

## **Simultaneous Localisation and Mapping (SLAM) mit Hierarchischen Partikelfiltern**

Holger Täubig

Technische Universität Ilmenau  
Fachgebiet Neuroinformatik  
holger.taeubig@carabas.de

Christof Schröter

Technische Universität Ilmenau  
Fachgebiet Neuroinformatik  
christof.schroeter@tu-ilmenau.de

### **Abstract**

Von vielen wird die Fähigkeit eines Roboters zur Lokalisation in einer unbekannt-ten Umwelt bei gleichzeitigem Aufbau einer genauen Karte, also die Lösung der aus der Literatur als „Simultaneous Localization and Mapping“ Problem bekann-ten Aufgabe, als wesentliche Vorbedingung für einen wirklich autonomen Robo-ter angesehen. Lösungen für dieses Problem wurden bisher vorwiegend mit Hilfe von Robotern vorgestellt, die über Laser-Range-Sensoren oder ähnlich geartete Sensoren verfügen. Diese stellen für einzelne wiedererkennbare Objekte der Um-welt, sogenannte Landmarken, sowohl Richtungs- als auch Entfernungsinforma-tionen relativ zur aktuellen Roboterposition zur Verfügung. Da derartige Sensoren nicht an jedem Roboter vorhanden sind, ergibt sich die Frage, ob die Lösung des SLAM Problems ebenso möglich ist, wenn statt Richtung und Entfernung lediglich die Richtung einer Landmarke zur Verfügung steht. Eine solche Lösung würde SLAM Anwendungen auf Basis anderer bzw. schlechterer Sensoren ermöglichen. Die gleiche Problemstellung entsteht, wenn SLAM visuell mit Hilfe einer einzel-nen Kamera erfolgen soll, ohne dabei aufwendige Pseudo-Stereo-Algorithmen zu verwenden. Dann stehen keine oder nur ungenaue Entfernungsbeobachtungen der Landmarke zur Verfügung. Der von uns entwickelte Algorithmus stellt eine Lösung für genau diese Problemstellung dar. Er realisiert SLAM mit Beobach-tungen geringerer Dimension (ohne Entfernungsangabe) unter Verwendung ei-ner einzelnen Kamera. Der Algorithmus arbeitet ähnlich wie FastSLAM. Dessen Kalman-Filter wurden durch weitere Partikelfilter ersetzt. Diese Partikelfilter bie-ten eine allgemeinere Möglichkeit, die Wahrscheinlichkeitsverteilung der Position jeder einzelnen Landmarke darzustellen, als Kalman-Filter dies tun, da Kalman-Filter auf normalverteilte unimodale Verteilungen beschränkt sind.

## **1 Einleitung**

Die Aufgabe, aus einer Reihe von Landmarkenbeobachtungen eines Roboters eine Karte für eine bisher unbekannt-ete Umwelt aufzubauen, induziert immer auch die Aufgabe, den Roboter in dieser



(Kartenaufbauproblem, siehe Abb. 1(c) ) darstellen. Es besteht in der Ermittlung der Zeitreihe des Roboterzustandes  $s_{1:t}$  und der Karte  $m$  (bestehend aus den über die Zeit konstanten globalen Positionen  $m_i$  der Landmarken  $i \in \{1, \dots, M\}$ ) aus den Daten der bekannten Zeitreihe der Steuerkommandos  $u_{1:t}$  und der Zeitreihe der durchgeführten Beobachtungen  $z_{1:t}$ . In diesem Artikel wird vorausgesetzt, dass zu jedem Zeitpunkt  $t$  eindeutig feststellbar ist, welche Landmarke  $i_t$  beobachtet wurde. Das Objekterkennungsproblem „Wiedererkennung einer Landmarke“ wird somit aus den Betrachtungen ausgeklammert und experimentell einfach über eine eindeutige Farbkodierung gelöst.

Zusätzlich zur dargestellten Problemstruktur wirken sich weitere Faktoren auf die Schwierigkeit der Lösung des SLAM Problems aus. Hier ist beispielsweise die Anzahl  $M$  der zu betrachtenden Landmarken als offensichtlichste Größe zu bedenken, die einen erheblichen, allerdings vom Lösungsalgorithmus abhängigen Einfluss auf die Zeitkomplexität der Lösung aufweist. Weiterhin sind der Bewegungsfehler des Roboters (Abweichung der tatsächlichen Bewegung des Roboters vom Steuerkommando  $u_t$ ) und der Beobachtungsfehler der Landmarken (Fehler bei der Bestimmung der relativen Landmarkenposition) zu nennen. Diese werden üblicherweise im Algorithmus modelliert und besitzen meist keinen Einfluss auf die Zeitkomplexität des Algorithmus. Sie wirken sich aber entsprechend der Größe des Fehlers und entsprechend der Genauigkeit der Modellierung des Fehlers auf die Qualität der Ergebnisse aus.

## Zielstellung

Ein weiterer Problemparameter bildet den Ausgangspunkt für die vorliegende Untersuchung. Es handelt sich dabei um die Struktur der Beobachtungen  $z_1, \dots, z_t$ . In zahlreichen Untersuchungen wurden Lösungen des SLAM Problems vorgestellt, denen durch die Sensorik der verwendeten Roboter Beobachtungen bestehend aus Beobachtungsrichtung und beobachteter Entfernung einer Landmarke zur Verfügung gestellt wurden [2, 4]. Da derartige Sensoren nicht an jedem Roboter vorhanden sind, ergibt sich die Frage, ob die Lösung des SLAM Problems ebenso möglich ist, wenn statt Richtung und Entfernung lediglich die Richtung einer Landmarke zur Verfügung steht. Eine solche Lösung würde SLAM Anwendungen auf Basis anderer bzw. schlechterer Sensoren ermöglichen. Es könnten dann zum Beispiel Sensoren, die ähnlich Bewegungssensoren lediglich Aktivitäten in einem definierten Sektor signalisieren können, verwendet werden. Die gleiche Problemstellung entsteht, wenn die Landmarkendetektion visuell mit Hilfe einer einzelnen Kamera erfolgen soll. Soll dann auf aufwendige Pseudo-Stereo-Algorithmen verzichtet werden, so stehen zur Realisierung von SLAM keine oder nur sehr ungenaue Beobachtungen der Entfernung der Landmarken zur Verfügung. Der vorliegende Artikel widmet sich genau diesem Problem und zeigt, dass sich SLAM auch mit Beobachtungen geringerer Dimension (ohne Entfernungsangabe) realisieren lässt. Offensichtlich wird das Problem dabei eher schwieriger denn leichter, da aus weniger Beobachtungen die gleiche Menge an Informationen geschlossen werden soll. Der praktische Nachweis der Funktionstüchtigkeit des Algorithmus wird unter Verwendung eines Roboters mit Panorama-Kamera und unter Zuhilfenahme eines einfachen farbbasierten Landmarkendetektors geführt. Die Aufgabenstellung besteht demnach in der Durchführung von SLAM unter Verwendung einer einzelnen Kamera.

## FastSLAM als Basis des Algorithmus

Der von uns entwickelte Ansatz baut auf dem FastSLAM Algorithmus auf, wie er in [4] vorgestellt wurde. Dieser löst das SLAM Problem mit  $M$  Landmarken selbst bei einfacher Implementierung mit einer Zeitkomplexität von  $O(fM)$  für einen konstanten Faktor  $f$ . Für die gleiche Aufgabenstellung wie der bis dato weit verbreitete Ansatz aus [5], einen EKF(Extended Kalman Filter) auf dem aus Roboterzustand  $s_t$  und Karte  $m$  kombinierten Zustandsvektor  $(s_t, m_1, \dots, m_M)^T$  zu berechnen, eine Komplexität von  $O(M^2)$  auf (siehe [2]). Aus diesem Grund wurde der hier vorgestellte Algorithmus von Anfang an auf Basis der Erkenntnisse und Algorithmen aus [4] entwickelt.

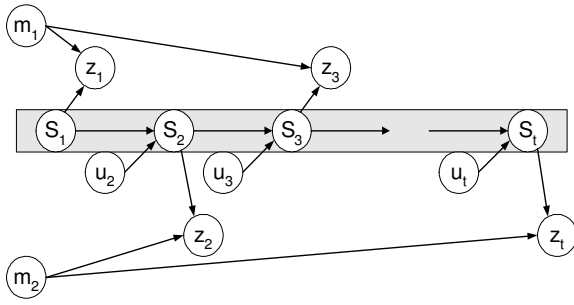


Abbildung 2: Darstellung des SLAM Problem als dynamisches Bayes-Netz. Sind alle Roboterzustände bekannt, so sind die Schätzprobleme der einzelnen Landmarkenpositionen von einander unabhängig (Darstellung entnommen aus [4])

## 2 Theorie und Algorithmus

Alle nachfolgend betrachteten Algorithmen weisen eine wahrscheinlichkeitstheoretische Herangehensweise auf. Deshalb soll das SLAM Problem zunächst in seiner wahrscheinlichkeitstheoretischen Form dargestellt werden. Da die durchgeführten Untersuchungen Roboter verwenden, die sich in der Ebene bewegen, besteht ein Roboterzustand  $s_t$  aus den (x,y)-Koordinaten des Roboters in der Ebene und der Blickrichtung des Roboters. Es wird angenommen, dass die Entwicklung des Roboterzustandes einem Wahrscheinlichkeitsgesetz folgt, dass sich aufgrund der Gültigkeit der Markov-Bedingung durch die Verteilung

$$p(s_t | u_t, s_{t-1}) \quad (1)$$

darstellen lässt. Diese Verteilung repräsentiert das Bewegungsmodell. Sie beschreibt die Wahrscheinlichkeiten mit denen der Roboter bei einer fehlerbehafteten Bewegung  $u_t$  ausgehend vom Zustand  $s_{t-1}$  bestimmte Zustände  $s_t$  erreicht. Das Bewegungsmodell  $p(s_t | u_t, s_{t-1})$  ist gewöhnlich zeitinvariant und stellt eine wahrscheinlichkeitstheoretische Generalisierung der Roboterkinematik dar. Die Sensormessungen werden durch das Beobachtungsmodell beschrieben. Das Beobachtungsmodell ergibt sich entsprechend Abb. 1(a) zu der Verteilung

$$p(z_t | s_t, m_i) \text{ bzw. } p(z_t | s_t, m, i_t) \quad (2)$$

Diese Verteilung repräsentiert die Wahrscheinlichkeiten unterschiedlicher Beobachtungen bei der fehlerbehafteten Messung der Lage der Landmarke  $i_t$ .

Das SLAM Problem, wie es im 2. Absatz des Abschnittes SLAM definiert wurde, besteht nun in der Berechnung der a-posteriori-Wahrscheinlichkeitsverteilung

$$p(s_{1:t}, m | z_{1:t}, u_{1:t}, i_{1:t}) \quad (3)$$

unter Verwendung von Bewegungs- und Beobachtungsmodell.

### Faktorisierung des SLAM Problem

Zur Berechnung dieser a-posteriori-Wahrscheinlichkeitsverteilung wurde in [4] eine Faktorisierung angegeben, die das SLAM Problem aufgrund bestehender Unabhängigkeitseigenschaften in  $M + 1$  einfachere Teilprobleme aufteilt. Die Basis für diese Faktorisierung bildet folgende Eigenschaft von Bayes-Netzen[1]:

**Sind die Zustände aller Eltern einer Variable gegeben, so ist diese Variable von allen Nicht-Nachfolgern im Netz unabhängig.**

Wird nun für das SLAM Problem angenommen, dass alle Roboterzustände bekannt sind (Abb. 2), so werden die Schätzprobleme zur Bestimmung der M Landmarkenpositionen unabhängig und es kann folgende Faktorisierung vorgenommen werden:

$$p(s_{1:t}, m | z_{1:t}, u_{1:t}, i_{1:t}) = p(s_{1:t} | z_{1:t}, u_{1:t}, i_{1:t}) \cdot p(m | s_{1:t}, z_{1:t}, u_{1:t}, i_{1:t}) \quad (4)$$

$$= p(s_{1:t} | z_{1:t}, u_{1:t}, i_{1:t}) \cdot \prod_i p(m_i | s_{1:t}, z_{1:t}, u_{1:t}, i_{1:t}) \quad (5)$$

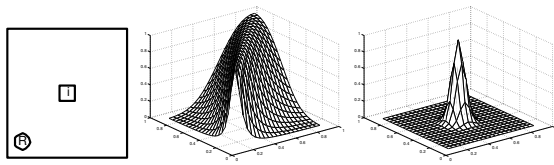


Abbildung 3: *links*: Szenario mit Roboter  $R$  und Landmarke  $i$ . *mitte*: Darstellung des Beobachtungsmodells nur unter Verwendung des Winkels. *rechts*: Darstellung des Beobachtungsmodells unter Verwendung von Winkel und Entfernung.

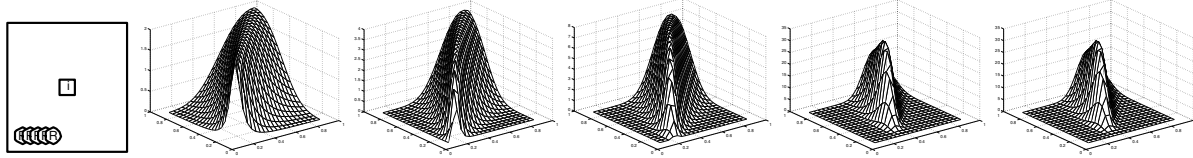


Abbildung 4: *links*: Szenario mit Roboterbewegung und Landmarke  $i$ . *folgend*: Darstellung der a-posteriori-Verteilung der Landmarkenposition zu den Zeitpunkten  $t=1, \dots, 5$

Damit wird das SLAM Problem defacto in ein Problem der Schätzung der a-posteriori-Wahrscheinlichkeit des Roboterpfades  $s_{1:t}$  und  $M$  vom Roboterpfad abhängige Probleme der Schätzung der a-posteriori-Wahrscheinlichkeit der  $M$  Landmarkenpositionen aufgeteilt.

## SLAM mit Hierarchischen Partikelfiltern

Zur Schätzung des Roboterpfades  $s_{1:t}$  verwendet der hier vorgestellte Algorithmus genau wie FastSLAM einen Partikelfilter(PF). Innerhalb jedes Partikels dieses Filters werden nun Landmarkenschätzer für jede einzelne Landmarke benötigt. Dies ist notwendig, da die Schätzungen der Landmarkenpositionen vom Roboterpfad  $s_{1:t}$  abhängig sind und dieser somit jedem Landmarkenschätzer bekannt sein muss. Da jeder einzelne Partikel einen speziellen Roboterpfad darstellt, ist es innerhalb eines solchen Partikels möglich, die  $M$  Landmarkenschätzungen durchzuführen. Jeder Partikel des PF enthält demzufolge also  $M$  Landmarkenschätzer.

Die Realisierung dieser Landmarkenschätzer unterscheidet nun den hier vorgestellten Algorithmus vom FastSLAM Algorithmus aus [4]. Während FastSLAM die Landmarkenschätzer durch je einen Kalman-Filter realisiert, sollen diese hier durch je einen PF implementiert werden. Damit entsteht eine 2-stufige Hierarchie von Partikelfiltern. Ein PF, der sogenannte Makropartikelfilter(MaPF), approximiert die Zeitreihe der Roboterzustände  $s_{1:t}$ , während jeder einzelne Partikel dieses Makropartikelfilters wiederum  $M$  sogenannte Mikropartikelfilter(MiPF) zur Schätzung der  $M$  Landmarkenpositionen besitzt.

Bevor im nächsten Abschnitt wichtige Implementationsdetails, die die Verrechnung von MaPF und MiPF betreffen, unter Verwendung von Rao-Blackwellisierten Partikelfiltern[3] theoretisch erarbeitet werden, soll nun zunächst die Frage erörtert werden, warum für die hier gestellte Aufgabe (SLAM mit einer einzelnen Kamera / SLAM mit reduzierter Beobachtung) die Verwendung von Partikelfiltern als Landmarkenschätzer der Verwendung von Kalman-Filtern vorgezogen werden sollte. Der wichtigste Grund liegt in der Ausprägung (Form) der Wahrscheinlichkeitsverteilung  $p(m_i | s_{1:t}, z_{1:t}, u_{1:t}, i_{1:t})$  die geschätzt werden soll. Kalman-Filter approximieren diese Verteilung unimodal und gaussförmig. Im vorliegenden Fall von Beobachtungen  $z_t$ , die zu einer Landmarke lediglich einen Beobachtungswinkel aber keine Entfernung enthalten, ist die Verteilung  $p(m_i | s_{1:t}, z_{1:t}, u_{1:t})$  für kleine  $t$  allerdings keinesfalls gaussförmig und auch nur schlecht durch eine Gaussverteilung zu approximieren. Dies liegt darin begründet, dass bereits das Beobachtungsmodell (Gl. 2) keine 2-dimensionale Gaussverteilung darstellt (Abb. 3 mitte), wie dies unter Vorhandensein von Landmarkenentfernungen üblicherweise der Fall ist (Abb. 3 rechts). Daraus ergibt sich, dass bei einer geringen Zahl von Beobachtungen der gleichen Landmarke, wenn diese Beobachtungen zudem alle aus einer ähnlichen Richtung gemacht wurden, die Verteilung  $p(m_i | s_{1:t}, z_{1:t}, u_{1:t})$  strukturelle Unterschiede zu einer Gaussverteilung

aufweist (Abb. 4). Zur Approximation solcher Verteilungen ist der Kalman-Filter nicht gut geeignet und wäre somit gerade zum schwierigen Beginn des Kartenaufbaus stark fehlerbehaftet. PF können dagegen Verteilungen beliebigen Charakters approximieren und sind deshalb bei den auftretenden ungleichförmigen Verteilungen vermutlich vorteilhafter.

## Rao-Blackwellisierte Partikelfilter

Rao-Blackwellisierte Partikelfilter (RBPF) behandeln das Problem der Verwendung von PF in großen Zustandsräumen unbekannter Variablen. Sie verwenden Abhängigkeiten innerhalb der gesuchten Variablen um die Dimension der zu erzeugenden Verteilung (und damit die Dimension des zu durchsuchenden Zustandsraumes) zu reduzieren, während die nicht im reduzierten Zustandsraum vorhandenen Variablen unter Zuhilfenahme der bestehenden Abhängigkeiten analytisch behandelt werden.

Die Möglichkeit der Anwendung dieses Verfahrens besteht[3], sobald sich die unbekannt Variablen  $x$  so in zwei Gruppen  $x^A$  und  $x^B$  teilen lassen, dass gilt

$$p(x_t|x_{t-1}) = p(x_t^B|x_{t,t-1}^A, x_{t-1}^B)p(x_t^A|x_{t-1}^A) \quad (6)$$

Desweiteren muss die a-posteriori-Verteilung der Variablen  $x_{1:t}^B$  in Abhängigkeit von den Beobachtungen  $z_{1:t}$  und den Variablen  $x_{1:t}^A$  analytisch handhabbar sein:

$$p(x_{1:t}^B|z_{1:t}, x_{1:t}^A) \quad (7)$$

Diese Bedingungen sind für das SLAM Problem gegeben. Die unbekannt Variablen sind in zwei Gruppen  $s_t$  und  $m$  teilbar, die die Bedingung (6) erfüllen. Die Handhabung der Wahrscheinlichkeit  $p(m|z_{1:t}, s_{1:t})$  wird mittels der Mikropartikelfilter geeignet realisiert (Details folgen). Die Implementation des MaPF in Form eines RBPF entsprechend [3] ist demzufolge möglich.

Die Vorteile der Verwendung eines RBPF (und damit der Verwendung Hierarchischer PF) gegenüber der Anwendung eines PF auf dem gesamten Zustandsvektor  $(s_t, m)^T$  liegen in der Reduzierung der durch Partikel des MaPF zu repräsentierenden Verteilung auf die Dimension von  $s_t$ . Ein detaillierter Nachweis dafür, dass RBPF mit weniger Partikeln die gleiche Präzision wie normale PF erreichen ist in [3] nachzulesen.

## Makropartikelfilter

Der MaPF ist ein RBPF, der den üblichen Algorithmus für Partikelfilter verwendet (Abb. 5). Beim Bewegungsupdate werden die Partikel entsprechend des Bewegungsmodells

$$q = p(s_t|s_{t-1}, u_t) \quad (8)$$

verändert. Zur Vereinfachung des Algorithmus wird also die Beobachtung  $z_t$  zum Zeitpunkt  $t$  vernachlässigt<sup>1</sup>. Aufgrund dieser Sample-Verteilung ergibt sich das Gewicht für jeden Makropartikel  $a$  zu

$$w_t^{(a)} = p(z_t|z_{1:t-1}, s_{1:t}) \quad (9)$$

Da eine direkte Berechnung nicht möglich ist, wird das Beobachtungsmodell verwendet und es ergibt sich das Integral

$$w_t^{(a)} = \int p(z_t|m_i, z_{1:t-1}, s_{1:t})p(m_i|z_{1:t-1}, s_{1:t})dm_i \quad (10)$$

zu dessen Bestimmung der entsprechende Mikropartikelfilter des zu berechnenden Makropartikels verwendet wird. Da die Partikel dieses MiPF entsprechend  $p(m_i|z_{1:t-1}, s_{1:t})$  verteilt sind, stellt folgende Summe eine gute Approximation des Integrals dar:

$$w_t^{(a)} = \frac{1}{N_{MiPF^i}} \sum_{b \in MiPF^i} p(z_t|\hat{s}_t, \hat{m}_i = position(b)) \quad (11)$$

<sup>1</sup>Ein optimaler Algorithmus müsste die Partikel entsprechend  $p(s_t|s_{t-1}, u_t, z_t)$  verändern[3].

1. **Bewegungsupdate:**

Verändere jeden Partikel  $a$  entsprechend der Sample-Verteilung  $q$

$$x_t^{(a)} = q(x_t | x_{1:t-1}^{(a)}, z_{1:t})$$

2. **Beobachtungsupdate:**

Bestimme das Gewicht  $w_t$  jedes Partikel  $a$  zu

$$w_t^{(a)} = \frac{p(x_{1:t}^{(a)} | z_{1:t})}{q(x_t^{(a)} | x_{1:t-1}^{(a)}, z_{1:t}) p(x_{1:t-1}^{(a)} | z_{1:t-1})}$$

3. **Resampling** des Partikelfilters entsprechend der berechneten Gewichte  $w_t$

Abbildung 5: *Algorithmus zur Berechnung eines Partikelfilters*

Dabei sei  $position(b)$  die Landmarkenposition  $m_i^b$ , die der Mikropartikel  $b$  des MiPF darstellt. Damit sind alle für die Berechnung des Makropartikelfilters notwendigen Parameter und Algorithmen beschrieben.

### Mikropartikelfilter

Jeder MiPF ist ein einfacher PF, der die Wahrscheinlichkeitsverteilung der Position einer Landmarke approximieren soll. Auch die MiPF realisieren lediglich ein einfaches Sampling entsprechend

$$q = p(m_t | m_{t-1}) \quad (12)$$

Daraus ergibt sich das Gewicht des Mikropartikel  $b$  zu

$$w_t^{(b)} = p(z_t | m_i) = p(z_t | m_i, s_t) \quad (13)$$

Dieses lässt sich im Gegensatz zum Gewicht des MaPF direkt mittels des Beobachtungsmodells berechnen, da jeder MiPF zu genau einem Partikel des MaPF gehört und somit unter der Bedingung des im Partikel festgelegten Roboterpfades  $s_{1:t}$  berechnet wird.

## 3 Praktische Aspekte

### Sample-Verteilung der Mikropartikelfilter

Die Sample-Verteilung  $p(m_t | m_{t-1})$  eines MiPF entspricht eigentlich dem Bewegungsmodell der Landmarke, die der MiPF approximiert. Da die Landmarken unbeweglich und daher auch ohne Bewegungsfehler sein sollen, müsste diese Verteilung einen Dirac-Impuls an der Position  $m_t = m_{t-1}$  darstellen. Eine solche Sample-Verteilung würde zwar dem eigentlichen Bewegungsmodell der Landmarke entsprechen, sie besitzt allerdings auch den Nachteil, dass der MiPF in seiner gesamten Laufzeit nur an den Positionen im Zustandsraum Partikel ausprägen kann, an denen er bei der Initialisierung bereits Partikel erzeugt hat. In Konsequenz dessen wäre die Approximation des MiPF entweder sehr ungenau, weil nur wenige Positionen des kontinuierlichen Zustandsraumes mit Partikeln belegt sind, oder die Approximation würde ungewöhnlich viele Partikel benötigen, um im Zustandsraum bereits zur Initialisierung möglichst viele Positionen für den MiPF verfügbar zu machen. Um diesen Effekt zu vermeiden, wird dem Bewegungsmodell der Landmarke ein künstlicher Bewegungsfehler in

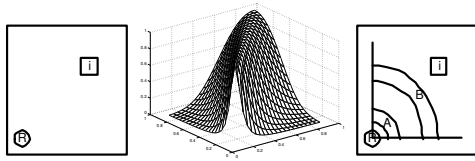


Abbildung 6: *links*: Szenario mit Roboter  $R$  und Landmarke  $i$ . *mitte*: Darstellung des Beobachtungsmodells. *rechts*: Die Sektoren  $A$  und  $B$  sollten nach der ersten Beobachtung der Landmarke  $i$  gleichwahrscheinlich sein. Das bedeutet, dass sie die gleiche Anzahl Partikel enthalten sollten.

Form einer Gaussverteilung mit Erwartungswert 0 hinzugefügt. Dieser macht die Partikel des MiPF beweglich ohne die Gesamtverteilung des PF beim Bewegungsupdate zu verändern. Der Filter wird damit flexibler. Intuitiv wird die Approximation durch die Beweglichkeit der Partikel genauer. Es muss lediglich darauf geachtet werden, dass die Bewegungsmöglichkeit der Partikel nicht derart groß gewählt wird, dass eine konvergierte Partikelwolke aufgrund der Beweglichkeit der einzelnen Partikel bei jedem Bewegungsupdate erneut zerfällt.

## Initialisierung der Partikelfilter

Zur Initialisierung der MiPF besteht zunächst die Möglichkeit, die Partikel gleichverteilt in der Ebene zu erzeugen. Diese Methode erweist sich auf den zweiten Blick jedoch als ungeeignet. Grund dafür sind die verwendeten Beobachtungen, die nur aus einem Beobachtungswinkel bestehen. Das Beobachtungsmodell ist daher nur über diesem Winkel gaussförmig. Für die Entfernung der beobachteten Landmarke vom Roboter wird eine Gleichverteilung angenommen (Abb. 6 mitte). Dies führt zu folgender Problematik der a-posteriori-Verteilung  $p(m_i | s_1, z_1)$  einer genau einmal beobachteten Landmarke  $i$ : Da die Verteilung der Partikel des MiPF die approximierte Verteilung der Landmarkenposition darstellt, und diese Verteilung in der Entfernung vom Roboter gleichverteilt ist, sollten in den Sektoren  $A$  und  $B$  der Abb. 6 nach genau einer Beobachtung jeweils gleich viele Partikel des MiPF vorhanden sein. Wurde der MiPF allerdings mit über der Ebene gleichverteilten Partikeln initialisiert, so sind nach der Initialisierung in Sektor  $B$  aufgrund der größeren Fläche mehr Partikel als in Sektor  $A$ . Wenn die Beobachtung aber lediglich aus einem Winkel besteht, bleibt das Verhältnis der Partikelzahlen in Sektor  $A$  und  $B$  bei Durchführung der 1. Beobachtung konstant (Die Verteilung der Partikel über die Entfernung vom Roboter bleibt erhalten). Demzufolge liegt bei Initialisierung des MiPF durch Gleichverteilung der Partikel in der Ebene nach der 1. Beobachtung einer Landmarke nicht die notwendige Verteilung der Partikel vor.

Die Initialisierung eines MiPF erfolgt aus diesem Grund erst zum Zeitpunkt der 1. Beobachtung einer Landmarke. Die Partikel werden dann entsprechend der benötigten Verteilung so initialisiert, dass ihre Entfernung vom Roboter gleichverteilt und die Abweichung vom beobachteten Winkel gaussverteilt ist.

Die Initialisierung des Makropartikelfilters erfolgt durch die Vorgabe der globalen Startposition  $s_1$  des Roboters. Ist diese unbekannt, so wird der Kartenaufbau am Koordinatenursprung begonnen. Die entstehende Karte sowie der berechnete Roboterpfad sind dann als Koordinaten relativ zur Startposition des Roboters zu interpretieren.

## 4 Ergebnisse

### Kartenaufbau

In praktischen Untersuchungen in einem einfachen Szenario konnte die Funktionsfähigkeit des Algorithmus gezeigt werden. In Abb. 7 ist zur Darstellung der Ergebnisse die Entwicklung der Mikropartikelfilter eines ausgewählten Makropartikels aufgeführt (Alle anderen Makropartikel wiesen gleichartige Eigenschaften auf.). Dargestellt sind die Zustände nach der Ausführung von  $k = 1, 2, 8, 12, 18$



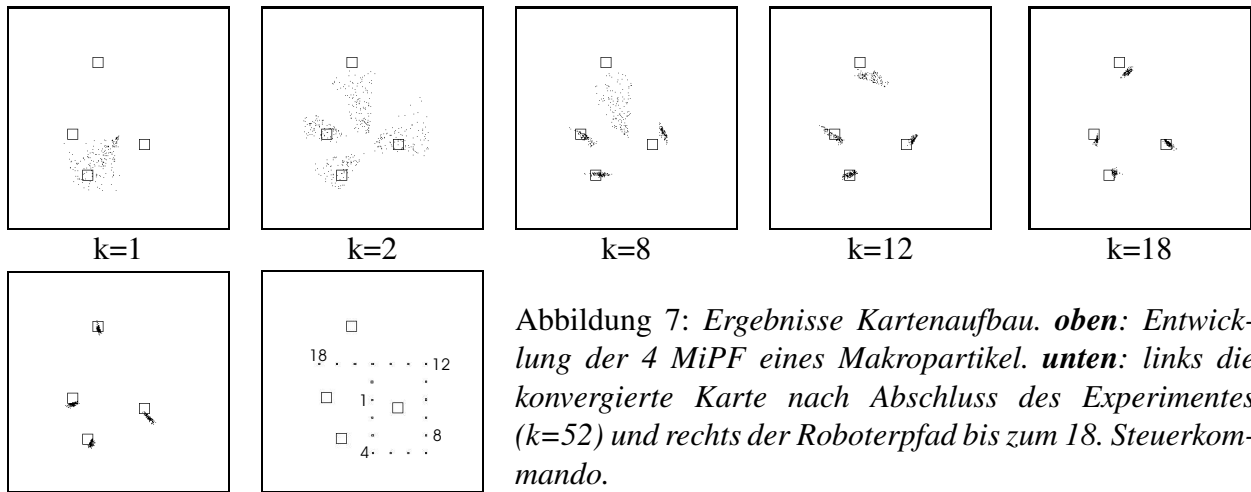


Abbildung 7: Ergebnisse Kartenaufbau. **oben:** Entwicklung der 4 MiPF eines Makropartikel. **unten:** links die konvergierte Karte nach Abschluss des Experimentes ( $k=52$ ) und rechts der Roboterpfad bis zum 18. Steuerkommando.

Steuerkommandos. Nach jeder Ausführung eines Steuerkommandos wurden alle vom Landmarkendetektor gefundenen Landmarken einzeln und nacheinander beobachtet. Dies entspricht der Funktionsweise des Algorithmus, der zu jedem Zeitpunkt genau eine beobachtete Landmarke verarbeiten kann.

Für  $k = 1$  wurde zur besseren Erkennlichkeit lediglich einer der 4 MiPF dargestellt. Diese Abbildung zeigt den Zustand eines MiPF nach der Initialisierung durch die erste Beobachtung einer Landmarke. Die Partikelwolken, die anfangs die Form eines Kreissektors aufweisen und die höchste Partikeldichte entlang der Winkelhalbierenden des Sektors besitzen, verkleinern sich durch jede weitere Beobachtung einer Landmarke (Abb. 7,  $k = 2$ ). Bereits nach wenigen Schritten sind die MiPF der mehrfach beobachteten Landmarken in der unteren Szenariohälfte zu einer gut approximierenden Partikelwolke zusammengeschrumpft. Im weiteren Verlauf (Abb. 7,  $k = 12, 18$ ) zeigt sich dann, dass auch die Verteilungen von anfangs selten beobachteten Landmarken konvergieren sobald der Roboter deren Umfeld erreicht und diese mehrfach beobachten kann.

Nach Abschluss des Experimentes (Abb. 7, unten linkes Bild) weist der dargestellte Makropartikel, genau wie alle anderen Makropartikel, eine konvergierte Karte der Umwelt auf. Die Ergebniskarte, die sich durch Verrechnung der einzelnen Makropartikel ergibt, stellt eine gute Approximation der Umwelt dar.

## Lokalisation

Ein funktionsfähiger SLAM Algorithmus besitzt neben einem korrekten Kartenaufbau auch die Fähigkeit zur korrekten Lokalisation des Roboters. Da aufgrund des relativ einfachen Szenario und eines zudem noch relativ geringen Bewegungsfehlers des verwendeten Khepera-Roboters in praktischen Versuchen keine größeren Fehllokalisationen auftraten, wurde die Lokalisationsfähigkeit des Algorithmus durch mehrmaliges Präsentieren der Daten eines geschlossenen Rundkurses getestet. Das bedeutet, dass der Roboter in Realität zu Beginn und Ende des Rundkurses einen identischen Zustand  $s_1 = s_T$  aufwies. Der Fehler, den das exakte Aufrechnen der Steuerkommandos  $u_{1:T}$  ergab, war entsprechend des geringen Bewegungsfehlers des Khepera nach dem 1. Durchlauf relativ gering (Abb. 8, 1. Darstellung). Da sich dieser Fehler nach mehrmaligem Durchlauf der Daten allerdings akkumuliert, ist bereits beim 5. Durchlauf eine deutliche Fehllokalisation des Roboters zu erkennen (Abb. 8, 2. Darstellung).

Unter Verwendung des vorgestellten SLAM Algorithmus ergab sich dann folgendes Ergebnis. Beim 1. Durchlauf wurde der Roboter zwar ausreichend gut, allerdings schlechter als beim exakten Aufrechnen der Steuerkommandos lokalisiert (Abb. 8, 3. Darstellung). Dies ist vermutlich der zu Beginn sehr ungenauen Karte zu verschulden. Bei den weiteren Durchläufen zeigte sich allerdings

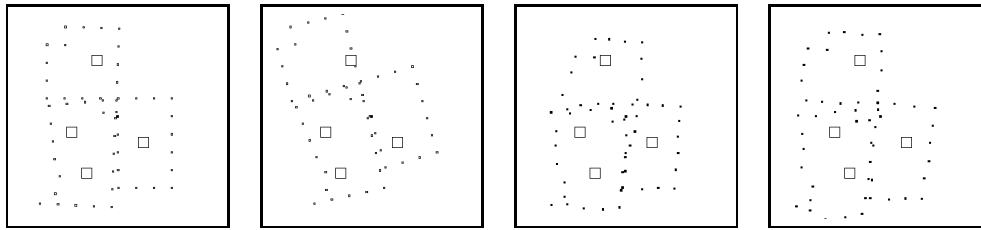


Abbildung 8: Ergebnisse Lokalisation. **von links**: Roboterpfad anhand Verrechnen der Steuerkommandos (1. und 5. Durchlauf eines Rundkurses)). Roboterpfad laut SLAM mit Hierarchischen PF (analog 1. und 5. Durchlauf).

kein Akkumulieren der Fehler. Der Algorithmus schaffte es, den Roboter weiterhin im Rahmen eines begrenzten Fehlers korrekt zu lokalisieren (Abb. 8, 4. Darstellung). Dies zeigt die Lokalisationsfähigkeit des vorgestellten Algorithmus, auch wenn diese natürlich in komplizierteren Szenarios noch ausführlicher untersucht werden muss.

## 5 Ausblick

Nachdem sich der vorliegende Artikel zunächst auf die Entwicklung und die prinzipielle Funktionstüchtigkeit des Algorithmus beschränkt, sollen weitere Untersuchungen sein Verhalten und die Qualität der Ergebnisse des Algorithmus in komplizierteren Szenarios testen. Später sind interessante Weiterentwicklungen zum Beispiel im Bereich multimodaler Landmarkenverteilungen möglich. Diese können sich etwa aufgrund nicht eindeutig detektierbarer Landmarken ergeben. Da der Algorithmus zur Schätzung der Landmarkenpositionen Partikelfilter verwendet, ist er prinzipiell zur Verarbeitung multimodaler Landmarkenverteilungen in der Lage. Weitere Untersuchungen müssten diese Behauptung allerdings im Detail bestätigen. Für derartige Untersuchungen müsste dann u.a. auf die hier verwendete Voraussetzung eindeutig identifizierbarer Landmarken verzichtet werden.

## Literatur

- [1] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *Proc. Conf. Uncertainty in Artificial Intelligence (UAI)*, 1996.
- [2] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.
- [3] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proc. Conf. Uncertainty in Artificial Intelligence (UAI)*, 2000.
- [4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.
- [5] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*. Springer, 1990.