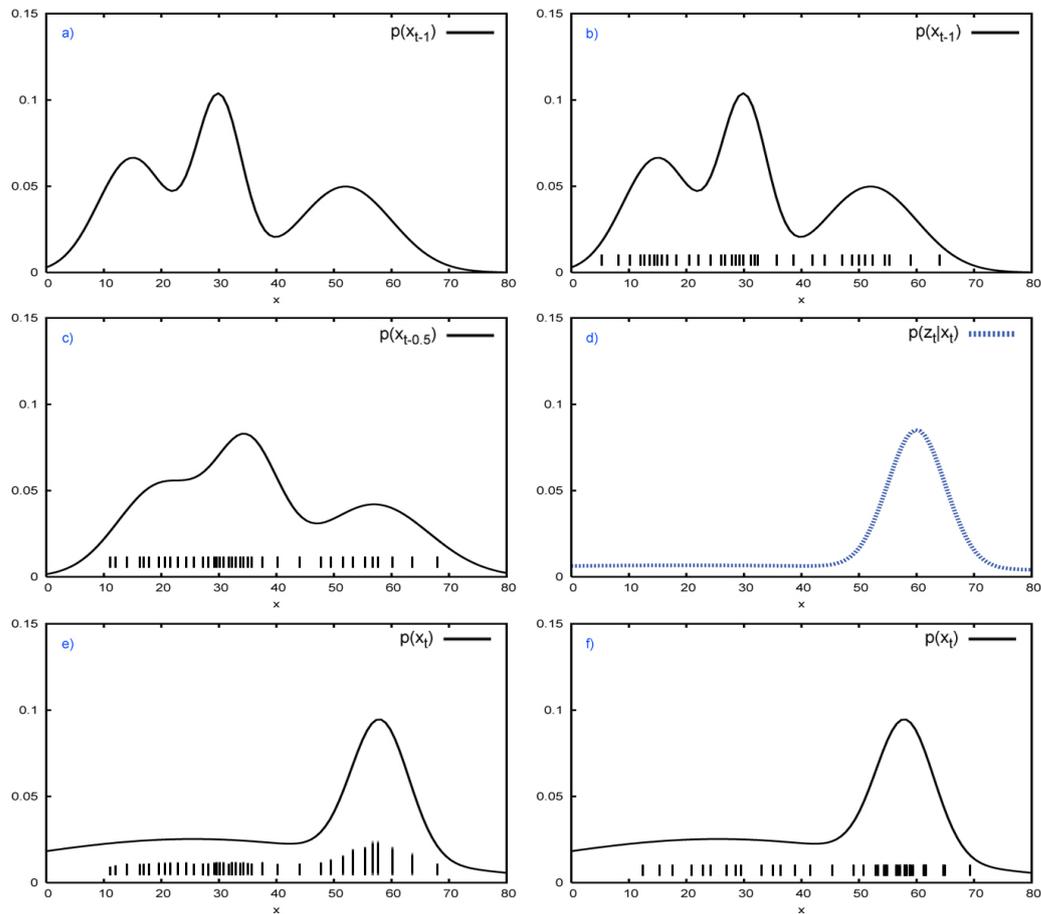


ABBILDUNG 3.5: Der Partikel-Filter-Algorithmus: a) Die a-priori-Verteilung des Zustands und (b) mit der zugehörigen Repräsentation durch Partikel. c) Der Filterzustand nach Anwendung des Bewegungsmodells und die dadurch approximierte Darstellung. d) Die Messung in der aktuellen Iteration. e) Die durch die Messung angepassten Gewichte der Partikel und die approximierte Verteilung. f) nach dem Resampling-Schritt: Partikel werden proportional zu ihrem Gewicht (ggf. mehrfach oder gar nicht) kopiert.

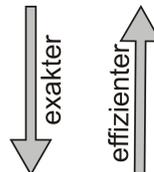
4 Grundlagen der Lokalisierung und Kartengenerierung

In Kapitel 3 wurden die Grundlagen der sequentiellen Zustandsschätzung basierend auf dem Bayes-Filter beschrieben. Aufbauend auf den dort vorgestellten Verfahren können nun Algorithmen zur Lokalisierung und Kartengenerierung erläutert werden. Dazu werden zunächst mögliche Kartendarstellungen vorgestellt und ein partikelfilterbasiertes Verfahren zur Lokalisierung bei bekannter Karte erläutert. Anschließend wird in Kapitel 4.3 auf den Aufbau einer Rasterkarte bei gegebener Trajektorie eingegangen. Schließlich wird das Problem der zeitgleichen Lokalisierung und Kartengenerierung (engl. *simultaneous localization and mapping*, kurz: SLAM) beschrieben und auf den in dieser Arbeit verwendeten Algorithmus eingegangen: der *Rao-Blackwellized Particle Filter*.

4.1 Kartenrepräsentation

Es existieren eine Reihe von Kartenrepräsentationen, von denen die wichtigsten kurz vorgestellt werden sollen:

- topologische Karten
- landmarkenbasierte Karten
- Rasterkarten (engl. *Occupancy Grid Maps*)



In topologischen Karten wird die Umgebung durch einen Graphen repräsentiert. Knoten im Graphen stehen für Bereiche, die beispielsweise durch eine Tür voneinander getrennt sind. Verbindungen zwischen Bereichen werden durch Kanten wiedergegeben. Topologische Karten liefern eine grobe Darstellung der Umgebung. Ein Vorteil von topologischen Karten liegt in der Lokalisierung des Roboters, da der Roboter lediglich wissen muss, in welchem Bereich er sich befindet. Die Nachteile liegen in der Navigation: Der konkrete Aufbau eines Bereiches einschließlich Hindernissen (z. B. Tische) wird von topologischen Karten nicht erfasst. Eine Navigation des Roboters kann sich deshalb nicht auf metrische Informationen stützen. Um dieses Problem ansatzweise zu kompensieren, verwenden [NEHMZOW et al. 1993] verhaltensgesteuerte Navigation. Das Bewegungsrepertoire des Roboters besteht dort aus Aktionen wie beispielsweise „folge dem Korridor“ oder „passiere Tür“ [OWEN und NEHMZOW 1996, IGLESIAS et al. 2005].

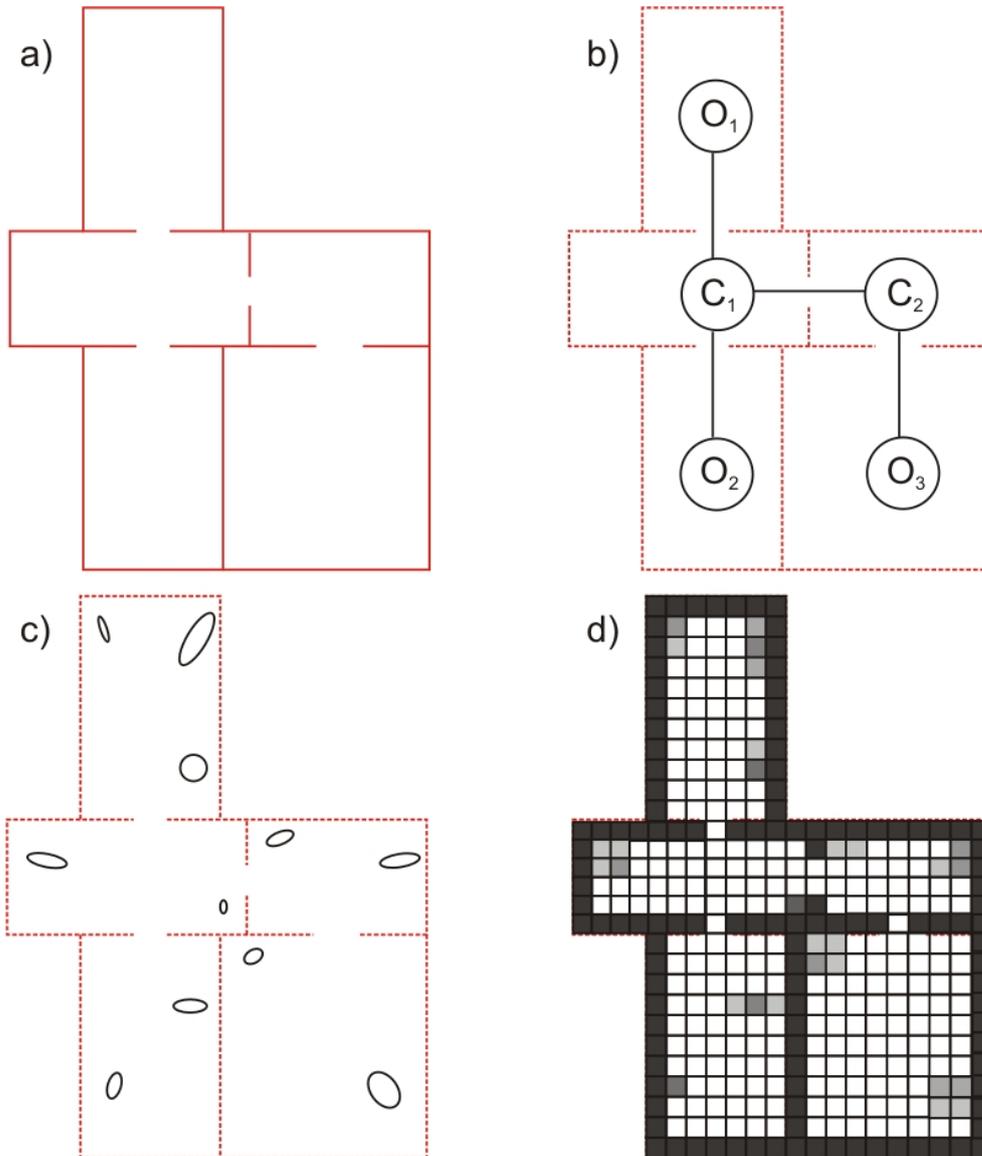


ABBILDUNG 4.1: Verschiedene Repräsentationen eines Gebäudes. a) Das Gebäude als Grundriss, b) als topologische Karte (mit den Knoten $Office_{1..3}$ und $Corridor_{1,2}$) und c) seine Repräsentation durch Landmarken (Orientierungspunkte). d) Das Gebäude als *Occupancy Grid Map*. Der Grauwert einer Zelle ist proportional zur Wahrscheinlichkeit, dass diese belegt ist.

```

1:   Algorithmus Markov_Localization( $bel(\mathbf{x}_0), \mathbf{z}_{1:k}, \mathbf{u}_{1:k}, \mathbf{m}$ ) :
2:     for  $i = 0, \dots, k - 1$  do
3:        $bel(\mathbf{x}_{i+1}) = \text{Markov\_Localization\_update}(bel(\mathbf{x}_i), \mathbf{z}_{i+1}, \mathbf{u}_{i+1}, \mathbf{m})$ 
4:     endfor
5:     return  $bel(\mathbf{x}_k)$ 

6:   Algorithmus Markov_Localization_update( $bel(\mathbf{x}_{t-1}), \mathbf{z}_t, \mathbf{u}_t, \mathbf{m}$ ) :
7:     for all  $\mathbf{x}_t$  do
8:        $\overline{bel(\mathbf{x}_t)} = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$ 
9:        $bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) \overline{bel(\mathbf{x}_t)}$ 
10:    endfor
11:    return  $bel(\mathbf{x}_t)$ 

```

TABELLE 4.1: Der Bayes-Filter-Algorithmus zur Lokalisierung bei bekannter Karte. In der Literatur ist dieses Verfahren als *Markov Localization* bekannt.

Landmarkenbasierte Karten repräsentieren ihre Umgebung durch eine Menge von Merkmalen. Sie enthalten metrische Informationen der Orientierungspunkte (erwartete Position) zusammen mit ihrer Unsicherheit (siehe Abbildung 4.1c). Landmarken können beliebige Merkmale sein, beispielsweise Bäume im Außenbereich. Bei der Kartengenerierung wird hierbei häufig der EKF oder der EIF verwendet ([EUSTICE et al. 2005, THRUN et al. 2004a, THRUN 2002]). Dabei besteht der Zustandsvektor \mathbf{x} aus dem Mittelwert der Position des Roboters, der Merkmale und ihrer Unsicherheiten ($\mathbf{x} = \{\mu_{\text{Roboter}}, \Sigma_{\text{Roboter}}, \mu_{\text{Merkmal}_1}, \Sigma_{\text{Merkmal}_1}, \dots, \Sigma_{\text{Merkmal}_n}\}$). Bei der in dieser Arbeit verwendeten Kartenstruktur handelt es sich um Rasterkarten (engl. *Occupancy Grid Maps*). Rasterkarten diskretisieren die Umgebung in Zellen einer bestimmten Größe (z. B. $5\text{cm} \times 5\text{cm}$). Eine Zelle enthält hierbei die Wahrscheinlichkeit, dass diese für den Roboter unpassierbar (belegt) ist. Abbildung 4.1 visualisiert die Repräsentationen anhand eines Beispiels. Das Gebäude in 4.1a wird durch eine topologische (4.1b), eine landmarkenbasierte (4.1c) und eine gerasterte (4.1d) Karte repräsentiert.

4.2 Lokalisierung mit dem Partikel-Filter bei bekannter Karte

Bei der Lokalisierung mit dem Partikel-Filter bei bekannter Karte (in der Literatur meist als *Monte Carlo Localization*, kurz: MCL beschrieben, siehe auch [FOX et al. 1999]) wird der Bayes-Filter aus Kapitel 3.2 um die Information der Karte erweitert (siehe Tabelle 4.1). Abbildung 4.2 visualisiert die daraus resultierenden Abhängigkeiten unter der Markow-Annahme.

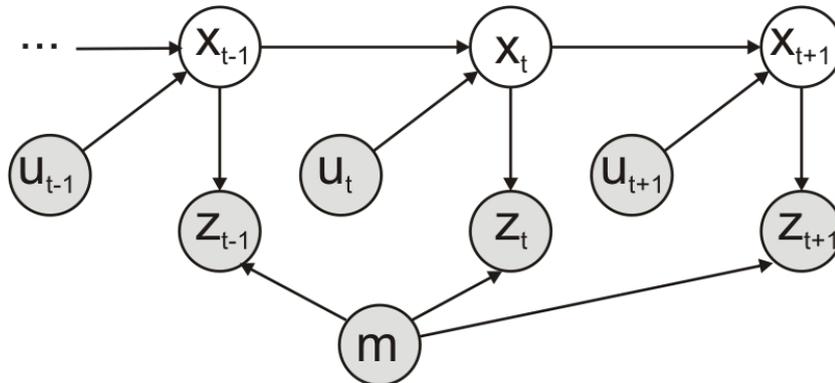


ABBILDUNG 4.2: Abhängigkeiten zwischen den Zuständen, Messungen und Kommandos unter der Markow-Annahme bei bekannter Karte \mathbf{m} . Die Werte der grau unterlegten Variablen sind bekannt.

Analog zu Kapitel 3.5 kann die Approximation durch den Partikel-Filter angegeben werden. Im Algorithmus aus Tabelle 3.6 wird hierzu die Information über die Karte hinzugefügt (siehe Zeile 5). Das daraus resultierende Verfahren zur Lokalisierung mit dem Partikel-Filter ist in Tabelle 4.2 beschrieben. Abbildung 4.4 visualisiert das Verfahren anhand eines Beispiels.

Die meisten Roboter mit Radantrieb, die heute kommerziell vertrieben werden, besitzen Sensoren an den Rädern mit denen der gefahrene Pfad gemessen werden kann. Diese Messungen werden Odometrie genannt. Die Odometrie \mathbf{u}_t dient als Basis für das Zustandsübergangsmodell $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ mit $\mathbf{x}_t = (x_t, y_t, \theta_t)$ (siehe Zeile 4 in Tabelle 4.2). Unterschiedliche Reibung der Räder, z. B. durch unterschiedlichen Luftdruck oder Untergrund, führen zwangsläufig zu einem Pfad, der von der Odometrie abweicht. Der daraus resultierende über die Zeit akkumulierte Fehler macht eine direkte Verwendung von Odometrieinformationen unmöglich. Daher wird die Odometrie als Basis für eine Normalverteilung, abhängig vom zurückgelegten Pfad, genommen [THRUN et al. 2005]. Dies wird in Abbildung 4.3 verdeutlicht. Für eine kontrastreichere Darstellung ist in Abbildung 4.3 die 2-D-Aufsicht der 3-D-Zustandsübergangsfunktion des Roboters visualisiert.

Zeile 5 in Tabelle 4.2 beschreibt die Observationsfunktion $p(\mathbf{z}_t | \mathbf{x}_{t-1}, \mathbf{m})$. Diese Verteilung wird in der Praxis meist durch eine Mischverteilung bestehend aus vier Wahrscheinlichkeitsverteilungen (uniforme Verteilung für zufällige Messungen, exponentielle Verteilung für dynamische Hindernisse, Normalverteilung für mögliche gemessene Hindernisse entlang des Messstrahls und eine Punktverteilung für den Fall einer Maximalmessung, falls kein Hindernis gesehen wurde) implementiert. Die Mischparameter sind hierbei vom Sensor und von der Distanz zum nächstgelegenen Objekt abhängig und können z. B. in einen vorangehenden Arbeitsschritt erlernt werden [THRUN et al. 2005].

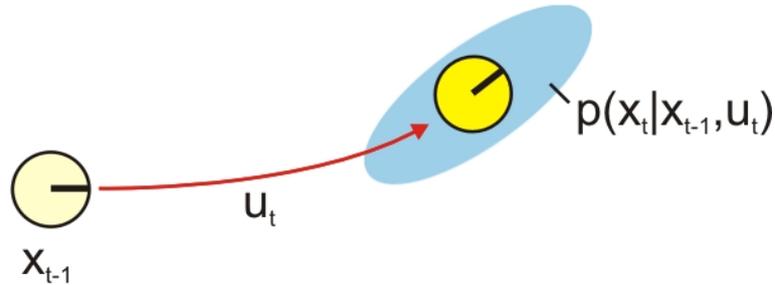


ABBILDUNG 4.3: Das Zustandsübergangsmodell $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ wird durch eine Normalverteilung basierend auf der Odometrie \mathbf{u}_t beschrieben.

1:	Algorithmus MCL ($\chi_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{m}$) :
2:	$\bar{\chi}_t = \chi_t = \emptyset$
3:	for $i = 1$ to M do
4:	$\mathbf{x}_t^{[i]} = \text{sample_motion_model}(\mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t)$
5:	$w_t^{[i]} = w_{t-1}^{[i]} \cdot \text{measurement_model}(\mathbf{z}_t, \mathbf{x}_t^{[i]}, \mathbf{m})$
6:	$\bar{\chi}_t = \bar{\chi}_t \cup \langle \mathbf{x}_t^{[i]}, w_t^{[i]} \rangle$
7:	endfor
8:	normalizeWeights ($\bar{\chi}_t$)
9:	for $i = 1$ to M do
10:	ziehe j mit Wahrscheinlichkeit proportional zu $w_t^{[j]}$
11:	$\chi_t = \chi_t \cup \langle \mathbf{x}_t^{[j]}, \frac{1}{M} \rangle$
12:	endfor
13:	return χ_t

TABELLE 4.2: der Monte-Carlo-Localization-Algorithmus

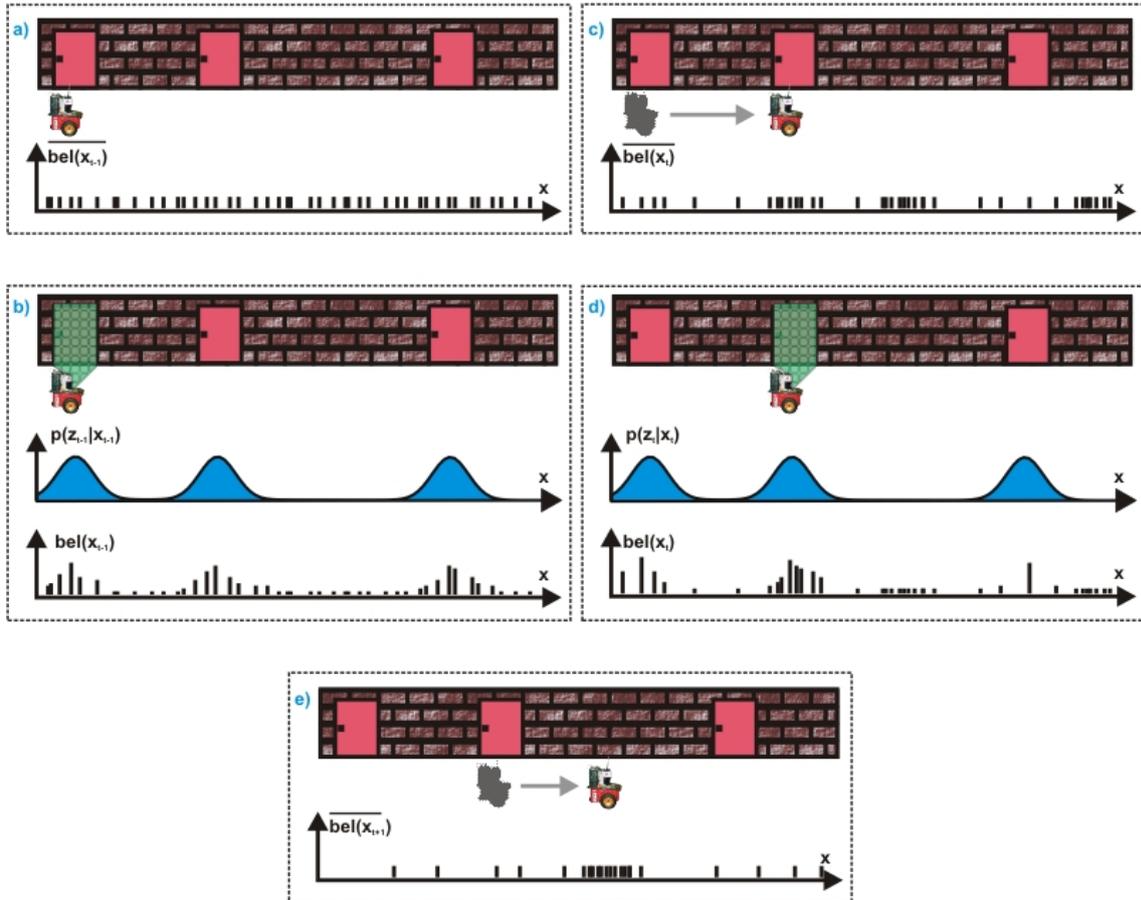


ABBILDUNG 4.4: Lokalisierung mit dem Partikel-Filter bei bekannter Karte m im eindimensionalen Fall. a) Der Filterzustand vor der Integration der aktuellen Messung. b) Die Messung in der aktuellen Iteration und die dadurch angepassten Gewichte. c) Der Filterzustand nach dem Resampling und Anwenden des Bewegungsmodells $p(x_t|x_{t-1}, u_t)$. d) Erneute Messung und die dadurch angepassten Gewichte. e) Der Filterzustand nach erneutem Resampling und Anwenden des Bewegungsmodells $p(x_{t+1}|x_t, u_{t+1})$.

```

1:   Algorithmus simple_occupancy_grid_mapping( $\mathbf{m}_{t-1}, \mathbf{x}_t, \mathbf{z}_t$ ) :
2:      $passedCells = \text{calculatePassedCells}(\mathbf{m}_{t-1}, \mathbf{x}_t, \mathbf{z}_t)$ 
3:     for every  $m_i$  in  $passedCells$  do
4:        $m_i.visited += 1$ 
5:     endfor
6:     if ( $|\mathbf{z}_t| < maxrange$ )
7:        $passedCells[lastindex].busy += 1$ 
8:     endif
9:     return  $\mathbf{m}_t$ 

```

TABELLE 4.3: Das in dieser Arbeit verwendete Verfahren zur Kartengenerierung bei bekannter Position

4.3 Kartengenerierung bei bekannter Trajektorie

Rasterkarten diskretisieren ihre Umgebung in Zellen gleicher Größe. Eine Zelle enthält dabei die Wahrscheinlichkeit, dass diese belegt und somit für den Roboter unpassierbar ist. Eine genaue Integration der aktuellen Messung ist deshalb entscheidend für eine akkurate Arbeitsweise eines autonomen Roboters. Dies beinhaltet im Allgemeinen die Ermittlung des inversen Sensormodells $p(\mathbf{m}|\mathbf{z}, \mathbf{x})$ (invers zu $p(\mathbf{z}|\mathbf{m}, \mathbf{x})$), welches z. B. mittels Neuronaler Netze erlernt werden kann [BAGNELL et al. 2000]. Die vorliegende Arbeit stützt sich auf einen mit einem Laserscanner ausgestatteten Roboter (siehe Anhang A.2). Der Laser als Sensor besitzt eine hohe Genauigkeit in der Messung und eine geringe Messbreite innerhalb eines Laserstrahls. Dies wird in Abbildung 4.5 durch einen Vergleich mit einem Ultraschallsensor verdeutlicht. Während der Ultraschallsensor mit einer Messung einen größeren Bereich abdeckt (und dadurch jedes Hindernis im Bereich als Distanzmessung in Frage kommt), ist der Laserstrahl durch seine geringe Messbreite äußerst genau. Deshalb kann für die Kartengenerierung ein vereinfachtes Verfahren verwendet werden, welches sich bereits in vielen Anwendungen bewährt hat (siehe [THRUN et al. 1998]). Die Wahrscheinlichkeit, dass eine Zelle m_i (mit Karte $\mathbf{m} := \bigcup_i m_i$) belegt ist, ist proportional zu ihren Sichtungen durch den Sensor als belegt oder frei. Somit gilt

$$p(m_i) = \frac{\#Sichtungen_belegt}{\#Sichtungen_gesamt}. \quad (4.1)$$

Das Verfahren wird in Tabelle 4.3 beschrieben. Zuerst werden für jeden Laserstrahl die Zellen ermittelt, die von diesem passiert werden, und anschließend als „besucht“ markiert. Falls die Messung keine Maximalmessung war, also ein Hindernis erfasst wurde, wird die durch die Position und Distanz berechnete Zelle zusätzlich als „belegt“ markiert. Abbildung 4.6 veranschaulicht die Integration einer Messung durch das vereinfachte Verfahren anhand eines Beispiels.

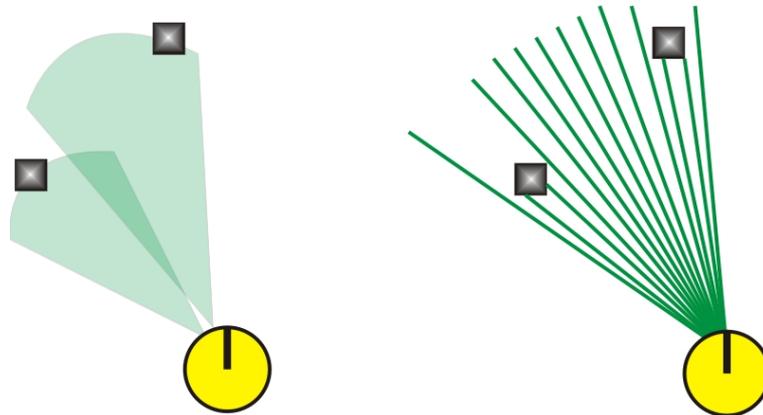


ABBILDUNG 4.5: Vergleich zwischen einem autonomen Roboter ausgestattet mit Ultraschallsensoren (links), und einem Laserscanner (rechts). Die Messbreite der Laserstrahlen ist äußerst gering, wodurch ein vereinfachtes Verfahren zur Kartengenerierung verwendet werden kann. Die schwarzen Blöcke sind jeweils Hindernisse

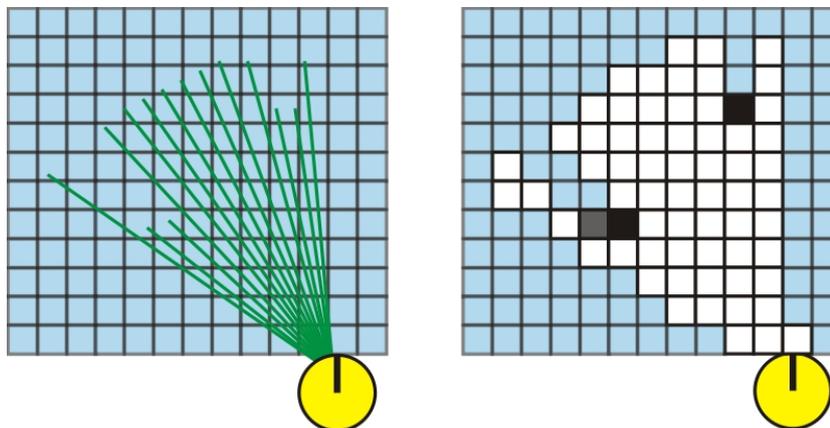


ABBILDUNG 4.6: Aktuelle Messung (links) und die daraus resultierende Rasterkarte (rechts) nach dem *Simple-Occupancy-Grid-Mapping*-Verfahren. Die graue Zelle im rechten Bild besitzt eine Belegungswahrscheinlichkeit von 0,5, da sie von einem Laserstrahl als belegt, von einem anderen als frei gesehen wurde. Zur kontrastreicheren Darstellung sind Zellen mit unbekanntem Inhalt hellblau eingefärbt.

4.4 Rao-Blackwellized-Partikel-Filter und das SLAM-Problem

SLAM steht abkürzend für den englischen Ausdruck *simultaneous localization and mapping* (in der Literatur auch unter *concurrent mapping and localization*, kurz CML, bekannt) und beschreibt das Problem der zeitgleichen Lokalisierung und Kartengenerierung. Wie in den vorangehenden Abschnitten beschrieben, ist für die Bestimmung der Position \mathbf{x}_t die Karte \mathbf{m} notwendig. Umgekehrt wird die Position benötigt, um eine Karte zu erstellen. Abhängig von der verwendeten Kartenstruktur existieren eine Reihe von Verfahren für das SLAM-Problem (siehe z. B. [THRUN et al. 1998, DISSANAYAKE et al. 2001, THRUN et al. 2004b]). So verwenden [THRUN und BÜCKEN 1996] den *Expectation-Maximization*-Algorithmus, um das SLAM-Problem zu lösen. Dabei iteriert das Verfahren zwei Schritte bis zur Konvergenz: den *Expectation*-Schritt, der den aktuell besten Pfad $\mathbf{x}_{1:t}$ basierend auf der Karte \mathbf{m}_t berechnet, und den *Maximization*-Schritt, der die aktuell beste Karte basierend auf dem Pfad errechnet. In dieser Arbeit wird der *Rao-Blackwellized Particle Filter* (RBPF) in Kombination mit Rasterkarten verwendet [DOUCET et al. 2000a]. Das Verfahren baut auf der Lokalisierung mit einem Partikel-Filter auf (siehe Abschnitt 4.2). Die Besonderheit des Verfahrens ist, dass jedes Partikel $\mathbf{x}_{1:t}^{[j]}$ eine eigene Karte $\mathbf{m}_t^{[j]}$ besitzt. Dazu wird ein Partikel innerhalb seiner eigenen Karte mittels des Zustandsübergangsmodells $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t)$ bewegt und anhand der Observationsfunktion $p(\mathbf{z}_t | \mathbf{x}_t^{[j]}, \mathbf{m}_{t-1}^{[j]})$ bewertet. Wird ein Partikel während des Resampling-Schrittes dupliziert, so wird entsprechend auch seine Karte kopiert. Tabelle 4.4 beschreibt den *Rao-Blackwellized-Particle-Filter*-Algorithmus und Abbildung 4.7 visualisiert das Verfahren anhand eines Beispiels mit drei Partikeln.

Das SLAM-Problem lässt sich als Schätzung der Verteilung

$$p(\mathbf{m}_t, \mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \quad (4.2)$$

formulieren [DOUCET et al. 2000a]. Nach [MURPHY 1999] besteht die Grundidee zur Lösung des SLAM-Problems mit dem *Rao-Blackwellized Particle Filter* in der Schätzung der a-posteriori-Verteilung $p(\mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ über mögliche Pfade $\mathbf{x}_{1:t}$. Durch zweimaliges Anwenden der Definition der bedingten Wahrscheinlichkeit lässt sich (4.2) zu

$$p(\mathbf{m}_t, \mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{m}_t | \mathbf{x}_{1:t}, \mathbf{z}_{1:t}) p(\mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}). \quad (4.3)$$

umschreiben. Dabei kann $p(\mathbf{m}_t | \mathbf{x}_{1:t}, \mathbf{z}_{1:t})$ beispielsweise mit dem im letzten Abschnitt vorgestellten *Simple-Occupancy-Grid-Mapping*-Verfahren analytisch berechnet werden. Die Schätzung der Verteilung über alle möglichen Pfade $p(\mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ wird durch einen Partikel-Filter realisiert (siehe Abschnitt 4.2). Das Gewicht eines Partikels ist analog zu (3.21)-(3.26)

$$w_t^{[j]} = \eta w_{t-1}^{[j]} p(\mathbf{z}_t | \mathbf{x}_t^{[j]}, \mathbf{m}_{t-1}^{[j]}). \quad (4.4)$$

```

1:   Algorithmus rbpf_slam( $\chi_{t-1}, \mathbf{z}_t, \mathbf{u}_t$ ) :
2:    $\bar{\chi}_t = \chi_t = \emptyset$ 
3:   for  $i = 1$  to  $M$  do
4:      $\mathbf{x}_t^{[i]} = \text{sample\_motion\_model}(\mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t)$ 
5:      $w_t^{[i]} = w_{t-1}^{[i]} \cdot \text{measurement\_model}(\mathbf{z}_t, \mathbf{x}_t^{[i]}, \mathbf{m}_{t-1}^{[i]})$ 
6:      $\mathbf{m}_t^{[i]} = \text{simple\_occupancy\_grid\_mapping}(\mathbf{m}_{t-1}^{[i]}, \mathbf{x}_t^{[i]}, \mathbf{z}_t)$ 
7:      $\bar{\chi}_t = \bar{\chi}_t \cup \langle \mathbf{x}_t^{[i]}, \mathbf{m}_t^{[i]}, w_t^{[i]} \rangle$ 
8:   endfor
9:   normalizeWeights( $\bar{\chi}_t$ )
10:   $\chi_t = \text{resampling}(\bar{\chi}_t)$ 
11:  return  $\chi_t$ 

```

TABELLE 4.4: Der Rao-Blackwellized Particle Filter in Kombination mit Occupancy Grid Maps.

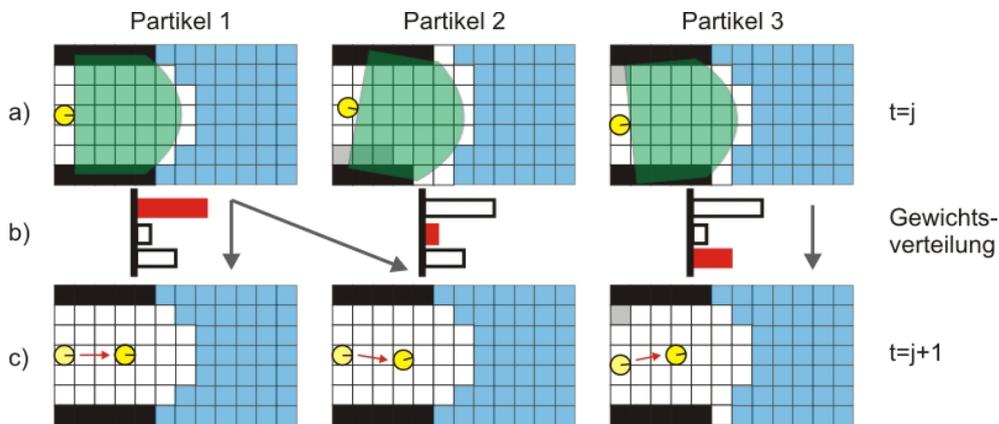


ABBILDUNG 4.7: Der Rao-Blackwellized Particle Filter anhand eines Beispiels mit drei Partikeln. a) Die Partikel nach Integration der aktuellen Messung und b) ihre zugehörigen Gewichte (rot). c) Der Filterzustand nach Resampling (das erste Partikel wurde inklusive Karte dupliziert) und Anwenden des Zustandsübergangmodells $p(\mathbf{x}_t^{[i]} | \mathbf{x}_{t-1}^{[i]}, \mathbf{u}_t)$. Zur kontrastreicheren Darstellung ist die Lasermessung als Fläche und die alte Position hellgelb eingefärbt.

5 Entwickelte Erweiterungen zum RBPf

Für ein wirklich autonom agierendes System ist es von entscheidender Bedeutung, dass es die Fähigkeit besitzt, zur Laufzeit eine konsistente Karte der Arbeitsumgebung zu erstellen. In den letzten Kapiteln wurden dazu die Grundlagen der sequentiellen Zustandsschätzung, Lokalisierung und Kartengenerierung erläutert und zuletzt ein SLAM-Verfahren beschrieben, auf dem diese Arbeit aufbaut: der *Rao-Blackwellized Particle Filter*.

Die Probleme bei der Konstruktion einer Karte nach dem RBPf-Verfahren sind die Folgenden. Theoretisch sind unendlich viele Partikel nötig, um eine Konvergenz der Karte bei den existierenden Unsicherheiten zu gewährleisten. Hoher Speicherplatzbedarf und der Wunsch nach einem Verfahren, welches in Echtzeit eingesetzt werden kann, führen dazu, dass in der Praxis nur eine geringe Anzahl von Partikeln verwendet werden kann. Während Rauschen und Störeinflüsse bei unendlich vielen Partikeln kompensiert werden, können sie bei Verwendung von einer geringen Anzahl an Partikeln zu schwerwiegenden Folgen führen. Aufgrund des zufälligen Ziehens eines Nachfolgezustands $\mathbf{x}_t \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ werden während der fortlaufenden Integration von Messungen diese zwangsläufig teils an falschen bzw. ungünstigen Positionen integriert. Schon eine geringe Abweichung der geschätzten von der wahren Position kann zu einem „Verschmieren“ (engl. *blur*) der Karte führen (siehe Abbildung 5.1). Im fortlaufenden Betrieb führt dies automatisch zu einer erhöhten Unsicherheit bezüglich einzelner Zellen innerhalb der Karte. Diese Unsicherheit kann entscheidenden Einfluss auf die akkurate Arbeitsweise eines Roboters haben, da die Lokalisierung in unsicheren Karten deutlich erschwert wird. Dies wird durch Abbildung 5.2 dargestellt. Durch die unpräzisen Wände sind die Positionen des Roboters in a) und b) gleich wahrscheinlich. Der Verlust der eigenen Positionsinformation kann schließlich zu einer Divergenz der Karte führen. In diesem Fall wäre die Karte für einen Arbeitseinsatz

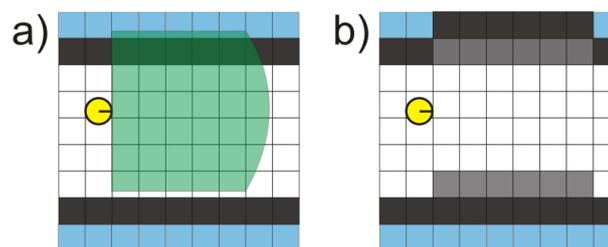


ABBILDUNG 5.1: a) Geringe Abweichungen in der geschätzten Position führen durch Integration der aktuellen Messung zu b) einer erhöhten Unsicherheit innerhalb der Karte.

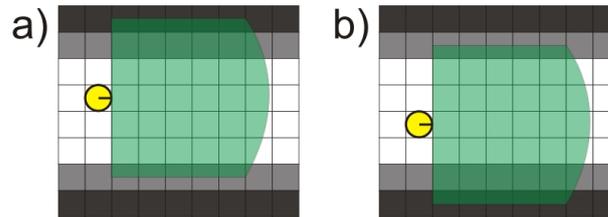


ABBILDUNG 5.2: Durch die unpräzise Karte sind die Roboterpositionen in a) und b) gleich wahrscheinlich, wodurch die Lokalisierung erschwert wird.

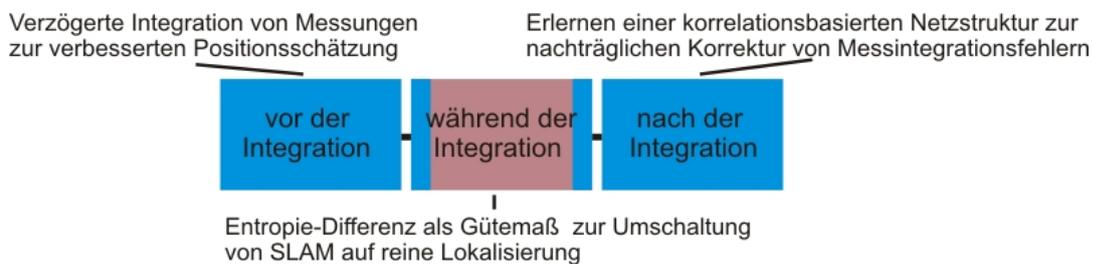


ABBILDUNG 5.3: Die entwickelten Verfahren und ihre zeitliche Arbeitsweise. Blau: Verfahren zur Minimierung der Kartenunsicherheit. Braun (Mitte): Verfahren zur adaptiven Umschaltung von SLAM auf reine Lokalisierung.

ungeeignet. In dieser Arbeit werden drei Erweiterungen des RBPF entwickelt, die den Einsatz auch in schwierigen Umgebungen ermöglichen sollen. Die drei Verfahren werden ergänzend zueinander in verschiedenen zeitlichen Abschnitten des SLAM-Algorithmus eingesetzt:

- vor der Integration der aktuellen Messung,
- während der Integration und
- nach der Integration.

Für jeden dieser Zeitpunkte wurde ein Verfahren entwickelt, welches eine Vergrößerung der Kartenunsicherheit verhindern bzw. minimieren soll. Das Verfahren, welches während der Integration zum Einsatz kommt, leistet neben der Reduzierung der Kartenunsicherheit auch eine automatische Umschaltung zwischen SLAM und reiner Lokalisierung, was für sich allein schon eine wünschenswerte Fähigkeit eines Kartographierungsverfahrens darstellt. Die zeitlichen Unterteilungen sowie die jeweils entwickelten Verfahren sind in Abbildung 5.3 dargestellt. In den nächsten Abschnitten werden die jeweiligen Verfahren unabhängig voneinander beschrieben. Im folgenden Kapitel werden die einzelnen Verfahren und das Gesamtsystem auf ihre Leistungsfähigkeit hin untersucht.

5.1 Verzögerte Integration von Messungen zur verbesserten Positionsschätzung

In diesem Abschnitt wird ein Verfahren vorgestellt, welches Messungen erst verzögert in die Karte integriert, diese aber sofort in der Zustandsübergangsfunktion berücksichtigt und dadurch zu einer akkurateren Positionsschätzung führt. Dadurch kann die Integration von Messungen an falschen oder ungünstigen Positionen minimiert werden, wodurch eine Vergrößerung der Kartenunsicherheit durch eine erhöhte Unsicherheit innerhalb der Zellen verringert wird.

Um die optimale Zustandsübergangsfunktion zu approximieren, verwendet der *Scanmatcher* (z. B. in *SM-FastSLAM*) [STACHNISS 2004] die aktuelle Messung, um die wahrscheinlichste Roboterposition für diese zu bestimmen. Dadurch kann eine genauere Zustandsübergangsfunktion berechnet werden als die, die durch die reine Odometrie bestimmt ist. Das hier vorgestellte Verfahren kann als stichprobenbasierte Alternative zum *Scanmatcher* angesehen werden. Der Vorteil des hier vorgestellten Verfahrens gegenüber dem *Scanmatching*-Ansatz besteht in der Möglichkeit, Multimodalität zu repräsentieren, während das SM-FastSLAM, basierend auf Gradientenabstieg und der aktuellen Messung, nur die wahrscheinlichste Position berücksichtigen kann. In dem im Folgenden beschriebenen Verfahren wird innerhalb jeder Kartenhypothese ein Lokalisierungsfilter gestartet. Die Grundidee ist, sich bezüglich jeder Karte zu lokalisieren und den Folgezustand aus der Wahrscheinlichkeitsverteilung, die durch die Lokalisierung generiert wurde, zu ziehen. Dazu werden Messungen erst verzögert (um k Schritte) in die Karte integriert. Die vorhandenen k Messungen werden für eine Lokalisierung innerhalb der jeweiligen Karte verwendet. Durch Verwenden der zusätzlichen Informationen (Messungen) kann eine genauere Schätzung der Zustandsübergangsfunktion berechnet werden. Formal gesehen ist die Idee, basierend auf \mathbf{x}_{t-1} , den vorhandenen Messungen $\mathbf{z}_{t:t+k}$ und der Odometrie $\mathbf{u}_{t:t+k}$ bis zum Zeitpunkt $t+k$ eine genauere Schätzung der Position \mathbf{x}_t zum Zeitpunkt t zu erreichen. Dazu wird im RBPF-Verfahren, welches durch einen Partikel-Filter zum Zeitpunkt $t-1$ mit den Partikeln $\langle \mathbf{x}_{1:t-1}^{[j]}, \mathbf{m}_{t-1}^{[j]}, w_{t-1}^{[j]} \rangle$ repräsentiert wird, eine Lokalisierung im Intervall $[t, t+k]$ mit einem weiteren Partikel-Filter (im Folgenden „Lokalisierungs-Partikel-Filter“ genannt), basierend auf der Karte $\mathbf{m}_{t-1}^{[j]}$, vollzogen (siehe Abschnitt 4.2). Dazu werden die Lokalisierungs-Partikel i des SLAM-Partikels j innerhalb des Lokalisierungs-Partikel-Filters gemäß der Zustandsübergangsfunktion $p(\mathbf{x}_{[i]t}^{[j]} | \mathbf{x}_{[i]t-1}^{[j]}, \mathbf{u}_t)$ bewegt und entsprechend durch $p(\mathbf{z}_t | \mathbf{x}_{[i]t}^{[j]}, \mathbf{m}_{t-1}^{[j]})$ bewertet. Anschließend wird die daraus resultierende Wahrscheinlichkeitsverteilung über mögliche Positionen $\mathbf{x}_{[i]t+k}^{[j]}$, repräsentiert durch die Gewichte der Lokalisierungs-Partikel $w_{[i]t+k}^{[j]}$, an die entsprechenden (Start-)Positionen $\mathbf{x}_{[i]t}^{[j]}$ propagiert. Dadurch gilt:

$$\mathbf{x}_{[i]t}^{[j]} \sim w_{[i]t+k}^{[j]}. \quad (5.1)$$

Die Menge der gewichteten Partikel $\langle \mathbf{x}_{[i]t}^{[j]}, w_{[i]t+k}^{[j]} \rangle$ approximiert die Zustandsübergangsverteilung $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{z}_{t:t+k}, \mathbf{u}_{t:t+k}, \mathbf{m}_{t-1}^{[j]})$. Abbildung 5.4 und 5.5 skizzieren das Verfahren.

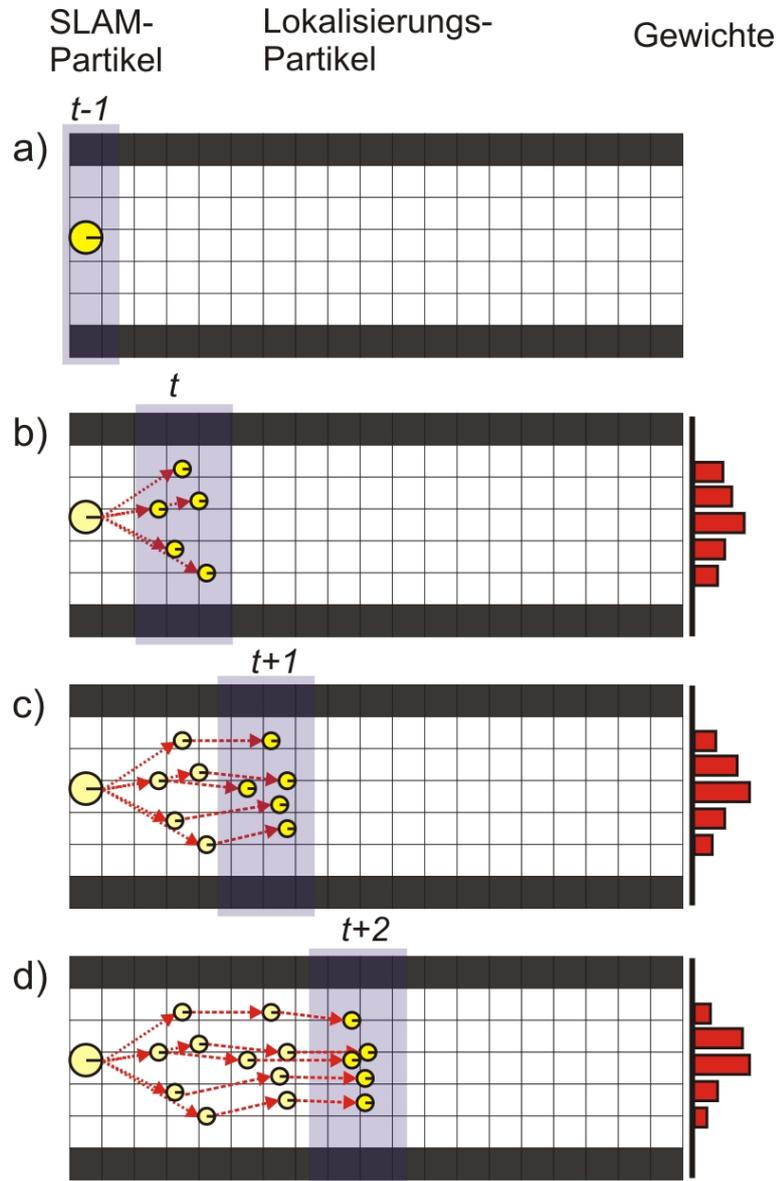


ABBILDUNG 5.4: Das Verfahren zur verzögerten Integration anhand eines Beispiels mit $k = 3$. a) Ein SLAM-Partikel j beschrieben durch seine Position und Karte zum Zeitpunkt $t - 1$. b) Links: ein zweiter Partikel-Filter, zur Lokalisierung bei gegebener Karte $\mathbf{m}_{t-1}^{[j]}$, wird instanziiert. Rechts: die Gewichte der Lokalisierungs-Partikel i zum Zeitpunkt t ($= w_{[i]t}^{[j]}$). c) Lokalisierung durch den Partikel-Filter zum Zeitpunkt $t + 1$ und d) zum Zeitpunkt $t + 2$. Innerhalb des Lokalisierers werden die Partikel gemäß der Zustandsübergangsfunktion $p(\mathbf{x}_{[i]t}^{[j]} | \mathbf{x}_{[i]t-1}^{[j]}, \mathbf{u}_t)$ bewegt und gemäß $p(\mathbf{z}_t | \mathbf{x}_{[i]t}^{[j]}, \mathbf{m}_{t-1}^{[j]})$ bewertet. Zur kontrastreicheren Darstellung ist der aktuelle Zeitpunkt farblich unterlegt. Zudem sind die Lokalisierungs-Partikel im Vergleich zum SLAM-Partikel kleiner dargestellt.

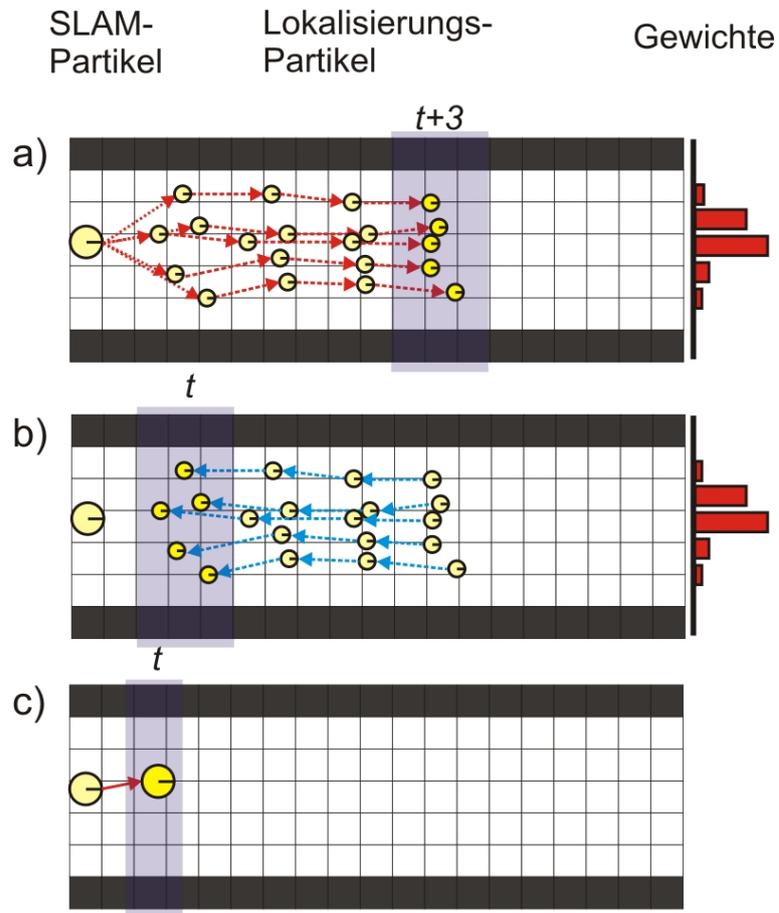


ABBILDUNG 5.5: Fortsetzung des Beispiels aus Abbildung 5.4. a) Die Lokalisierungs-Partikel zum Zeitpunkt $t + 3$ und rechts die zugehörigen normalisierten Gewichte $w_{[i]t+k}^{[j]}$. b) Die Gewichte zum Zeitpunkt $t + k$ werden zu den entsprechenden Startpartikeln propagiert. Rechts: die daraus resultierende Gewichtsverteilung. c) Der Folgezustand des SLAM-Partikels wird entsprechend der neuen Zustandübergangsfunktion gezogen.

Formal lässt sich das Verfahren wie folgt beschreiben: Der *Rao-Blackwellized Particle Filter* (siehe Abschnitt 4.4) ist ein Verfahren zur sequentiellen Schätzung der Verteilung

$$p(\mathbf{m}_t, \mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \quad (5.2)$$

mit

$$p(\mathbf{m}_t, \mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{m}_t | \mathbf{x}_{1:t}, \mathbf{z}_{1:t}) p(\mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t}). \quad (5.3)$$

Dabei wird die Verteilung $p(\mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ durch einen Partikel-Filter, wie in Tabelle 4.4 beschrieben, repräsentiert. Das in diesem Abschnitt beschriebene Verfahren ersetzt die dort verwendete Übergangsfunktion $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t)$ durch $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{u}_{t+1:t+k}, \mathbf{m}_{t-1}^{[j]})$, welches für die folgende Herleitung der Berechnung von $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{u}_{t+1:t+k}, \mathbf{m}_{t-1}^{[j]})$ abkürzend ohne $\mathbf{m}_{t-1}^{[j]}$ als $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{u}_{t+1:t+k})$ beschrieben wird. Mit dem Satz von der totalen Wahrscheinlichkeit lässt sich $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k})$ umformen zu

$$\int p(\mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k}) d\mathbf{x}_{t+1:t+k}. \quad (5.4)$$

Weiterhin gilt mit der Definition der bedingten Wahrscheinlichkeit ($p(A|B) = p(A, B)p(B)^{-1}$)

$$\int \frac{p(\mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k})}{\underbrace{p(\mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k})}_{=: \phi^{-1}}} d\mathbf{x}_{t+1:t+k} \quad (5.5)$$

$$= \int p(\mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k}) \phi d\mathbf{x}_{t+1:t+k}. \quad (5.6)$$

Anwenden der Definition der bedingten Wahrscheinlichkeit auf \mathbf{z}_{t+k} ergibt

$$\int p(\mathbf{z}_{t+k} | \mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k-1}) \cdot p(\mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k-1}) \phi d\mathbf{x}_{t+1:t+k}. \quad (5.7)$$

Im Folgenden wird vorausgesetzt, dass keine Messung \mathbf{z}_{t+i} (mit $i = 0 \dots k$) bezüglich $\mathbf{m}_{t-1}^{[j]}$ unbekanntes Gebiet erfasst. Dies wird in der Praxis durch eine reduzierte Reichweite des Laserscanners während der Lokalisierung erreicht. Dann ist \mathbf{z}_{t+k} , bei fester Karte $\mathbf{m}_{t-1}^{[j]}$ und gegebenem \mathbf{x}_{t+k} , unabhängig von allen anderen Variablen (Markow). Dadurch lässt sich (5.7) zu

$$\int p(\mathbf{z}_{t+k} | \mathbf{x}_{t+k}) p(\mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{z}_{t:t+k-1}) \phi d\mathbf{x}_{t+1:t+k} \quad (5.8)$$

vereinfachen. Wiederholtes Anwenden der Definition der bedingten Wahrscheinlichkeit und der Markow-Annahme für $\mathbf{z}_{t+k-1} \dots \mathbf{z}_t$ führt zu

$$\int p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) \prod_{m=t+1}^{t+k} p(\mathbf{z}_m | \mathbf{x}_m) p(\mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k}, \mathbf{u}_t, \mathbf{z}_{t:t+k}) \phi d\mathbf{x}_{t+1:t+k}. \quad (5.9)$$

Weiterhin lässt sich (5.9) durch Anwenden der Definition der bedingten Wahrscheinlichkeit auf \mathbf{x}_{t+k} zu

$$\int p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) \prod_{m=t+1}^{t+k} p(\mathbf{z}_m | \mathbf{x}_m) p(\mathbf{x}_{t+k} | \mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k-1}, \mathbf{u}_{t:t+k}) \cdot p(\mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k-1}, \mathbf{u}_{t:t+k}) \phi \, d\mathbf{x}_{t+1:t+k} \quad (5.10)$$

umschreiben. Unter der Markow-Annahme ist \mathbf{x}_{t+k} , gegeben \mathbf{x}_{t+k-1} und \mathbf{u}_{t+k} , unabhängig von allen anderen Variablen. Dadurch kann (5.10) vereinfacht werden zu

$$\int p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) \prod_{m=t+1}^{t+k} p(\mathbf{z}_m | \mathbf{x}_m) p(\mathbf{x}_{t+k} | \mathbf{x}_{t+k-1}, \mathbf{u}_{t+k}) \cdot p(\mathbf{x}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{x}_{t+1:t+k-1}, \mathbf{u}_{t:t+k}) \phi \, d\mathbf{x}_{t+1:t+k}. \quad (5.11)$$

Erneutes Anwenden der Definition der bedingten Wahrscheinlichkeit und der Markow-Annahme für $\mathbf{x}_{t+k-1} \dots \mathbf{x}_{t+1}, \mathbf{x}_t^{[j]}$ führt zu

$$\int p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) \prod_{m=t+1}^{t+k} p(\mathbf{z}_m | \mathbf{x}_m) p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t) p(\mathbf{x}_{t+1} | \mathbf{x}_t^{[j]}, \mathbf{u}_{t+1}) \cdot \prod_{m=t+2}^{t+k} p(\mathbf{x}_m | \mathbf{x}_{m-1}, \mathbf{u}_m) p(\mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k}) \phi \, d\mathbf{x}_{t+1:t+k}, \quad (5.12)$$

mit

$$p(\mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k}) \cdot \phi = \frac{p(\mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k})}{p(\mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k})} = p(\mathbf{z}_{t:t+k} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k})^{-1} =: \kappa^{[j]}. \quad (5.13)$$

Einsetzen von (5.13) in (5.12) führt zu

$$\begin{aligned} & p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k}) \\ &= \kappa^{[j]} \int \underbrace{p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t) p(\mathbf{x}_{t+1} | \mathbf{x}_t^{[j]}, \mathbf{u}_{t+1}) \prod_{m=t+2}^{t+k} p(\mathbf{x}_m | \mathbf{x}_{m-1}, \mathbf{u}_m)}_{\text{Zustandsübergänge in } [t, t+k]} \cdot \\ & \quad \underbrace{p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) \prod_{m=t+1}^{t+k} p(\mathbf{z}_m | \mathbf{x}_m)}_{\text{Observationen in } [t, t+k]} \, d\mathbf{x}_{t+1:t+k}. \end{aligned} \quad (5.14)$$

Schätzung von (5.14) durch den (Lokalisierungs-)Partikel-Filter mit $\mathbf{x}_{[i]t-1}^{[j]} = \mathbf{x}_{t-1}^{[j]}$ und Hinzunahme von $\kappa^{[j]}$ in die Normalisierungskonstante $\eta_{t+k}^{[j]}$ ergibt schließlich die Repräsentation von $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k})$ durch $\langle \mathbf{x}_{[i]t}^{[j]}, w_{[i]t+k}^{[j]} \rangle$ mit

$$w_{[i]t+k}^{[j]} \stackrel{(5.14)}{=} \eta_{t+k}^{[j]} \prod_{m=t}^{t+k} p(\mathbf{x}_{[i]m}^{[j]} | \mathbf{x}_{[i]m-1}^{[j]}, \mathbf{u}_m) p(\mathbf{z}_m | \mathbf{x}_{[i]m}^{[j]}). \quad (5.15)$$

Dabei wurde aus Gründen der Übersichtlichkeit angenommen, dass innerhalb des Lokalisierungs-Partikel-Filters kein Resampling stattfindet. Ansonsten müssten in Gleichung (5.15) die Familienhierarchien der Partikel betrachtet werden. Dies führt beispielsweise zu $p(\mathbf{x}_{[ii]t+1}^{[j]} | \mathbf{x}_{[i]t}^{[j]})$ mit $i = \text{parent}(ii)$. Dies ändert jedoch nichts an der prinzipiellen Herleitung und wurde in der praktischen Realisierung des Filters berücksichtigt.

Wie in Abschnitten 3.5 und 4.4 beschrieben, ist das Gewicht des SLAM-Partikels j durch

$$w_t^{[j]} = \frac{p(\mathbf{x}_{1:t}^{[j]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}{\pi(\mathbf{x}_{1:t}^{[j]} | \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k})} \quad (5.16)$$

mit der wahren (unbekannten) Zustandsübergangsfunktion $p(\mathbf{x}_{1:t}^{[j]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ und der verwendeten $\pi(\mathbf{x}_{1:t}^{[j]} | \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k})$ bestimmt. Analog zu (3.22) wird für die in diesem Abschnitt erläuterte Zustandsübergangsfunktion die Annahme

$$\pi(\mathbf{x}_{1:t}^{[j]} | \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k}) = \pi(\mathbf{x}_t^{[j]} | \mathbf{x}_{1:t-1}^{[j]}, \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k}) \pi(\mathbf{x}_{1:t-1}^{[j]} | \mathbf{z}_{1:t+k-1}, \mathbf{u}_{1:t+k-1}) \quad (5.17)$$

vorausgesetzt. Äquivalente Umformungen wie in (3.21) bis (3.24) führen zu

$$\begin{aligned} w_t^{[j]} &= \frac{p(\mathbf{x}_{1:t}^{[j]} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}{\pi(\mathbf{x}_{1:t}^{[j]} | \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k})} \\ &= \eta \frac{p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t)}{\pi(\mathbf{x}_t^{[j]} | \mathbf{x}_{1:t-1}^{[j]}, \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k})} \frac{p(\mathbf{x}_{1:t-1}^{[j]} | \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})}{\pi(\mathbf{x}_{1:t-1}^{[j]} | \mathbf{z}_{1:t+k-1}, \mathbf{u}_{1:t+k-1})} \end{aligned} \quad (5.18)$$

$$= \eta w_{t-1}^{[j]} \frac{p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t)}{\pi(\mathbf{x}_t^{[j]} | \mathbf{x}_{1:t-1}^{[j]}, \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k})}. \quad (5.19)$$

Dabei kann $\pi(\mathbf{x}_t^{[j]} | \mathbf{x}_{1:t-1}^{[j]}, \mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k})$, analog zu (5.4) bis (5.14) mit

$$\kappa^{[j]} = p(\mathbf{z}_{t:t+k} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{1:t+k}, \mathbf{z}_{1:t-1})^{-1} \quad (5.20)$$

umgeformt werden. Mit einer konkreten Realisierung des Nachfolgezustandes $\mathbf{x}_{[i]t}^{[j]}$ und unter Verwendung des (Lokalisierungs-)Partikel-Filters gilt mit (5.15) und (5.20)

$$w_t^{[j]} = \eta w_{t-1}^{[j]} \frac{\eta_t^{[j]} p(\mathbf{z}_t | \mathbf{x}_{[i]t}^{[j]}) p(\mathbf{x}_{[i]t}^{[j]} | \mathbf{x}_{[i]t-1}^{[j]}, \mathbf{u}_t)}{\eta_{t+k}^{[j]} \prod_{m=t}^{t+k} p(\mathbf{x}_{[i]m}^{[j]} | \mathbf{x}_{[i]m-1}^{[j]}, \mathbf{u}_m) p(\mathbf{z}_m | \mathbf{x}_{[i]m}^{[j]})} \quad (5.21)$$

$$\stackrel{(5.15)}{=} \eta w_{t-1}^{[j]} \frac{\eta_t^{[j]} p(\mathbf{z}_t | \mathbf{x}_{[i]t}^{[j]}) p(\mathbf{x}_{[i]t}^{[j]} | \mathbf{x}_{[i]t-1}^{[j]}, \mathbf{u}_t)}{w_{[i]t:t+k}^{[j]}} \quad (5.22)$$

$$\stackrel{(5.15) \text{ mit } k=0}{=} \eta w_{t-1}^{[j]} \frac{w_{[i]t}^{[j]}}{w_{[i]t:t+k}^{[j]}}. \quad (5.23)$$

Dabei ist zu beachten, dass durch die Verwendung eines Partikel-Filters

$$\sum_i w_{[i]t:t+k}^{[j]} = 1 \quad (5.24)$$

gilt. Dadurch wird das in Gleichung (5.20) erhaltene und zu Gleichung (5.13) unterschiedliche $\kappa_{[i]t}^{[j]}$ durch eine entsprechende Normalisierungskonstante korrigiert.

Anschaulich betrachtet wird durch Gleichung (5.23) die Integration der zusätzlichen Informationen im Intervall $[t+1, t+k]$ rückgängig gemacht. Dadurch wird verhindert, dass Messungen mehrfach integriert werden. Das hier vorgestellte Verfahren wirkt also nur auf die räumliche Verteilung der begrenzten Anzahl der zur Verfügung stehenden Partikel. An der repräsentierten Wahrscheinlichkeitsverteilung ändert sich nichts und die bekannte Herleitung des RBPF-Verfahrens gilt weiterhin. Da die Korrektur durch Gleichung (5.23) *nach* dem Ziehen des Folgezustandes getätigt wird, bleibt die Wahl von \mathbf{x}_t davon unberührt. In Gleichung (5.23) wird zudem deutlich, dass sich die Korrektur einfach implementieren lässt, da nur die Gewichte der Lokalisierungs-Partikel zum Zeitpunkt t und $t+k$ verwendet werden.

5.2 Entropie-Differenz als Gütemaß zur Umschaltung von SLAM auf reine Lokalisierung

Wie bereits im ersten Abschnitt beschrieben, kann die Integration einer Messung an einer ungünstigen Position zu einer Verschlechterung der Karte und dadurch auch zu einer erschwerten Lokalisierung in ihr führen. Im letzten Abschnitt wurde dazu ein Verfahren vorgestellt, welches eine bessere Zustandsübergangsfunktion berechnet, als die, die nur durch die Odometrie bestimmt ist. Die dadurch deutlich gesteigerte Kartenqualität darf jedoch nicht darüber hinwegtäuschen, dass die verwendeten Modelle nur für ein gewisses Maß an Störungen und Rauschen ausgelegt sind. Stärkere Störungen können durch die Modelle nicht erfasst werden und im Extremfall zur Divergenz der Karte führen. Solch eine Divergenz ist in Abbildung 5.6 dargestellt. Nach mehreren Umrundungen einer quadratischen Kiste wurde ihre Struktur in der Karte immer unpräziser. Dies führte schließlich zu einer ungenauen Positionsinformation. Dadurch wurden Messungen an falschen Stellen integriert. Daraus folgte, dass die Kiste über die Zeit immer mehr zu einem Kreis wurde. Bei einer Messung der Ränder war anschließend jede Position um die Kiste herum gleich wahrscheinlich. Dies führte letztendlich zu einer Divergenz der Struktur und der Karte. Deshalb ist es sinnvoll, solche Störungen zu erkennen und deren Integration zu verhindern. Dazu wird in diesem Abschnitt ein Verfahren vorgestellt, welches basierend auf dem Informationsgewinn der aktuellen Messung entscheidet, ob diese integriert wird oder nicht.

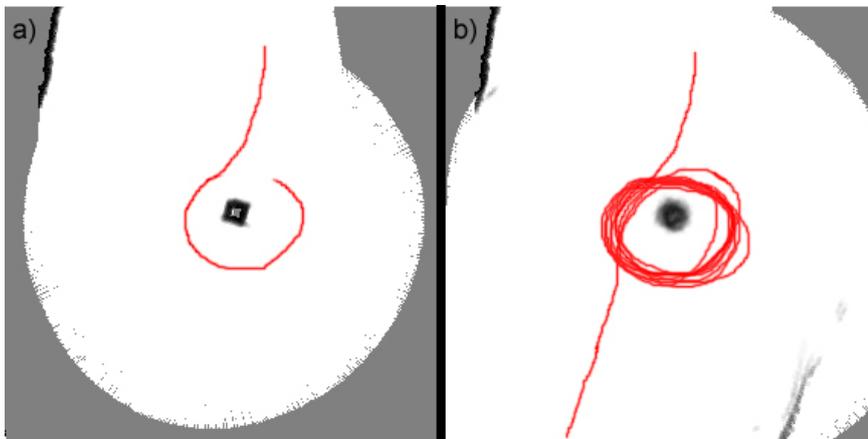


ABBILDUNG 5.6: Beispiel für den Verlust einer Struktur. Durch die Integration von Messungen an falschen bzw. ungünstigen Positionen wurde die Kiste über die Zeit immer mehr zu einem Kreis. Dies führte schließlich zu einer Divergenz der Karte. Eine Lokalisierung an der Kiste war nicht mehr möglich. Deshalb schien der Roboter letztendlich in entgegengesetzter Richtung herauszufahren, obwohl er real in die selbe Richtung gefahren ist, aus der er gekommen war.

Die Entropie H einer Wahrscheinlichkeitsverteilung $p(x)$ ist definiert durch

$$H_p(x) = E[-\log_2 p(x)]. \quad (5.25)$$

Bei einer diskreten Zufallsvariablen x wird (5.25) zu

$$H_p(x) = - \sum_x p(x) \log_2 p(x). \quad (5.26)$$

Die Entropie $H_p(x)$ kann als statistisches Maß für die Anzahl an Bits interpretiert werden, die für die Übertragung der Zufallsvariablen x im Mittel notwendig sind [MITCHELL 1997]. Im Kontext der Kartengenerierung kann die Entropie als Maß für die Information der Karte angesehen werden. [STACHNISS et al. 2004b, STACHNISS et al. 2005] verwenden die Entropie für die aktive Exploration im SLAM-Kontext. Dazu wird der erwartete Informationsgewinn für mögliche Zielpositionen berechnet und basierend darauf die nächste Aktion gewählt. Basierend auf den Zellen, die durch die aktuelle Messung erfasst werden, berechnet das hier vorgestellte Verfahren die Entropie-Differenz, die durch die Integration der Messung zustandekommen würde. Die Entropie-Differenz ist dabei als

$$ED[\mathbf{m}_t, \mathbf{m}_{t-1}] := H_p(\mathbf{m}_t) - H_p(\mathbf{m}_{t-1}) \quad (5.27)$$

mit

$$H_p(\mathbf{m}) = - \sum_{m_i} (p(m_i = \text{frei}) \log_2 p(m_i = \text{frei}) + p(m_i = \text{belegt}) \log_2 p(m_i = \text{belegt})) \quad (5.28)$$

definiert. Ist diese Differenz unterhalb einer Schwelle τ , so wird die Messung integriert, ansonsten verworfen. Die Kernidee des Verfahrens ist, dass sich Unsicherheiten, anschaulich „Verschmierungen“ der Karte, und dadurch der Verlust von Struktur, in einer positiven Entropie-Differenz widerspiegeln, während passende Messungen und Exploration als negative (bzw. unveränderte) Differenz auftreten.

Gerade während der Exploration ist die Entropie-Differenz meist stark negativ: unbekannte Zellen besitzen nach Definition eine Belegungswahrscheinlichkeit von 0,5 und dadurch eine maximale Entropie von 1. Nach Integration der Messung durch das *Simple-Occupancy-Grid-Mapping*-Verfahren (siehe Kapitel 4.3) besitzen die Zellen im Extremfall eine Wahrscheinlichkeit von 0 oder 1 und damit die minimale Entropie von 0. Eine aktuelle Messung würde demnach, unabhängig von ihrer Qualität, eine hohe negative Entropie-Differenz besitzen, wenn entsprechend viele unbekannte Zellen erfasst werden. Um die Integration ungünstiger Messungen auch während der Explorationsphase zu minimieren, wird daher die Entropie-Differenz nur auf dem Teil der Messung berechnet, der geschnitten mit der aktuellen Karte für jeden Laserstrahl maximal eine unbekannte Zelle beinhaltet. Abbildung 5.7 verdeutlicht das Verfahren anhand eines Beispiels. In der Praxis führte die Wahl von $\tau = -2$ stets zu guten Ergebnissen. Die praktischen Auswirkungen dieser Vorgehensweise und die Leistungsfähigkeit des Gesamtsystems werden in Kapitel 6 untersucht.

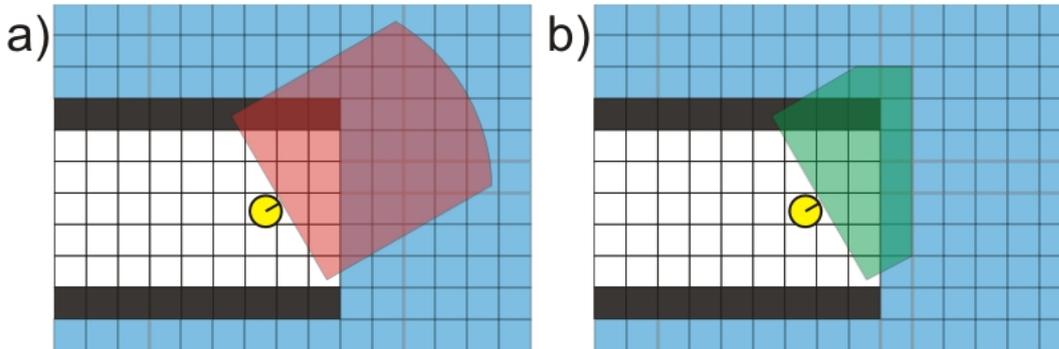


ABBILDUNG 5.7: a) Für die Berechnung der Entropie-Differenz werden nicht alle Zellen, die von der aktuellen Messung erfasst werden, verwendet. b) Stattdessen wird der Schnitt mit der aktuellen Karte und für jeden Laserstrahl maximal eine unbekannte Zelle für die Berechnung betrachtet. Dadurch können Messintegrationsfehler auch während der Explorationsphase verringert werden.

5.3 Erlernen einer korrelationsbasierten Netzstruktur zur nachträglichen Korrektur von Messintegrationsfehlern

In den letzten beiden Abschnitten wurden Verfahren vorgestellt, die die Integration einer Messung an einer ungünstigen oder falschen Position minimieren bzw. verhindern sollen. Das im letzten Abschnitt beschriebene Verfahren (Entscheidung über die Integration einer Messung basierend auf der Entropie-Differenz) ist abhängig von der Wahl des Schwellwertes τ . Eine zu niedrige Wahl führt dazu, dass kaum Messungen integriert werden (und somit keine Karte erstellt wird). Eine zu hohe Wahl erlaubt die Integration von Messungen an falschen Positionen. Bei der Wahl von τ muss deshalb ein Kompromiss zwischen der Integration und der Verwerfung von Messungen getroffen werden. Deshalb kann die Integration einer Messung an einer ungünstigen Position nicht vollständig verhindert werden (siehe Abbildung 5.8a). Daher ist es sinnvoll, ein Verfahren zu entwickeln, das die fehlerhafte Integration von Messungen nachträglich korrigieren kann (indem es diese Integration rückgängig macht), wie in Abbildung 5.8b dargestellt. Dabei ist zu beachten, dass sich fehlerhafte Integrationen insbesondere an Rändern (z. B. Wänden) bemerkbar machen, da dort Zellen evtl. nur ein einziges Mal erfasst werden. Diese fehlerhaften, meist einmaligen Integrationen werden im Folgenden Artefakte genannt. Durch die Integration einer Messung entsteht eine Korrelation zwischen den durch eine Messung erfassten, benachbarten Zellen. Der Grundgedanke in dem hier vorgestellten Verfahren ist, diese Korrelation während der fortlaufenden Kartengenerierung zu erlernen und sie für eine nachträgliche Entfernung von Integrationen für einzelne Zellen zu verwenden. Angenommen, die Zellen m_i und m_j sind direkt benachbart und werden während der Integration eines Laserstrahls als „frei“ aktualisiert. Zu einem späteren Zeitpunkt wird wieder eine Messung aus demselben Winkel gemacht, bei der jedoch nur die erste von beiden Zellen, m_i , erfasst und als „belegt“ markiert

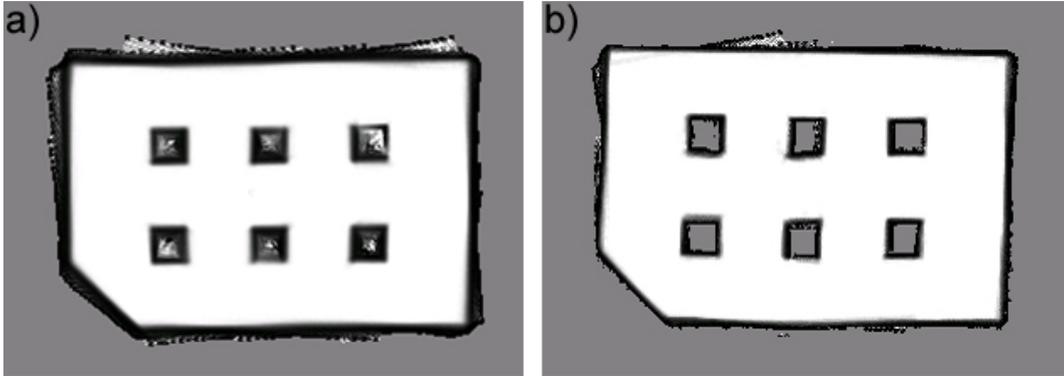


ABBILDUNG 5.8: Für das in Abschnitt 5.2 vorgestellte Verfahren muss durch die Wahl von τ ein Kompromiss zwischen der Integration und dem Verwerfen von Messungen getroffen werden. Deshalb ist eine vollständige Verhinderung von Messintegrationsfehlern nicht möglich (a). b) Das in diesem Abschnitt beschriebene Verfahren ist durch Erlernen einer korrelationsbasierten Netzstruktur in der Lage, fehlerhafte Integrationen nachträglich zu korrigieren. Das verbleibende Artefakt im rechten Bild (oben) konnte nicht korrigiert werden, da nach Integration der Messung, die diesen Fehler generiert hat, keine korrekte Messung aus einem ähnlichen Winkel mehr integriert wurde.

wird. Die vorher zwischen m_i und m_j erlernte Korrelation kann nun dazu verwendet werden, den Laserstrahl entsprechend der erlernten Korrelation zu m_j zu propagieren und somit die frühere fälschliche Integration zu korrigieren. Abbildung 5.9 verdeutlicht das Verfahren anhand eines Beispiels im eindimensionalen Fall.

Formal lässt sich das Verfahren wie folgt beschreiben: Zwei benachbarte Zellen m_i und m_j werden als Zufallsvariablen desselben Experimentes betrachtet. Dadurch ist ihre Korrelation durch

$$\rho(m_i, m_j) = \frac{Cov(m_i, m_j)}{\sqrt{(Var(m_i)Var(m_j))}} \quad (5.29)$$

definiert. Da in der Praxis die Richtung, aus der der Laserstrahl eingefallen ist, von Bedeutung ist, müssen die Zellen und dadurch auch Gleichung (5.29) um die Information des Einfallswinkels θ erweitert werden:

$$\rho(m_i^\theta, m_j^\theta) = \frac{Cov(m_i^\theta, m_j^\theta)}{\sqrt{(Var(m_i^\theta)Var(m_j^\theta))}}. \quad (5.30)$$

Bei jeder Aktualisierung der Zellen m_i und m_j kann nun durch Gleichung (5.30) die Korrelation zwischen den beiden errechnet werden. Dazu wird in der Praxis für jede Zelle eine Acht-Nachbarschaft (siehe Abbildung 5.10) betrachtet und die Korrelation entsprechend ihrem θ -Anteil der jeweiligen Nachbarzellen berechnet. Da aufgrund des *Simple-Occupancy-Grid-Mapping*-Verfahrens (siehe Abschnitt 4.3) die Varianz innerhalb einer Zelle meist 0 ist, wird die

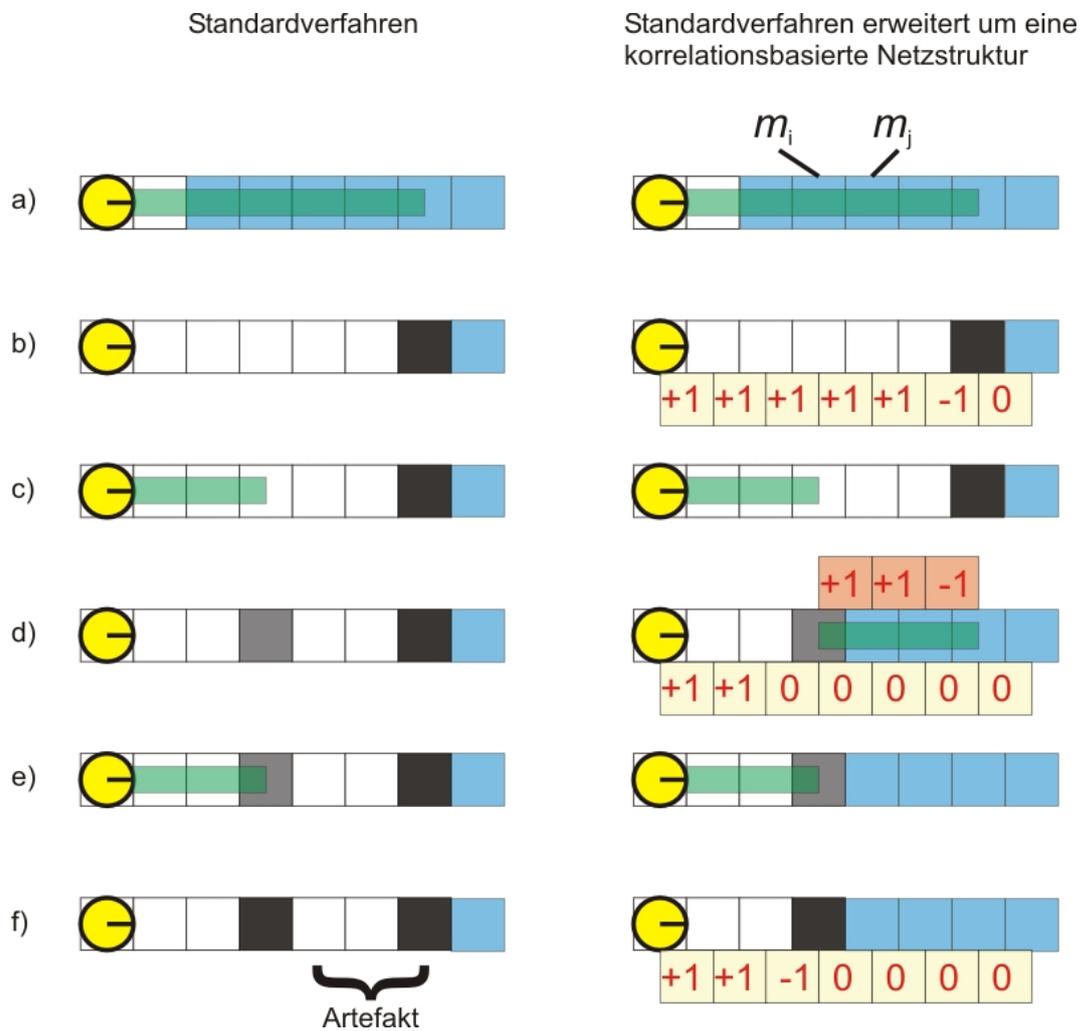


ABBILDUNG 5.9: Beispiel für eine Korrektur und Korrelationserlernung im eindimensionalen Fall. Rechts: Der Rao-Blackwellized Particle Filter erweitert durch eine erlernte Netzstruktur. Links: das Standardverfahren. a) Eine Fälschmessung und b) der Kartenzustand nach deren Integration. Unten: die zusätzlich erlernten Korrelationen zwischen den benachbarten Zellen (für die Richtung links nach rechts). c) Erneute (korrekte) Messung und d) ihre Integration. Durch die vorher erlernte Korrelation zwischen den Nachbarzellen (oben) kann die Messung entsprechend der Korrelation durchpropagiert werden. Aufgrund der Korrelation von $+1$ zwischen m_i und m_j bzw. m_j und m_{j+1} wird daher eine Evidenz „frei“ rückgängig gemacht. Dies ist analog dazu, dass die Evidenz „belegt“ integriert würde. Da zwischen m_{j+1} und m_{j+2} eine Korrelation von -1 existiert, wird von m_{j+2} eine Evidenz „belegt“ entfernt (was entsprechend der Integration von „frei“ gleichkommt). Daher wird die Integration aus a) für die nach m_i durch die Korrelation erfassten Zellen rückgängig gemacht. Anschließend wird die Korrelationsänderung erlernt (unten). e) Eine weitere (korrekte) Messung und f) der Kartenzustand nach ihrer Integration. Das Standardverfahren (links) besitzt nun ein Artefakt.

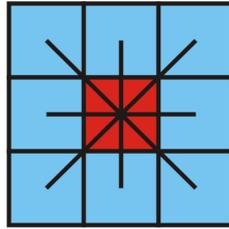


ABBILDUNG 5.10: Eine Zelle (Mitte, rot) und ihre Acht-Nachbarschaft.

Korrelation durch ein Zählmodell approximiert. Werden zwei benachbarte Zellen mit derselben Evidenz („frei“ bzw. „belegt“) erfasst, entspricht dies einer Korrelation von $+1$, ansonsten -1 . Durch die Gewichtung mit θ werden entsprechend Korrelationen zwischen -1 und $+1$ generiert. Der absolute Korrelationswert wird anschließend als Wahrscheinlichkeit dafür angesehen, dass der Strahl entsprechend dem Vorzeichen von ρ weiterpropagiert wird. Angenommen, zwischen den oben beschriebenen Zellen m_i und m_j besteht eine Korrelation von $0,6$. Weiterhin werde durch die aktuelle Messung die Zelle m_i als belegt markiert. Dies ist gleichbedeutend damit, dass der Laserstrahl in Zelle m_i endet. Aufgrund der vorher erlernten Korrelation kann die Information der „belegten“ Zelle an m_j weitergereicht werden. In diesem Beispiel wird für die Zelle m_j mit einer Wahrscheinlichkeit von $0,6$ eine Evidenz „frei“ rückgängig gemacht, was gleichbedeutend mit einer Integration der Evidenz „belegt“ ist. Falls, wie bei Artefakten üblich, die Zelle nur einmal erfasst wurde, fällt ihre Belegungswahrscheinlichkeit dementsprechend auf „unbekannt“ zurück.

5.4 Implementierung des Gesamtsystems

Die in diesem Kapitel vorgestellten Erweiterungen zum *Rao-Blackwellized Particle Filter* sind in Tabelle 5.1 beschrieben. Im Detail lässt sich der Algorithmus wie folgt erklären:

Zeilen 4-13 implementieren das Verfahren zur verzögerten Integration, wie in Abschnitt 5.1 beschrieben. Dazu wird für jedes SLAM-Partikel j ein eigener Lokalisierer mit der bisherigen Position initialisiert (Zeile 4). Anschließend werden die Lokalisierer-Partikel entsprechend der Odometrie \mathbf{u}_t bewegt und mit der Messung \mathbf{z}_t bewertet (Zeile 5-6). Der Lokalisierer hat zu diesem Zeitpunkt den Filterzustand wie in Abbildung 5.4b. Um später aus dieser Menge von Partikeln den Folgezustand $\mathbf{x}_t^{[j]}$ ziehen zu können, wird die aktuelle Partikelmenge zwischengespeichert (also ein *Snapshot* erstellt; siehe Zeile 7). Anschließend werden die Lokalisierer-Partikel entsprechend der restlichen Odometrie bewegt und durch die jeweiligen Messungen bewertet (Zeilen 8-11). Der Filterzustand des Lokalisierers j entspricht nun dem, der in Abbildung 5.5a dargestellt ist. Analog zu Gleichung (5.15) wird in Zeile 12 der Folgezustand $\mathbf{x}_t^{[j]}$ gezogen und entsprechend zu Gleichung (5.23) in Zeile 13 bewertet.

In Zeile 14 wird die Messung \mathbf{z}_t , wie beim Standard-RBPF-Verfahren, in die aktuelle Karte integriert. Anschließend (Zeile 15) wird die Entropie-Differenz, die auf der aktuellen Messung, sowie der neuen und der alten Karte basiert, berechnet (siehe Abschnitt 5.2). Liegt die Entropie-Differenz unter einer vorher festgelegten Schranke τ , wird die aktuelle Messung verwendet, um mit Hilfe des Verfahrens zu nachträglichen Korrektur von Integrationsfehlern (Abschnitt 5.3) die Karte gegebenenfalls zu korrigieren und die Änderungen der Korrelationen zu erlernen (Zeilen 17-18). Liegt die Entropie-Differenz oberhalb der Schranke τ , so wird die aktuelle Messung \mathbf{z}_t verworfen und somit die bestehende Karte $\mathbf{m}_{t-1}^{[j]}$ ohne Änderung übernommen.

Zeile 26 berechnet die Anzahl der „effektiven“ Partikel, die nach [LIU 1996] durch

$$N_{\text{eff}} = \frac{1}{\sum_{j=1}^M (w_t^{[j]})^2}$$

definiert ist. Dabei kann N_{eff} als Maß für die Varianz innerhalb der Partikel angesehen werden. Die Wahl von $\frac{M}{2}$ als Schranke ist an [DOUCET et al. 2000b] angelehnt. Abhängig von N_{eff} wird in den Zeilen 27-29 ein Resampling nach der *Low-Variance-Sampling*-Methode vollzogen. Diese adaptive Art des Resampling hat sich bereits in vielen SLAM-Verfahren bewährt (siehe beispielsweise [STACHNISS et al. 2004a]).

```

1: Algorithmus extended_rbpf_slam( $\chi_{t-1}, \mathbf{z}_{t:t+k}, \mathbf{u}_{t:t+k}$ ) :
2:    $\bar{\chi}_t = \chi_t = \emptyset$ 
3:   for  $j = 1$  to  $M$  do
4:     localizer[ $j$ ].initPosition( $\mathbf{x}_{t-1}^{[j]}$ )
5:     localizer[ $j$ ].processScan( $\mathbf{u}_t$ )
6:     localizer[ $j$ ].weightOwnParticles( $\mathbf{z}_t, \mathbf{m}_{t-1}^{[j]}$ )
7:     localizer[ $j$ ].createParticlesSnapshot()
8:     for  $l = t + 1$  to  $t + k$  do
9:       localizer[ $j$ ].processScan( $\mathbf{u}_l$ )
10:      localizer[ $j$ ].weightOwnParticles( $\mathbf{z}_l, \mathbf{m}_{t-1}^{[j]}$ )
11:    endfor
12:     $\mathbf{x}_t^{[j]}, index = \mathbf{localizer}[j].\mathbf{drawPoseFromSnapshotUsingLatestWeights}()$ 
13:     $w_t^{[j]} = w_{t-1}^{[j]} \cdot \mathbf{localizer}[j].\mathbf{getWeightCorrection}(index)$ 
14:     $\bar{\mathbf{m}}_t^{[j]} = \mathbf{simple\_occupancy\_grid\_mapping}(\mathbf{m}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}, \mathbf{z}_t)$ 
15:     $ed = \mathbf{calculateEntropyDifference}(\bar{\mathbf{m}}_t^{[j]}, \mathbf{m}_{t-1}^{[j]}, \mathbf{z}_t)$ 
16:    if  $ed < \tau$  then
17:      propagateScanIfNeeded( $\bar{\mathbf{m}}_t^{[j]}, \mathbf{z}_t$ )
18:      updateCorrelations( $\bar{\mathbf{m}}_t^{[j]}, \mathbf{z}_t$ )
19:       $\mathbf{m}_t^{[j]} = \bar{\mathbf{m}}_t^{[j]}$ 
20:    else
21:       $\mathbf{m}_t^{[j]} = \mathbf{m}_{t-1}^{[j]}$ 
22:    endif
23:     $\bar{\chi}_t = \bar{\chi}_t \cup \langle \mathbf{x}_t^{[j]}, \mathbf{m}_t^{[j]}, w_t^{[j]} \rangle$ 
24:  endfor
25:  normalizeWeights( $\bar{\chi}_t$ )
26:   $N_{eff} = \mathbf{calculateNumberOfEffectiveParticles}(\bar{\chi}_t)$ 
27:  if  $N_{eff} < \frac{M}{2}$  then
28:     $\chi_t = \mathbf{resampling}(\bar{\chi}_t)$ 
29:  else
30:     $\chi_t = \bar{\chi}_t$ 
31:  endif
32:  return  $\chi_t$ 

```

TABELLE 5.1: Der erweiterte Rao-Blackwellized Particle Filter in Kombination mit Occupancy Grid Maps.

6 Evaluierung

In diesem Kapitel werden verschiedene Experimente präsentiert, die mit echten Robotern sowohl im Innen- als auch im Außenbereich durchgeführt wurden. Die Implementierung erfolgte durch Erweiterung von Giorgio Grisettis Implementierung des Standard-RBPF-Algorithmus um die in Kapitel 5 beschriebenen Verfahren. Sie wurden mit einem Pioneer 3-DX Roboter, der mit einem SICK Laserscanner ausgerüstet ist, getestet (siehe Anhang A.1 und A.2).

Um ein Maß für die Qualität der Karte zu erhalten, wurde das Fernsteuerungsmodul für Roboter in der bestehenden CARMEN-Implementierung [CAR] erweitert. Durch diese Erweiterung ist es per Fernsteuerung möglich, Zeitmarkierungen im Datenstrom zu setzen. Dazu wurden auf dem Boden Orientierungspunkte angebracht (z. B. Orientierungspunkt „A“). Jedes Mal, wenn der Roboter im gleichen Winkel einen der Orientierungspunkte passierte, wurde dieser per Fernsteuerung aktiviert und dadurch der Zeitindex markiert. Durch mehrmaliges Anfahren der so markierten Punkte ist es möglich, die zeitliche Stabilität und Konsistenz der geschätzten Karte zu messen. Eine Karte gilt als gut, wenn dieselben Punkte in der echten Umgebung durch möglichst nahe beieinanderliegende Punkte in der Karte geschätzt werden. Dadurch ist es möglich, ein Maß für die Qualität der Verfahren zu entwickeln. Da nach der obigen Verwendungsweise der Roboter zu allen Zeitpunkten, zu denen per Fernsteuerung derselbe Orientierungspunkt aktiviert wurde, an derselben Position sein muss, kann die Abweichung in dieser Position als Fehlermaß angesehen werden. Der Fehler wird dabei als Distanz zum Mittelwert der Orientierungspunkte berechnet. Für zwei Punkte P_1 und P_2 wird dieser mit der in [PFAFF et al. 2006] vorgestellten Fehlerformel

$$Error(P_1, P_2) = \sqrt{(1 - \lambda)[(P_1.x - P_2.x)^2 + (P_1.y - P_2.y)^2] + \lambda(P_1.\theta - P_2.\theta)^2} \quad (6.1)$$

mit $\lambda = 0,5$ berechnet. Anschaulich erfasst dieses Maß die gewichtete euklidische Distanz in x -, y - und θ -Richtung. Alternativ dazu wäre es möglich, eine Grundwahrheit mit dem *scanmatchingbasierten FastSLAM* bei einer Reichweite von beispielsweise 40 Metern zu erstellen. Da sich jedoch besonders im Außenbereich die Karten durch eine (unbekannte) Drehung voneinander unterscheiden, wären Vorverarbeitungsschritte (z. B. mit einem ICP-Verfahren) notwendig, um Karten mit den berechneten Punkten vergleichen zu können. Für die praktische Durchführung der Experimente wurden Arbeitsplatzrechner mit einer Taktfrequenz zwischen 2,4 GHz und 3,4 GHz und einem Hauptspeicher von 2GB verwendet. Die Experimente fanden ausschließlich unter dem Betriebssystem Linux statt.

Die Leistungsfähigkeit wurde quantitativ anhand zweier Experimente untersucht, die in den folgenden Abschnitten präsentiert werden. Davon wurde eines im Innen-, das andere im Außenbereich durchgeführt. In jedem der Experimente wurden am Boden Orientierungspunkte angebracht, um die Güte der erstellten Karten und der Lokalisierung in ihr zu bewerten. In allen präsentierten Experimenten wurde für die Entropie-Differenz ein Schwellwert von $\tau = -2$ gewählt. Der Distanz-Fehler beschreibt den mittleren Abstand (Fehler) eines geschätzten Orientierungspunktes zum Mittelwert aller (gleicher) Orientierungspunkte. Weiterhin sind die mittleren Standardabweichungen der SLAM-Partikel in einer separaten Grafik dargestellt. Nach der quantitativen Auswertung werden die Ergebnisse weiterer Experimente qualitativ diskutiert und die Wirkung der drei einzelnen in dieser Arbeit entwickelten Verfahren dargestellt.

6.1 Experiment innerhalb eines Gebäudes

In diesem Experiment wurden Daten innerhalb eines Gebäudes aufgezeichnet. Der Roboter fuhr entlang eines Korridors hin und her und hatte einen identischen Start- und Endpunkt (siehe Abbildung 6.1). Für alle im Folgenden verglichenen Verfahren wurde die Reichweite des Laserscanners auf fünf Meter reduziert.



ABBILDUNG 6.1: Das verwendete Szenario im Innenbereich.

Die mit dem Standardverfahren bei 350 bzw. 50 Partikeln erstellten Karten sind in Abbildung 6.2 dargestellt. Abbildung 6.3 zeigt die erzeugten Karten mit dem erweiterten Verfahren (siehe Kapitel 5) bei 50 bzw. 10 SLAM-Partikeln mit je 50 Lokalisierungs-Partikeln und $k = 3$. Dabei bezeichnet k die Anzahl der verwendeten zusätzlichen zukünftigen Messungen im Verfahren zur verzögerten Integration (siehe Abschnitt 5.1). In diesen Abbildungen wird deutlich, dass das Standardverfahren schlechtere Karten erzeugt als das durch die in dieser Arbeit entwickelten Algorithmen erweiterte Verfahren bei Verwendung von nur etwa drei Prozent sovielen SLAM-Partikeln. Im Innenbereich, wo viel Struktur vorhanden ist, liefert der *Scanmatcher* optimale Ergebnisse. In der folgenden Auswertung wird deutlich, dass das hier entwickelte System sehr nahe an die Leistung von SM-FastSLAM herankommt. Im Außenbereich (folgendes Experiment), wo weniger Struktur vorhanden ist, werden deutlich bessere Ergebnisse als mit SM-FastSLAM erzielt.

Der identische Start- und Endpunkt wurde für eine Evaluierung der Lokalisierung innerhalb der Karten verwendet. Abbildungen 6.4 und 6.5 zeigen die Fehlerkurven für das Standardverfahren bei 350 beziehungsweise 50 SLAM-Partikeln, das SM-FastSLAM-Verfahren bei 50 bzw. 10 SLAM-Partikeln und das in dieser Arbeit erweiterte Verfahren bei 50 und 10 SLAM- mit je 50 Lokalisierungs-Partikeln beziehungsweise 20 SLAM mit je 100 Lokalisierungs-Partikeln. Um eine Evaluation der einzelnen Verfahren zu ermöglichen, wurde die Testreihe jeweils mit und ohne die korrelationsbasierte Netzstruktur (in den Abbildungen mit KbN abgekürzt) erstellt. Im Vergleich zu dem Standardverfahren ist in den Abbildungen deutlich zu erkennen, dass sich die entwickelten Verfahren positiv auf die Lokalisierung des Roboters auswirken. Bereits das Verfahren, das die Entscheidung für oder gegen die Integration einer Messung trifft, hat einen großen Einfluss auf die generierte Karte, und damit auf die Lokalisierung in ihr. Die Fehlerkurve für die hier entwickelten Verfahren lässt sich dabei wie folgt interpretieren. Mit wachsendem k besteht eine erhöhte Wahrscheinlichkeit dafür, dass Lokalisierungs-Partikel vermehrt zu späteren Zeitpunkten eine gute Position haben. Damit steigt auch die Wahrscheinlichkeit dafür, dass Partikel gezogen werden, die zwar k Schritte weiter eine gute Position haben, deren Anfangsposition (und damit der gezogene Nachfolgezustand) jedoch suboptimal war. Dennoch ist zu erkennen, dass die Fehlerkurve der entwickelten Verfahren unter der des Standardverfahrens liegt, und dass die Leistung von SM-FastSLAM nahezu erreicht wird. Dies ist, wie bereits oben beschrieben, dadurch zu erklären, dass gerade im Innenbereich sehr viele Merkmale vorhanden sind, an denen sich der *Scanmatcher* orientieren kann. Wie zu erwarten, führt die Verwendung von mehr Lokalisierungs-Partikeln zu einem geringeren Fehler.

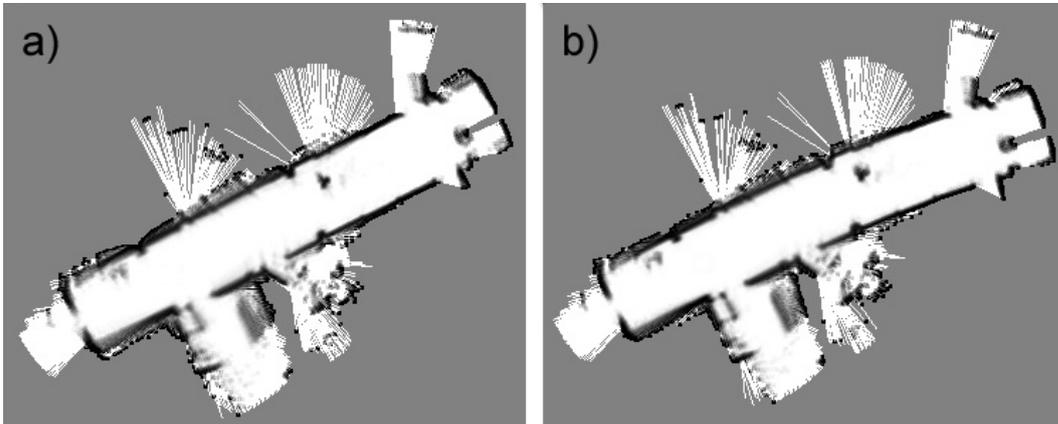


ABBILDUNG 6.2: Mit dem Standardverfahren erzeugte Karten bei a) 350 und b) 50 SLAM-Partikeln. Ungenauigkeiten in der Karte sind deutlich zu erkennen. Die Karte in b) besitzt zusätzlich einen starken Rotationsfehler.

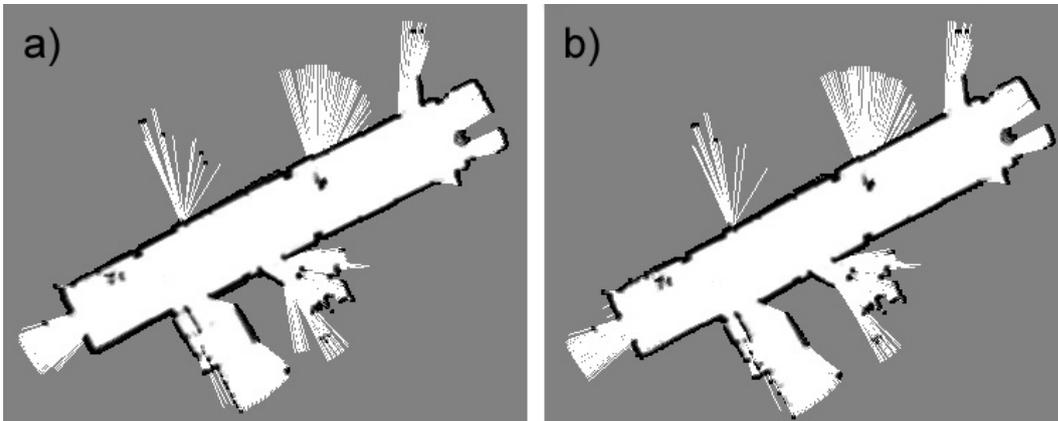


ABBILDUNG 6.3: Mit den in dieser Arbeit entwickelten Verfahren erzeugte Karten mit a) 50 und b) 10 SLAM-Partikeln. Für dieses Verfahren wurden pro SLAM-Partikel je 50 Lokalisierungs-Partikel verwendet. Die zeitliche Verzögerung bei der Integration betrug $k = 3$. Messungen, die zu einer Entropie-Differenz von mehr als $\tau = -2$ führen würden, wurden nicht integriert.

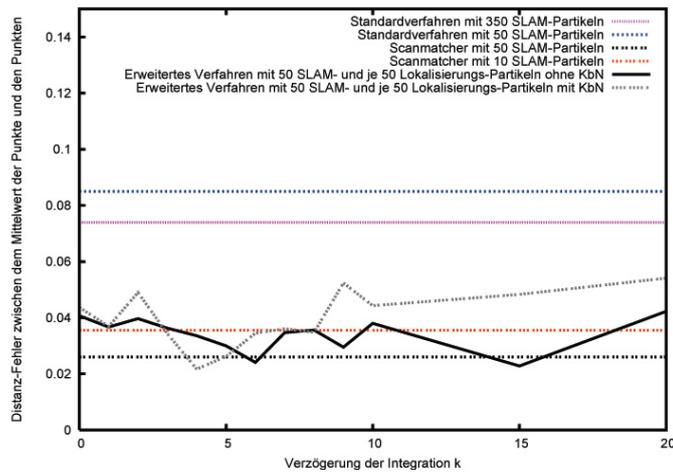


ABBILDUNG 6.4: Distanz-Fehler zwischen dem identischen Start- und Endpunkt bei verschiedenen Verfahren. Der Fehler wurde gemäß Gleichung (6.1) berechnet. Das Standardverfahren mit 350 bzw. 50 SLAM-Partikeln besitzt einen Fehler von 0,074 bzw. 0,086. Das SM-FastSLAM-Verfahren hat bei 50 Partikeln einen Fehler von 0,026 und bei 10 Partikeln einen Fehler von 0,036. Das Standardverfahren in Kombination mit den in dieser Arbeit entwickelten Verfahren ist fast so gut wie SM-FastSLAM und besitzt in den meisten Fällen einen mittleren Fehler zwischen 0,022 und 0,045.

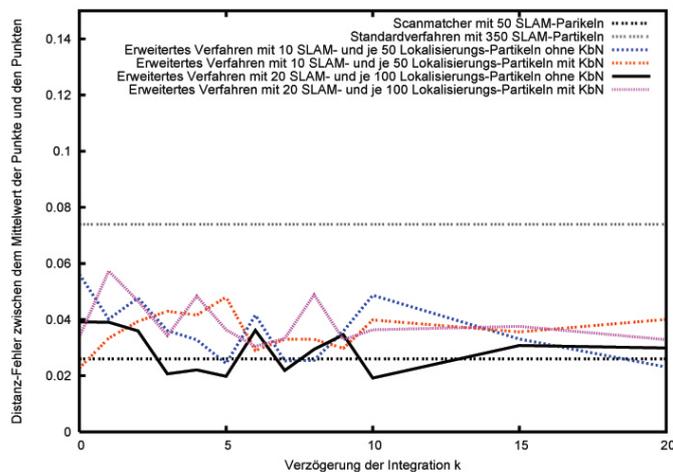


ABBILDUNG 6.5: Zweiter Teil der Fehlerkurven für verschiedene Verfahren. Zur kontrastreichereren Darstellung ist der Fehler für das Standardverfahren bei 350 SLAM-Partikeln und das SM-FastSLAM bei 50 SLAM-Partikeln erneut aufgetragen.

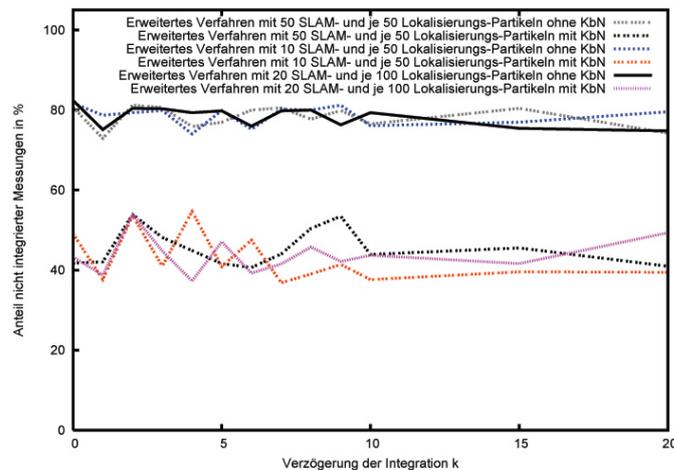


ABBILDUNG 6.6: Der prozentuale Anteil an Messungen, die nicht integriert wurden. Ohne die erlernte Netzstruktur wurden im Schnitt 20%, mit der erlernten Netzstruktur im Schnitt 58% der Messungen für die Kartographierung verwendet.

Abbildung 6.7 und 6.8 visualisieren die entsprechenden Standardabweichungen der SLAM-Partikel der getesteten Verfahren. Auch hier ist zu erkennen, dass die Abweichungen der entwickelten Verfahren deutlich unter dem Standardverfahren liegen. In den meisten Fällen ist die Standardabweichung geringer als die von SM-FastSLAM. Bei den Standardverfahren mit 350 bzw. 50 SLAM-Partikeln ist die erhöhte Abweichung bei 350 mit der erhöhten Anzahl an möglichen Hypothesen zu erklären.

In Abbildung 6.6 ist der prozentuale Anteil an nicht-integrierten Messungen für verschiedene Zeitverzögerungen k aufgetragen. Das Verfahren, das basierend auf der Entropie-Differenz entscheidet, ob die aktuelle Messung integriert wird oder nicht, verhinderte in diesem Experiment im Schnitt die Integration von 78% der Messungen, falls keine korrelationsbasierte Netzstruktur verwendet wurde. Mit diesem Verfahren wurden nur etwa 41% der Messungen verworfen. Dies ist dadurch zu erklären, dass gerade im Innenbereich die Netzstruktur Evidenzen rückgängig gemacht hat, die anschließend teils wieder integriert wurden.

In diesem Experiment wurde gezeigt, dass für den Innenbereich die in dieser Arbeit entwickelten Verfahren beinahe die Leistungsfähigkeit von SM-FastSLAM (der für solche Umgebungen optimal ist) erreichen. Die Verfahren haben einen deutlich geringeren Fehler als das Standardverfahren bei zeitgleich geringerer Varianz. Die Varianz liegt weiterhin unterhalb der von SM-FastSLAM, während der Fehler nur geringfügig größer ist.

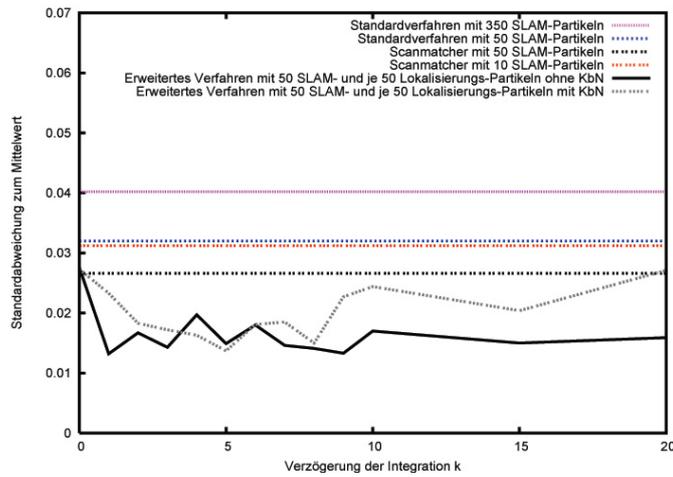


ABBILDUNG 6.7: Standardabweichungen der SLAM-Partikel der getesteten Verfahren. Das Standardverfahren mit 350 bzw. 50 SLAM-Partikeln besitzt eine Abweichung von 0,040 bzw. 0,032. Das SM-FastSLAM-Verfahren mit 50 beziehungsweise 10 SLAM-Partikeln besitzt eine Standardabweichung von 0,026 bzw. 0,031. Die hier entwickelten Verfahren besitzen eine Abweichung zwischen 0,013 und 0,026.

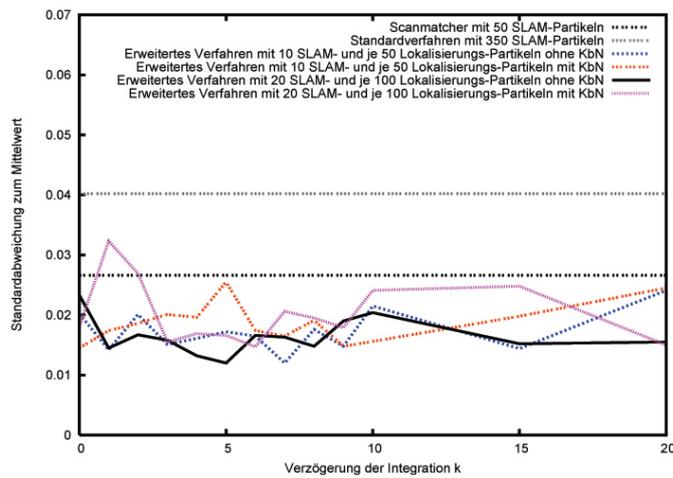


ABBILDUNG 6.8: Zweiter Teil der Standardabweichungen für verschiedene Verfahren. Zur kontrastreicheren Darstellung sind die Werte des Standardverfahrens bei 350 SLAM-Partikeln und des scanmatcherbasierten FastSLAM bei 50 SLAM-Partikeln erneut aufgetragen.

6.2 Experiment im Außenbereich

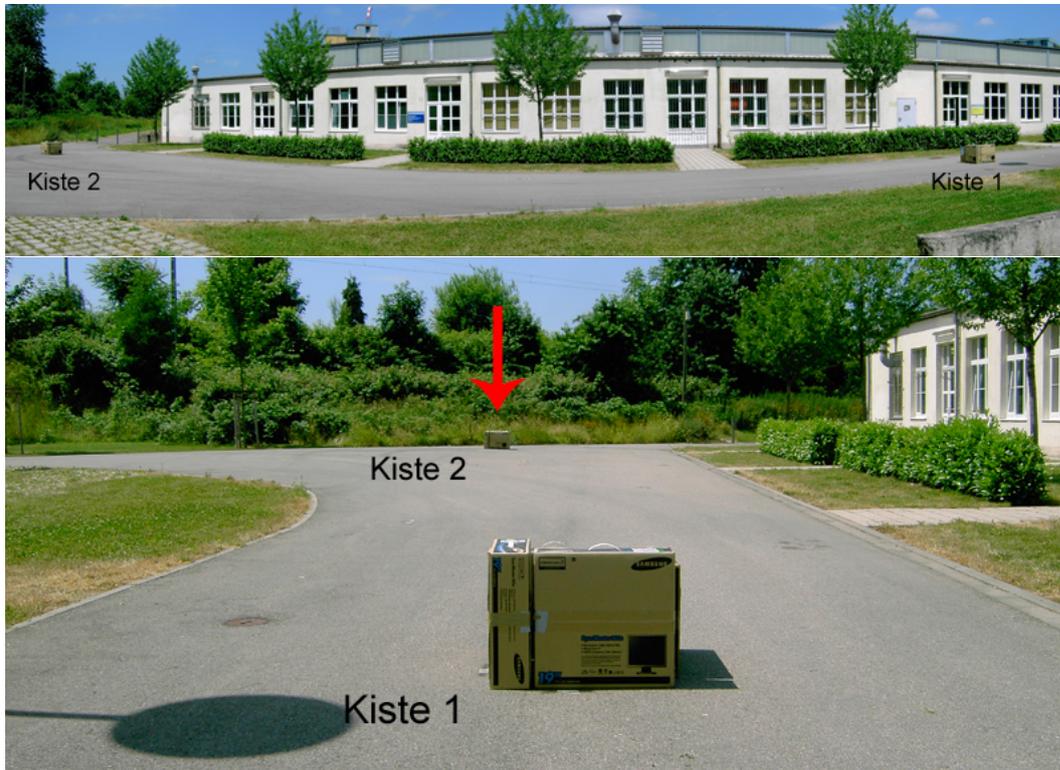


ABBILDUNG 6.9: Ein Experiment im Außenbereich mit zwei Kisten als lokale Struktur.

Um die entwickelten Verfahren auch unter schwierigen Bedingungen zu testen, wurden Experimente mit einer reduzierten Reichweite des Laserscanners (fünf Meter) im Außenbereich durchgeführt (siehe Abbildung 6.9). In dieser Umgebung divergiert das Standardverfahren, da die oft umrundete Kiste 1 in der Karte kontinuierlich an Struktur verliert und sich nicht mehr zur Lokalisierung eignet (siehe Abbildung 6.10). Dagegen ermöglichen die in dieser Arbeit entwickelten Erweiterungen die Erstellung einer akkuraten Karte. Für die quantitative Auswertung wird ein vereinfachtes Szenario mit nur einer Kiste verwendet (siehe Abbildung 6.11). Dieses Experiment soll stellvertretend für die Außenexperimente im Detail beschrieben werden. Dabei umrundete der Roboter die Kiste fünfmal, um anschließend zu einem festen Punkt „X“ im Freien zu fahren. Diese Fahrt wurde insgesamt dreimal wiederholt. Zum Schluss hat der Roboter also 15-mal die Kiste umrundet und war dreimal am Punkt „X“. Während der Umrundung ist der Roboter zusätzlich über den Orientierungspunkt „A“ gefahren, der sich neben der Kiste befand. Somit ergeben sich insgesamt 18 Orientierungspunkte, die für die Fehlerkurven verwendet wurden. Das verwendete Szenario mit eingezeichneten Orientierungspunkten „A“ und „X“ ist in Abbildung 6.11a dargestellt. Abbildung 6.11b skizziert die abgefahrene Trajektorie.

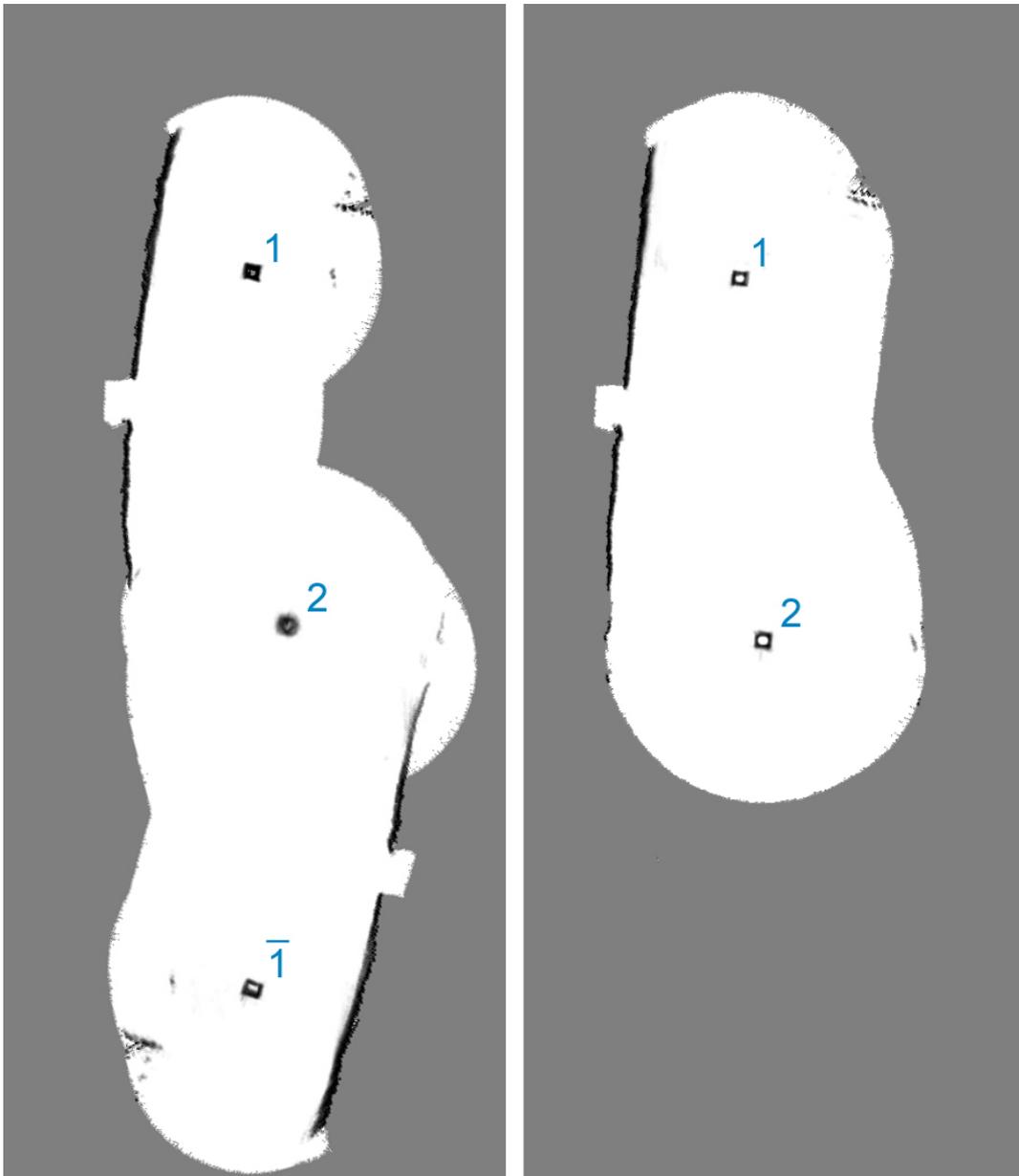


ABBILDUNG 6.10: Generierte Karten des Experimentes im Außenbereich mit zwei Kisten. Der Roboter fuhr von Kiste 1 zu Kiste 2, umrundete Kiste 2 mehrmals, um anschließend wieder zu Kiste 1 zu fahren. Links: Beim Standardverfahren führte die Divergenz der lokalen Struktur (Kiste) zu einer Divergenz der Karte. Rechts: die mit den in dieser Arbeit entwickelten Verfahren generierte Karte.

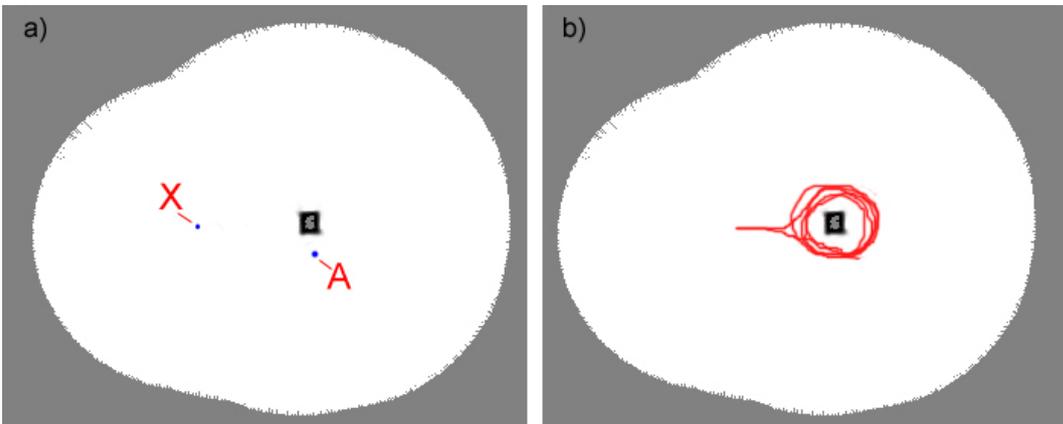
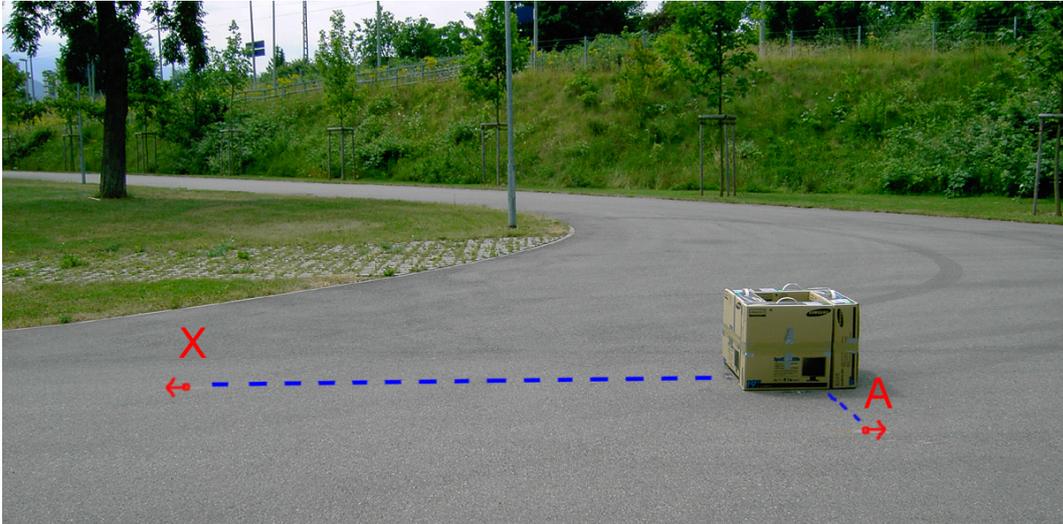


ABBILDUNG 6.11: Oben: Foto des verwendeten Szenarios für die quantitative Auswertung mit den Orientierungspunkten „A“ und „X“. Unten: a) Das Szenario aus der Sicht des Roboters. b) Skizze der abgefahrenen Trajektorie.

Die mit dem Standardverfahren bei 350 SLAM-Partikeln erzeugte Karte ist in Abbildung 6.12 dargestellt. Abbildungen 6.13 und 6.11a zeigen die erzeugten Karten mit dem erweiterten Verfahren bei 20 SLAM- und je 100 Lokalisierungs-Partikeln mit $k = 20$ bzw. $k = 5$. In Abbildung 6.12 ist deutlich zu erkennen, dass die Struktur der Kiste nicht beibehalten werden konnte. Innerhalb der Karten ist die Kiste über die Zeit hinweg zu einem Kreis geworden. Bei einer Messung der Ränder ist nun jede Position um die Kiste herum gleich wahrscheinlich. Im Gegensatz dazu konnte die Struktur mit den hier entwickelten Verfahren vollständig beibehalten werden.

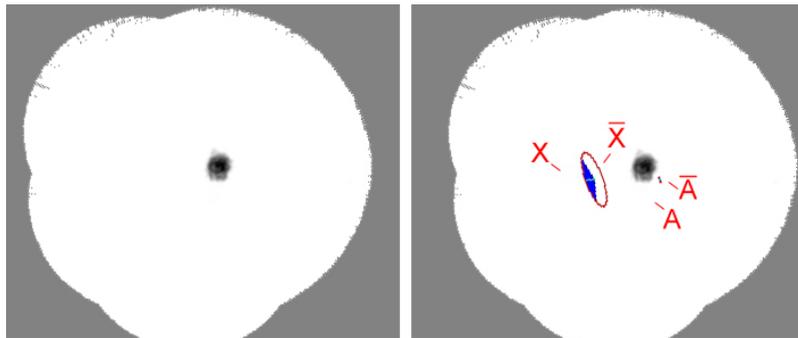


ABBILDUNG 6.12: Mit dem Standardverfahren erzeugte Karte bei 350 SLAM-Partikeln und rechts die zugehörigen geschätzten Orientierungspunkte „ \bar{A} “ und „ \bar{X} “ einschließlich Standardabweichungen. Zur Verdeutlichung wurden die realen Orientierungspunkte „ A “ und „ X “ zusätzlich eingezeichnet. Die Kiste ist bereits nach wenigen Runden zu einem Kreis geworden. Eine genaue Lokalisierung ist an dieser nicht mehr möglich. Deshalb liegen die geschätzten Orientierungspunkte weitab von den realen. Die Unsicherheit in den Punkten zeigt, dass die Karte divergiert.

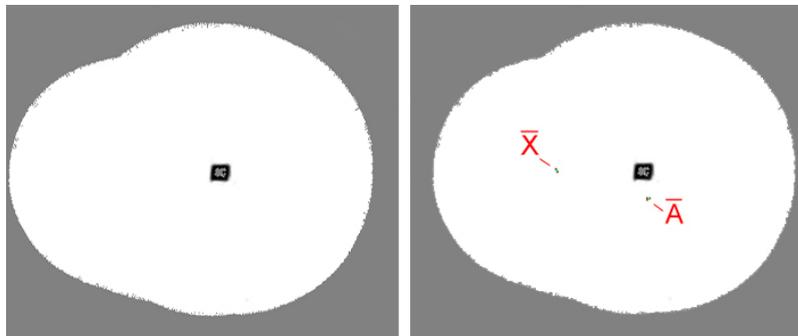


ABBILDUNG 6.13: Mit den in dieser Arbeit entwickelten Verfahren erzeugte Karte bei 20 SLAM- und jeweils 100 Lokalisierungs-Partikeln mit $k = 20$. Rechts sind zusätzlich die Orientierungspunkte einschließlich Standardabweichungen eingezeichnet. Da im Gegensatz zu Abbildung 6.12 die Standardabweichungen sehr gering sind, können diese visuell nicht so deutlich dargestellt werden. Die Struktur der Kiste konnte beibehalten werden. Deshalb ist die Karte nicht divergiert.

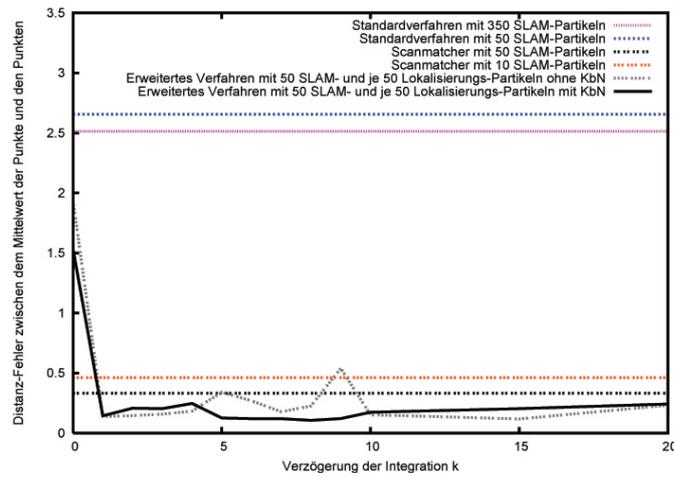


ABBILDUNG 6.14: Erster Teil der Fehlerkurven: Der mittlere Fehler in den Orientierungspunkten für verschiedene Verfahren. Das Standardverfahren besitzt bei 350 bzw. 50 SLAM-Partikeln einen Fehler von 2,51 bzw. 2,66. Das SM-FastSLAM besitzt bei 50 bzw. 10 SLAM-Partikeln einen Fehler von 0,33 bzw. 0,46. Die entwickelten Verfahren besitzen in den meisten Fällen einen Fehler unter 0,20, der damit unter dem des SM-FastSLAM-Verfahrens liegt.

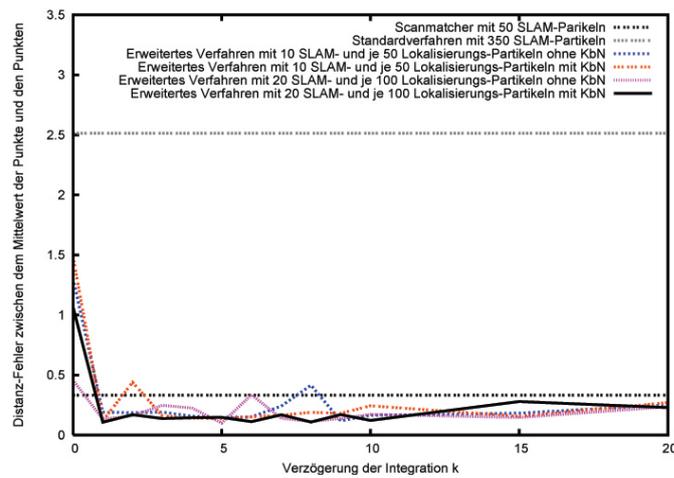


ABBILDUNG 6.15: Zweiter Teil der Fehlerkurven. Zur kontrastreicheren Darstellung ist der Fehler für das Standardverfahren bei 350 SLAM-Partikeln und des SM-FastSLAM-Verfahrens bei 50 SLAM-Partikeln erneut aufgetragen.

In Abbildungen 6.14 und 6.15 sind die mittleren Fehler für die Orientierungspunkte aufgetragen. Zur Verdeutlichung wurden die Fehler der beiden Punkte akkumuliert. In den Grafiken ist zu erkennen, dass der Fehler mit den entwickelten Verfahren deutlich unter dem des Standardverfahrens liegt. Der Fehler liegt in den meisten Fällen, selbst bei einer reduzierten Anzahl an SLAM-Partikeln, unter dem von SM-FastSLAM. In den Abbildungen ist auch zu erkennen, dass bei der Verwendung einer korrelationsbasierten Netzstruktur die Fehlerkurve glatter ist, und in den meisten Fällen unter der liegt, die mit den entwickelten Verfahren, aber ohne die erlernte Netzstruktur erzeugt wurde. Wie erwartet, hat bereits das Verfahren, das der Umschaltung von SLAM auf reine Lokalisierung dient, zu einer präziseren Karte geführt. Dies lässt sich für $k = 0$ an dem geringeren Fehler in den Abbildungen erkennen.

In Abbildungen 6.17 und 6.18 sind die jeweiligen Standardabweichungen der SLAM-Partikel für die Verfahren dargestellt. Die Abweichungen sind etwa halb so groß wie die von SM-FastSLAM und liegen deutlich unter der des Standardverfahrens.

In Abbildung 6.16 sind die Anteile der nicht integrierten Messungen für die jeweiligen Verfahren aufgetragen. Wie beim Experiment im Innenbereich wurden die meisten (hier ca. 78%) Messungen verworfen, d. h. es wurde in den meisten Fällen nur lokalisiert.

In diesem Experiment wurde demonstriert, dass die entwickelten Verfahren die Kartographie auch in einer schwierigen Umgebung ermöglichen und zu einer präziseren Karte führen. Gerade wenn Struktur vorhanden ist, ist es wichtig, diese präzise zu kartieren. Die Verwendung einer korrelationsbasierten Netzstruktur hat sich hierbei als besonders vorteilhaft herausgestellt.

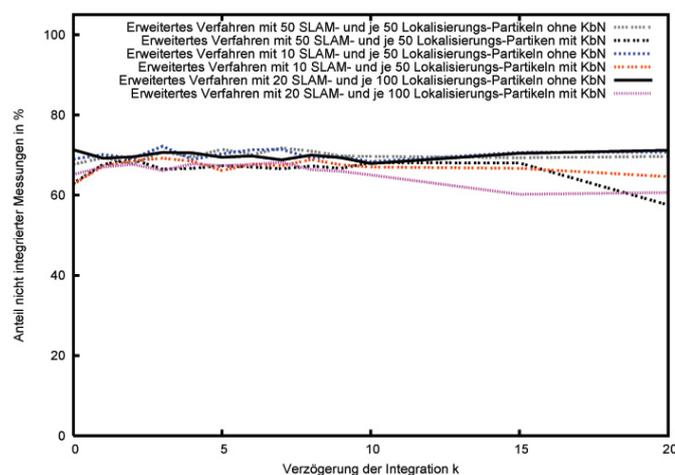


ABBILDUNG 6.16: Im Schnitt wurden ca. 32% der Messungen für die Kartengenerierung verwendet.

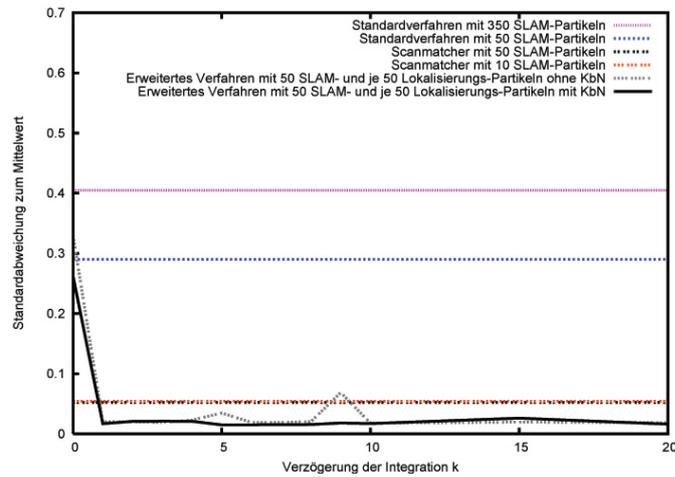


ABBILDUNG 6.17: Erster Teil der Standardabweichungen für verschiedene Verfahren. Die Abweichung des Standardverfahrens bei 350 bzw. 50 SLAM-Partikeln beträgt 0,405 bzw. 0,29. Die geringere Abweichung bei 50 Partikeln ist durch die geringere Anzahl an Kartenhypothesen zu erklären. Die Abweichungen von SM-FastSLAM betragen 0,052 bei 50 SLAM-Partikeln und 0,054 bei 10 SLAM-Partikeln. Für $k > 0$ liegt die Standardabweichung für die entwickelten Verfahren in fast allen Fällen unter 0,021.

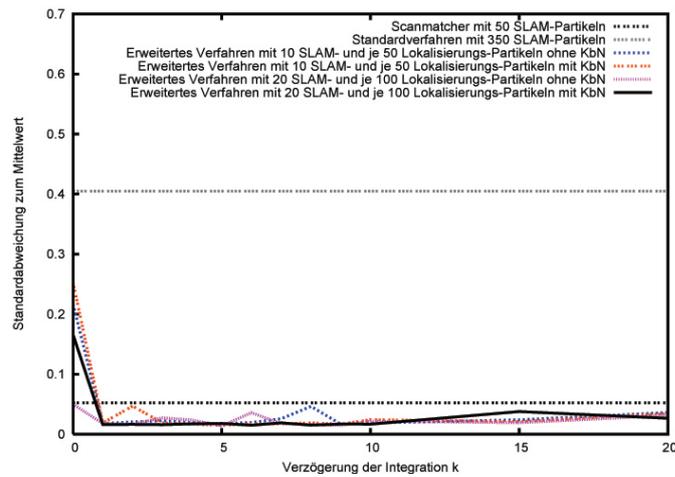


ABBILDUNG 6.18: Zweiter Teil der Standardabweichungen für verschiedene Verfahren. Zur kontrastreicheren Darstellung sind die Werte des Standardverfahrens bei 350 SLAM-Partikeln und von SM-FastSLAM bei 50 SLAM-Partikeln erneut aufgetragen.

6.3 Evaluation der einzelnen Verfahren

6.3.1 Verzögerte Integration

In den Experimenten hat sich bestätigt, dass das Verfahren zur verzögerten Integration einen positiven Einfluss auf den gezogenen Folgezustand besitzt. Die Verwendung eines Lokalisierungs-Partikel-Filters ermöglicht es, ein genaueres Modell zu berechnen, insbesondere, wenn die Odometrie stark vom wahren Pfad abweicht. Dies ist beispielsweise dann der Fall, wenn unterschiedlicher Luftdruck in den Reifen dazu führt, dass der Roboter eine gekrümmte Trajektorie abfährt, obwohl er entlang einer Geraden fahren sollte. Durch die Lokalisierung innerhalb jeder Karte können Fehler dieser Art besser kompensiert werden als mit dem Standardverfahren. Dies ist in den Abbildungen 6.4, 6.5, 6.14 und 6.15 deutlich sichtbar. Der Fehler bei Verwendung einer verzögerten Integration sank im Innenbereich von 0,086 (Standardverfahren mit 50 SLAM-Partikeln) auf (in den meisten Fällen) unter 0,045 (erweitertes Verfahren mit 50 SLAM- und je 50 Lokalisierungs-Partikeln). Die Standardabweichung für dieses Verfahren war stets unter 0,026, was zudem unter der von SM-FastSLAM liegt. Im Außenbereich sank der Fehler von 2,66 auf (in den meisten Fällen) unter 0,20. Weiterhin kann dieses Verfahren für jede Kartenhypothese eine multimodale Positionsverteilung repräsentieren. Diese Eigenschaft besitzt weder das Standard- noch das SM-FastSLAM-Verfahren. Sind weiterhin nur wenige Strukturen vorhanden (beispielsweise im Außenbereich), so kann es vorkommen, dass mehrere der Messungen nacheinander vollständig freies Gebiet abdecken. In diesem Fall wäre jede Position in der Karte, die entsprechend im Erfassungsbereich der Sensoren nur freies Gebiet besitzt, gleich wahrscheinlich. Das SM-FastSLAM-Verfahren kann in diesem Fall auch nur die Odometrie für die Zustandsübergangsverteilung zugrunde legen. Das hier vorgestellte Verfahren ist, bei einer entsprechenden Wahl von k , in der Lage, diese „unsicheren“ Bereiche zu überbrücken. Dadurch können Fehler in der Positionsschätzung früher erkannt und korrigiert werden.

In der verwendeten Implementierung wird für die Berechnung von $p(\mathbf{x}|\mathbf{z}, \mathbf{m})$ ein strahlenbasiertes Modell verwendet. Anschaulich wird dazu der Laserstrahl in die Karte projiziert und die Wahrscheinlichkeit entlang der erfassten Zellen berechnet. Dies ist eine kostenintensive Operation. Dadurch kann für $k > 3$ die verzögerte Integration nicht mehr in Echtzeit eingesetzt werden. Eine Alternative dazu, welche in ersten Experimenten gute Ergebnisse erzielt hat, ist die Verwendung eines endpunktbasierten Modells. Dazu wird die aktuelle Karte mit einem Tiefpassfilter gefaltet. Die Werte in den entsprechenden Zellen können nun als Wahrscheinlichkeiten interpretiert werden. Dies würde auch bei hohem k eine Anwendung zur Laufzeit erlauben.

Die individuelle Stärke dieses Verfahrens ist die Schätzung einer genaueren Zustandsübergangsverteilung. Durch die Lokalisierung innerhalb jeder Kartenhypothese kann ein genauerer Folgezustand gezogen werden. Zudem ist es möglich, die verwendeten Übergangsmodelle mit einem stärkeren Rauschen zu verwenden, falls entsprechend viele Lokalisierungs-Partikel verwendet

werden. Das Verfahren ist durch eine entsprechende Wahl von k in der Lage, „unsichere“, z. B. vollkommen freie, Bereiche zu überbrücken.

6.3.2 Adaptives Umschalten von SLAM auf reine Lokalisierung

Das Verfahren zur adaptiven Umschaltung von SLAM auf reine Lokalisierung hat sich als ein effizientes Verfahren erwiesen. In den Experimenten wurden im Schnitt 80% der Messungen verworfen und somit nur etwa 20% für die Kartographierung verwendet. Das alleinige Anwenden dieses Verfahrens hat bereits einen entscheidenden Einfluss auf die Präzision der generierten Karte. Es sind deutlich weniger „Verschmierungen“ in der Karte zu erkennen. Quantitativ ist der Lokalisierungsfehler im Innenbereich etwa halb so groß. Der Fehler beträgt im Innenbereich nur etwa 0,04 mit und 0,086 ohne das Verfahren (siehe Abbildung 6.4 und 6.5 für die Experimente ohne erlernte Netzstruktur und $k = 0$). Im Außenbereich ist der Fehler von 2,66, was bei Verwenden von $\lambda = 0,5$ in Gleichung 6.1 als ein mittlerer Fehler in allen Dimensionen angesehen werden kann, auf im Schnitt 1,4 gesunken. Die Wahl der Schranke τ hat dabei einen entscheidenden Einfluss auf das Verfahren. Je höher der Wert von τ , desto mehr Messungen werden integriert und desto höher ist die Wahrscheinlichkeit für die Integration ungünstiger Messungen. Hier muss ein Kompromiss zwischen Kartographierung und Lokalisierung getroffen werden. Für die Berechnung der Entropie-Differenz ist die Integration der Messung notwendig. Daher ergibt sich kein Berechnungsgewinn durch das Verwerfen von Messungen.

Die individuelle Stärke dieses Verfahrens ist die adaptive Umschaltung von SLAM auf reine Lokalisierung. Fehlmessungen und Messungen, die an ungünstigen Positionen integriert würden, können erkannt und ihre Integration verhindert werden. Dadurch wird die Messung nur für eine Lokalisierung innerhalb der Karte verwendet.

6.3.3 Erlernen einer korrelationsbasierten Netzstruktur

Gerade in Bereichen mit geringer lokaler Struktur ist es wichtig, diese präzise in der Karte beizubehalten. Die erlernte Netzstruktur konnte in den meisten Fällen die Struktur mit Hilfe späterer korrekter Messungen wiederherstellen. Dabei hat sich gezeigt, dass die Netzstruktur besonders gut mit dem Verfahren zum adaptiven Umschalten von SLAM auf reine Lokalisierung zusammenarbeitet. Das letztgenannte Verfahren erhöht die Wahrscheinlichkeit, dass nur „gute“ Messungen dazu verwendet werden, bestehende Artefakte zu korrigieren. Abbildung 6.19 zeigt erzeugte Karten für verschiedene Szenarien jeweils mit und ohne die Verwendung einer korrelationsbasierten Netzstruktur.

Die individuelle Stärke dieses Verfahrens ist die Möglichkeit, Artefakte nachträglich zu entfernen und dadurch eine präzisere Karte zu generieren.

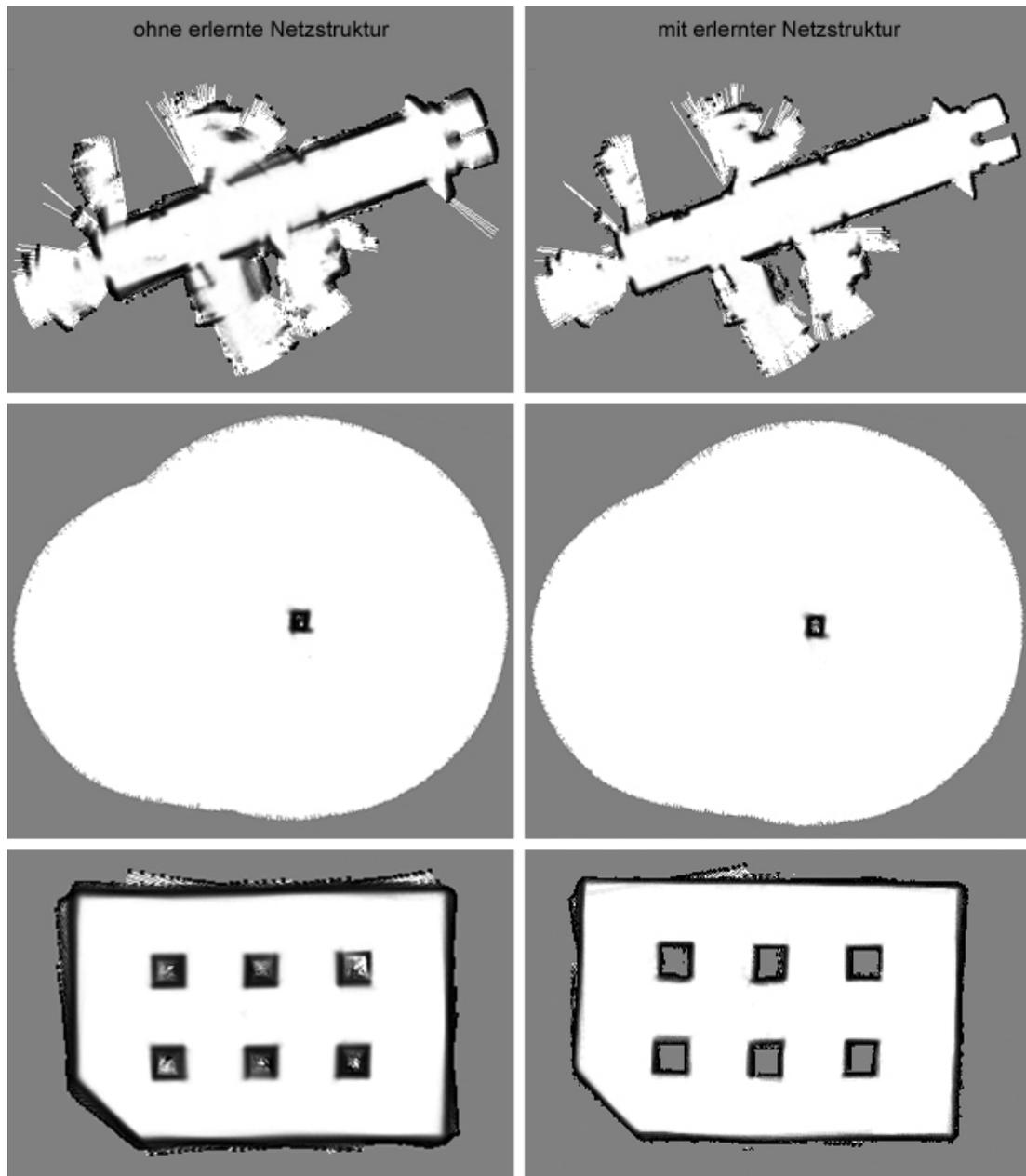


ABBILDUNG 6.19: Erstellte Karten für verschiedene Szenarien jeweils ohne (links) und mit (rechts) einer erlernten Netzstruktur.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurden aufbauend auf dem *Rao-Blackwellized Particle Filter* drei Ansätze vorgestellt, die der Reduzierung von Kartenunsicherheiten dienen. Eines der Verfahren dient zugleich als Verfahren zur Umschaltung von SLAM auf reine Lokalisierung. Dazu wurde der Kontext der Kartengenerierung und zeitgleichen Lokalisierung in drei Zeitabschnitten untersucht: die Zeitpunkte vor, während und nach der Integration der aktuellen Messung. Für jeden dieser Zeitpunkte wurde ein Verfahren entwickelt. Das Verfahren vor der Integration der aktuellen Messung verzögert deren Integration, verwendet sie aber, um eine genauere Zustandsübergangsfunktion zu schätzen. Die vorhandenen, aber aufgrund der Verzögerung nicht integrierten Messungen werden für eine Lokalisierung innerhalb der jeweiligen Karte mit einem Lokalisierungs-Partikel-Filter verwendet. Anschließend wird die nächste Position aus der Stichprobe des Lokalisierungs-Partikel-Filters gezogen. In Abschnitt 5.1 wurde dazu die formale Herleitung der Zustandsübergangsfunktion und Gewichtskorrektur der SLAM-Partikel präsentiert. Während der Integration wird durch ein weiteres Verfahren errechnet, welchen Informationsgewinn die Integration der aktuellen Messung hätte. Dies wird durch Berechnen der Entropie-Differenz auf den erfassten Zellen erreicht. Ist die Differenz unter einer festgelegten Schranke τ , wird die Messung integriert, ansonsten verworfen. Um die Integration von Fehlmessungen auch während der Exploration zu minimieren, wird die Entropie-Differenz für maximal eine unbekannte Zelle je Laserstrahl berechnet. Durch das dritte Verfahren wird fortlaufend eine korrelationsbasierte Netzstruktur erlernt. Dadurch kann für einzelne Zellen die Integration von Messungen nachträglich rückgängig gemacht werden. Mit Hilfe dieses Verfahrens ist es möglich, Artefakte zu entfernen und Strukturen beizubehalten.

Die entwickelten Verfahren wurden implementiert und auf einem realen Roboter, der mit einem Laserscanner ausgerüstet ist, getestet. In den Experimenten wurde demonstriert, dass diese Vorgehensweise auch in schwierigen Umgebungen zu einer Reduzierung der Kartenunsicherheit führt und basierend auf der aktuellen Messung für eine Umschaltung von SLAM auf reine Lokalisierung geeignet ist. Dafür wurden typische Arbeitsumgebungen eines Roboters, sowohl im Innen- als auch im Außenbereich, verwendet.

Zusammenfassend lässt sich sagen, dass in Bereichen mit viel Struktur, also dort, wo der *Scan-matcher* optimal ist, die hier entwickelten Verfahren nahe an die Leistungsfähigkeit von SM-FastSLAM heranreichen. Im Bereichen mit geringer lokaler Struktur (z. B. im Außenbereich) sind die Ergebnisse der entwickelten Verfahren deutlich besser als die von SM-FastSLAM. Die

hier entwickelten Verfahren können also universell eingesetzt werden. Für die Wahl des Parameters k hat sich ein Wert zwischen 3 und 5 als gut erwiesen. Die Wahl der Schranke $\tau = -2$ führte stets zu guten Ergebnissen.

Die feste Schranke τ für das Verfahren, das basierend auf der Entropie-Differenz entscheidet, ob die aktuelle Messung integriert wird oder nicht, ist bisher aufgrund von Erfahrungswerten ermittelt worden. Das Erlernen dieser Schranke würde die Arbeitsweise des Verfahrens weiter optimieren. Weiterhin könnte es sinnvoll sein, die Wahl der Schranke von der Umgebung abhängig zu machen. Werden durch die aktuelle Messung viele unbekannte Zellen erfasst, so kann dies zwei Gründe haben. Zum einen könnte es sich um eine Exploration handeln, zum anderen um eine Fehlmessung. Um diese Diskrepanz zu entdecken, könnten Folgen von Messungen betrachtet werden. Während der Exploration könnte anschließend ein geringeres τ verwendet werden.

Eine zusätzliche Erweiterung des Systems wäre die Möglichkeit, die Verzögerung der Integration k adaptiv zu bestimmen. In Bereichen mit wenig Struktur (z. B. im Außenbereich) ist ein größeres k effizienter, da dadurch die Wahrscheinlichkeit erhöht wird, unsichere Regionen (z. B. Bereiche, in denen alle Zellen „frei“ sind) zu überbrücken und einen möglichen Positionsfehler früher zu erkennen und korrigieren. Im Gegensatz dazu ist bei Vorhandensein von vielen Merkmalen (z. B. im Innenbereich) ein kleines k ausreichend.

Ein weiteres Verfahren konnte aus zeitlichen Gründen nicht weiter verfolgt werden. Die Kernidee dieses Verfahrens ist es, eine Teststatistik für einzelne Zellen zu entwickeln. Durch eine solche Statistik können Konfidenzaussagen bezüglich der Belegungswahrscheinlichkeit gemacht werden. Als Testverfahren kann beispielsweise der Likelihood-Quotienten-Test verwendet werden. Um ein Testverfahren zu entwickeln, sind dazu theoretisch alle möglichen Wahrscheinlichkeitsverteilungen für die Belegungswahrscheinlichkeit einer Zelle für einen gegebenen Abstand und Umgebung zu erlernen. Die Idee ist, dass sich die meisten dieser Verteilungen in der Praxis kaum voneinander unterscheiden würden und deshalb nur eine geringe Anzahl von Verteilungen erlernt werden müsste. Dieses Testverfahren könnte anschließend dazu verwendet werden, sogenannte „Konfidenzkarten“ zu erlernen. Diese würden dann die Sicherheiten über die Belegungswahrscheinlichkeit einer Zelle wiedergeben. Diese Karten können einerseits als Gütemaß, andererseits für eine aktive Exploration bzw. reine Navigation verwendet werden.

A Technische Daten

A.1 ActivMedia Robotics Pioneer 3-DX



ABBILDUNG A.1: Der Roboter der Firma *ActivMedia Robotics* vom Typ *Pioneer 3-DX*

Parameter:	Wert:
Abmessungen (L × B × H)	44,5cm × 40cm × 24,5cm
Gewicht	min. 9kg
Nutzlast	max. 23kg
Antrieb	2-Rad-Antrieb, Differential
Geschwindigkeit	max. 1,2 $\frac{m}{s}$
Ultraschallsensoren (front)	8 Stück, je 15°
Laser	SICK LMS
Prozessor	Hitachi H8S

TABELLE A.1: Herstellerangaben zum *Pioneer 3-DX*

A.2 SICK-LMS-Laserscanner



ABBILDUNG A.2: Der Laserscanner der Firma SICK vom Typ LMS

Parameter:	Wert:
Abmessungen (L × B × H)	15,6cm × 15,5cm × 21cm
Gewicht	4,5kg
Öffnungswinkel	180°
Winkelauflösung	1° ... 0,25°
Auflösung	10mm
Reichweite	80m
Systematischer Fehler	± 15mm
Statistischer Fehler (1 Sigma)	5mm

TABELLE A.2: Herstellerangaben zum LMS 200-30106

Abbildungsverzeichnis

1.1	Probleme der rasterkartenbasierten Kartographierung	3
1.2	Beispiel für den <i>Rao-Blackwellized Particle Filter</i>	4
3.1	Abhängigkeiten der Zustandsvariablen unter der Markow-Annahme	11
3.2	Beispiel für die sequentielle Zustandsschätzung mit dem Kalman-Filter im ein-dimensionalen Fall	14
3.3	Eine nichtlineare Funktion und ihre gaußsche Approximation	15
3.4	Visualisierung verschiedener Resampling-Verfahren	21
3.5	Beispiel für den Partikel-Filter-Algorithmus	22
4.1	Beispiel für verschiedene Kartenrepräsentationen	24
4.2	Abhängigkeiten der Zustandsvariablen unter der Markow-Annahme bei bekannter Karte	26
4.3	Darstellung des odometriebasiertem Zustandsübergangsmodells	27
4.4	Beispiel für die Lokalisierung mit dem Partikel-Filter bei bekannter Karte	28
4.5	Vergleich zwischen Ultraschall- und Lasersensoren	30
4.6	Beispiel für die Integration einer Messung nach dem <i>Simple-Occupancy-Grid-Mapping</i> -Verfahren	30
4.7	Der <i>Rao-Blackwellized Particle Filter</i> anhand eines Beispiels mit drei Partikeln	32
5.1	Beispiel für eine Vergrößerung der Kartenunsicherheit durch Integration einer Messung an einer ungünstigen Position	33
5.2	Beispiel für eine erschwerte Lokalisierung aufgrund einer unpräzisen Karte	34
5.3	Die entwickelten Verfahren und ihre zeitliche Arbeitsweise	34
5.4	Erster Teil des Beispiels zur verzögerten Integration	36
5.5	Zweiter Teil des Beispiels zur verzögerten Integration	37
5.6	Beispiel für den Verlust lokaler Struktur durch Integration von Messungen an ungünstigen Positionen	42
5.7	Beispiel für die Berechnung der Entropie-Differenz	44
5.8	Erstellte Karten mit und ohne das Verfahren zur nachträglichen Korrektur von Messintegrationsfehlern	45
5.9	Beispiel für eine Korrektur und Korrelationserlernung im eindimensionalen Fall	46

5.10	Die Acht-Nachbarschaftsbeziehung für eine Zelle	47
6.1	Experiment im Innenbereich: Foto des Szenarios	52
6.2	Experiment im Innenbereich: Mit dem Standardverfahren erzeugte Karten . . .	54
6.3	Experiment im Innenbereich: Mit den in dieser Arbeit entwickelten Verfahren erzeugte Karten	54
6.4	Experiment im Innenbereich: Fehlerkurven der getesteten Verfahren (erster Teil)	55
6.5	Experiment im Innenbereich: Fehlerkurven der getesteten Verfahren (zweiter Teil)	55
6.6	Experiment im Innenbereich: prozentualer Anteil an Messungen, die nicht integriert wurden	56
6.7	Experiment im Innenbereich: Standardabweichungen der getesteten Verfahren (erster Teil)	57
6.8	Experiment im Innenbereich: Standardabweichungen der getesteten Verfahren (zweiter Teil)	57
6.9	Experiment im Außenbereich: Foto des Szenarios mit zwei Kisten	58
6.10	Experiment im Außenbereich: Generierte Karten bei zwei Kisten	59
6.11	Experiment im Außenbereich: Foto und Skizze des Szenarios mit einer Kiste .	60
6.12	Experiment im Außenbereich: Mit dem Standardverfahren erzeugte Karte . . .	61
6.13	Experiment im Außenbereich: Mit den in dieser Arbeit entwickelten Verfahren erzeugte Karte	61
6.14	Experiment im Außenbereich: Fehlerkurven der getesteten Verfahren (erster Teil)	62
6.15	Experiment im Außenbereich: Fehlerkurven der getesteten Verfahren (zweiter Teil)	62
6.16	Experiment im Außenbereich: prozentualer Anteil an Messungen, die nicht integriert wurden	63
6.17	Experiment im Außenbereich: Standardabweichungen der getesteten Verfahren (erster Teil)	64
6.18	Experiment im Außenbereich: Standardabweichungen der getesteten Verfahren (zweiter Teil)	64
6.19	Erstellte Karten mit und ohne erlernte Netzstruktur für verschiedene Szenarien .	67
A.1	Der Roboter der Firma <i>ActivMedia Robotics</i> vom Typ <i>Pioneer 3-DX</i>	71
A.2	Der Laserscanner der Firma <i>SICK</i> vom Typ <i>LMS</i>	72

Literaturverzeichnis

- [CAR] *Carnegie Mellon Robot Navigation Toolkit*. <http://carmen.sourceforge.net/>.
- [BAGNELL et al. 2000] BAGNELL, D., A. COURVILLE und C. WELLINGTON (2000). *Learning Inverse Sensor Models*.
- [DISSANAYAKE et al. 2001] DISSANAYAKE, M., P. NEWMAN, S. CLARK, H. DURRANT-WHYTE und M. CSORBA (2001). *A solution to the simultaneous localization and map building (SLAM) problem*. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241.
- [DOUCET et al. 2000a] DOUCET, A., N. DE FREITAS, K. MURPHY und S. RUSSELL (2000a). *Rao-Blackwellised particle filtering for dynamic Bayesian networks*. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, S. 176–183.
- [DOUCET et al. 2000b] DOUCET, A.S., S. GODSILL und C. ANDRIEU (2000b). *On sequential Monte Carlo sampling methods for Bayesian filtering*. *Statistics and Computing*, 10(3):197–208.
- [EUSTICE et al. 2005] EUSTICE, R. M., H. SINGH und J. J. LEONAR (2005). *Exactly sparse delayedstate filters*. In: *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, S. 2428–2435.
- [FOX 2003] FOX, D. (2003). *Adapting the Sample Size in Particle Filters Through KLD-Sampling*. *International Journal of Robotics Research*, 22(12):985–1003.
- [FOX et al. 1999] FOX, D., W. BURGARD, F. DELLAERT und S. THRUN (1999). *Monte carlo localization: Efficient position estimation for mobile robots*. *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.
- [GRISSETTI et al. 2005] GRISSETTI, GIORGIO, G. D. TIPALDI, C. STACHNISS, W. BURGARD und D. NARDI (2005). *Speeding-Up Rao-Blackwellized SLAM*. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Orlando, FL, USA. To Appear.
- [GRZONKA 2005] GRZONKA, SLAWOMIR (2005). *Kamerabasierte Lageschätzung eines autonomen Roboters mittels Partikelfiltern*. Studienarbeit, Albert-Ludwigs-Universität Freiburg.

- [HAHNEL et al. 2003] HAHNEL, D., W. BURGARD, B. WEGBREIT und S. THRUN (2003). *Towards lazy data association in SLAM*. Proceedings of the 11th International Symposium of Robotics Research (ISRR).
- [IGLESIAS et al. 2005] IGLESIAS, U.N.R., T. KYRIACOU und S. BILLINGS (2005). *Robot programming through a combination of manual training and system identification*. 2nd European Conference on Mobile Robots.
- [KALMAN 1960] KALMAN, R.E. (1960). *A new approach to linear filtering and prediction problems*. Journal of Basic Engineering, 82(1):35–45.
- [LIU 1996] LIU, J.S.G. (1996). *Metropolized independent sampling with comparisons to rejection sampling and importance sampling*. Statistics and Computing, 6(2):113–119.
- [MITCHELL 1997] MITCHELL, T.M. (1997). *Machine Learning*. McGraw-Hill Higher Education.
- [MURPHY 1999] MURPHY, K. (1999). *Bayesian map learning in dynamic environments*. Advances in Neural Information Processing Systems (NIPS), 12:1015–1021.
- [NEHMZOW et al. 1993] NEHMZOW, U., T. SMITHERS und B. MCGONIGLE (1993). *Increasing Behavioural Repertoire in a Mobile Robot*.
- [OWEN und NEHMZOW 1996] OWEN, CARL und U. NEHMZOW (1996). *Route Learning in Mobile Robots through Self-Organisation*. In: *Euromicro workshop on advanced mobile robots*. IEEE Computer Society, ISBN 0-8186-7695-7.
- [PFAFF et al. 2006] PFAFF, P., W. BURGARD und D. FOX (2006). *Robust Monte-Carlo Localization using Adaptive Likelihood Models*.
- [RISTIC et al. 2004] RISTIC, BRANKO, S. ARULAMPALAM und N. GORDON (2004). *Beyond the Kalman Filter, Particle Filters for Tracking Applications*, Kap. 1-3, S. 1–66. Artech House Publishers.
- [ROTTMANN et al. 2005] ROTTMANN, A., O. MARTÍNEZ MOZOS, C. STACHNISS und W. BURGARD (2005). *Place Classification of Indoor Environments with Mobile Robots using Boosting*. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, USA.
- [STACHNISS 2004] STACHNISS, C. (2004). *Exploration and Mapping with Mobile Robots*. PhD Thesis, University of Freiburg, Department of Computer Science.
- [STACHNISS et al. 2005] STACHNISS, C., G. GRISSETTI und W. BURGARD (2005). *Information Gain-based Exploration Using Rao-Blackwellized Particle Filters*. In: *Proc. of Robotics: Science and Systems (RSS)*, S. 65–72, Cambridge, MA, USA.

- [STACHNISS et al. 2004a] STACHNISS, C., G. GRISSETTI, D. HÄHNEL und W. BURGARD (2004a). *Improved Rao-Blackwellized Mapping by Adaptive Sampling and Active Loop-Closure*. In: *Proc. of the Workshop on Self-Organization of Adaptive behavior (SOAVE)*.
- [STACHNISS et al. 2004b] STACHNISS, C., D. HAEHNEL und W. BURGARD (2004b). *Exploration with Active Loop-Closing for FastSLAM*. S. 1505–1510, Sendai, Japan.
- [TEN HAGEN und KROSE 2002] TEN HAGEN, S. und B. KROSE (2002). *Trajectory reconstruction for self-localization and map building*. Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference, 2.
- [THRUN 2002] THRUN, S. (2002). *Robotic Mapping: A Survey*. School of Computer Science, Carnegie Mellon University.
- [THRUN und BÜCKEN 1996] THRUN, S. und A. BÜCKEN (1996). *Learning Maps for Indoor Mobile Robot Navigation*. School of Computer Science, Carnegie Mellon University.
- [THRUN et al. 2004a] THRUN, S., Y. LIU, D. KOLLER, A. NG, Z. GHAHRAMANI und H. DURRANT-WHYTE (2004a). *Simultaneous Localization and Mapping With Sparse Extended Information Filters*. International Journal of Robotics Research. To Appear.
- [THRUN et al. 2004b] THRUN, S., M. MONTEMERLO, D. KOLLER, B. WEGBREIT, J. NIETO und E. NEBOT (2004b). *FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association*. Journal of Machine Learning Research.
- [THRUN et al. 2005] THRUN, SEBASTIAN, W. BURGARD und D. FOX (2005). *Probabilistic Robotics*. The MIT Press.
- [THRUN et al. 1998] THRUN, S.I., W. BURGARD und D. FOX (1998). *A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots*. Autonomous Robots, 5(3):253–271.
- [UNECE 2005] UNECE, UNITED NATIONS ECONOMIC COMMISSION FOR EUROPE (2005). *Press Release ECE/STAT/05/P03*. http://www.unece.org/press/pr2005/05stat_p03e.pdf.
- [ZHANG et al. 2005] ZHANG, SEN, L. XIE und M. D. ADAMS (2005). *Entropy Based Adaptive Feature Selection Scheme for Real Time Simultaneous Localization and Map Building*. Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference.