

LOKALISATION MOBILER ROBOTER MIT ODOMETRIE UND BILDVERARBEITUNG IN EINER THEATERINSTALLATION

M A S T E R A R B E I T

eingereicht am

03.09.2013

bei

Prof. Dr.-Ing. Udo Frese

Prof. Dr.-Ing. X Y

Universität Bremen

von

Josef F. Hiller

Matr. Nr: 2055491

Osterstr 79

28199 Bremen

Zusammenfassung

Schlagwörter: abc, def, xyz

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig angefertigt und mich fremder Hilfe nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß veröffentlichtem oder unveröffentlichtem Schrifttum entnommen sind, habe ich als solche kenntlich gemacht.

Bremen, xx.09.2013

Josef F. Hiller

Inhaltsverzeichnis

Zusammenfassung	iii
Erklärung	v
Abbildungsverzeichnis	ix
Tabellenverzeichnis	xi
1 Einleitung	1
1.1 Ziel und Aufgabenstellung der Arbeit	1
1.2 Gliederung	3
2 Grundlagen	5
3 Software	9
3.1 Simulation	9
3.1.1 Die Szene	9
3.1.2 Messen in der Simulation	11
3.2 Lokalisation	12
4 Versuchsdurchführung	13
4.1 Theoretische Vorüberlegungen	13
4.2 Versuchsaufbau	13
4.3 Versuchsvorbereitung	13
4.4 Praktische Versuchsdurchführung	13
5 Versuchsergebnisse	15
6 Diskussion der Messergebnisse	17
7 Weitere Problemstellungen / Ausblick	19

8	Fazit	21
A	Abkürzungsverzeichnis	I
B	Literaturverzeichnis	III

Abbildungsverzeichnis

1.1	Roboter und Besucher auf der EPKOT	2
1.2	Roboter vor Lichtwand auf der EPKOT	2
1.3	@@ PLATZHALTER @@	3
3.1	Bild Simulierte Szene	10

Tabellenverzeichnis

2.1	Anzahl Probanden pro Gruppe bei $\alpha = 0,05$	7
-----	---	---

1 Einleitung

In dieser Arbeit soll, mit Hilfe einer Simulation, untersucht werden, wie gut sich Kameras bei der Standortbestimmung in mobilen Roboter einsetzen lassen. Anlass dieser Arbeit war eine Anfrage der Künstlergruppe Beobachter der Bediener von Maschinen (BBM), die schon bei mehreren Ausstellungen/Performances¹ mobile Roboter eingesetzt haben. Dabei interessierten sie sich für eine Lokalisierungs-lösung welche möglichst ohne weitere spezial Hardware und geringem Installationsaufwand vor Ort auskommt. Der Ansatz der daraus entstand war: die Kameras, die bereits an jedem der Roboter verbaut waren, zu nutzen um markante Muster in der Bühneninstallation zu erkennen und gemeinsam mit den Odometrie-Daten zur Positions-berechnung zu verwenden. Dabei sollte ein Partikelfilter als Zustandsschätzer verwendet werden. Zu der Bühneninstallation gehören große Lichtwände, zu sehen auf Abbildung 1.1 und 1.2. Auf diese Lichtwände könnte ein hell/dunkel Bit-Muster angezeigt werden das es mit Hilfe geeigneter Bildverarbeitungsalgorithmen und mit den Kameras zu erkennen gilt.

1.1 Ziel und Aufgabenstellung der Arbeit

Im Rahmen dieser Arbeit soll eine Simulationsumgebung mit Hilfe geeigneter 3D-Visualisierungs Bibliotheken erstellt werden. Die in der Lage ist eine 3D-Szene der Bühne zu simulieren und Kamerabilder sowie Odometrie-Daten einer Roboterfahrt zu erzeugen. Außerdem soll ein Lokalisationsalgorithmus entwickelt werden, der auf Grundlage dieser Daten die Position und Orientierung des Roboters auf der Bühne schätzen kann. Anschließend soll die Qualität dieser geschätzten Position beurteilt werden und mögliche Fehlerquellen diskutiert werden.

¹unter Anderem:

2000 Themenpark "Wissen" der Expo 2000 Hannover

2010 Joybots in der BMW-Welt

2012 EPKOT Experimental Prototype Killers of Tomorrow , Hannover

siehe auch <http://www.bbm.cfnt3.de>



Abbildung 1.1: Roboter und Besucher auf der EPKOT Quelle: <http://www.bbm.cfnt3.de>

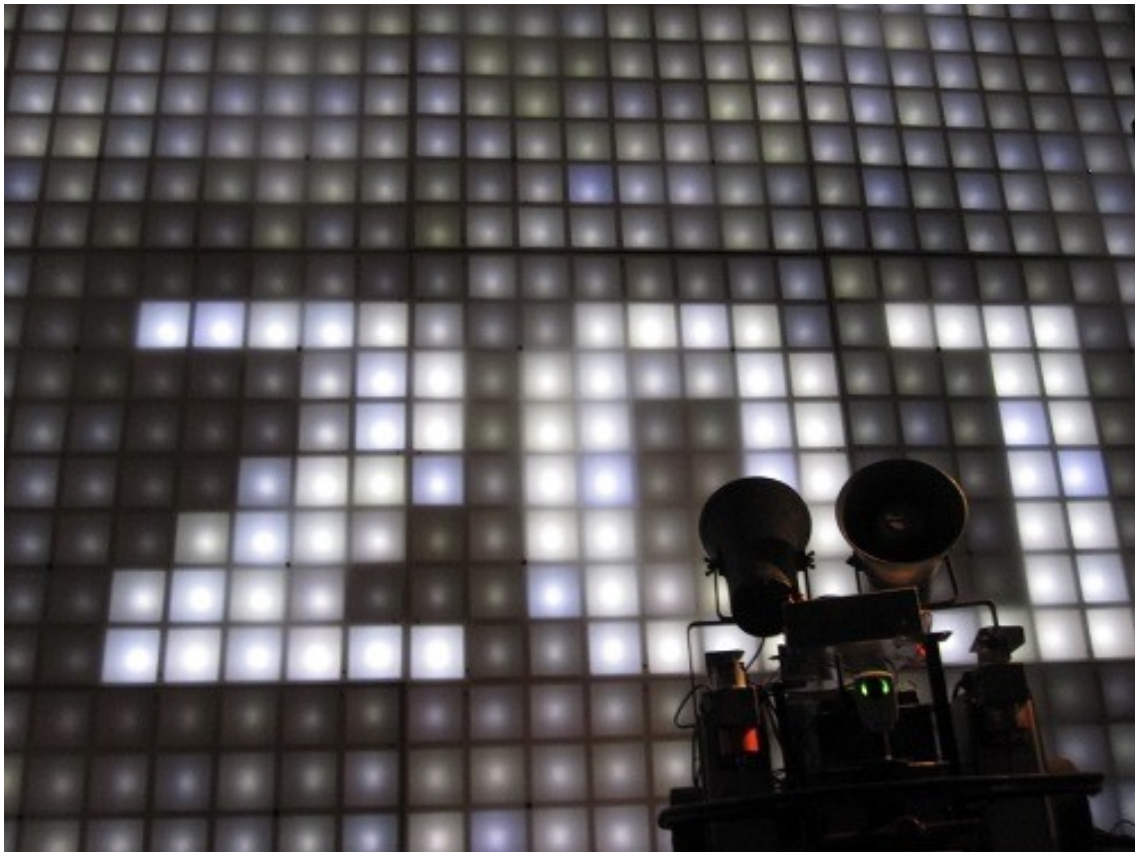


Abbildung 1.2: Roboter vor Lichtwand auf der EPKOT Quelle: <http://www.bbm.cfnt3.de>

1.2 Gliederung

Die Arbeit wird im folgenden Kapitel eine Übersicht über gängige Lokalisationsverfahren sowie ihre Vor- und Nachteile geben. Anschließend wird die Simulations- und Lokalisationssoftware vorgestellt und deren Implementation erklärt und begründet. In Kapitel 4 beginnt die Beschreibung verschiedene Versuche die zum Beurteilen der Lokalisationsergebnisse durchgeführt werden sollen. Danach werden die Ergebnisse dieser Versuche vorgestellt und in Kapitel 6 Diskutiert. Abschließend wird ein Ausblick zur möglichen Anwendung dieses Verfahrens gegeben und ein Fazit gezogen.

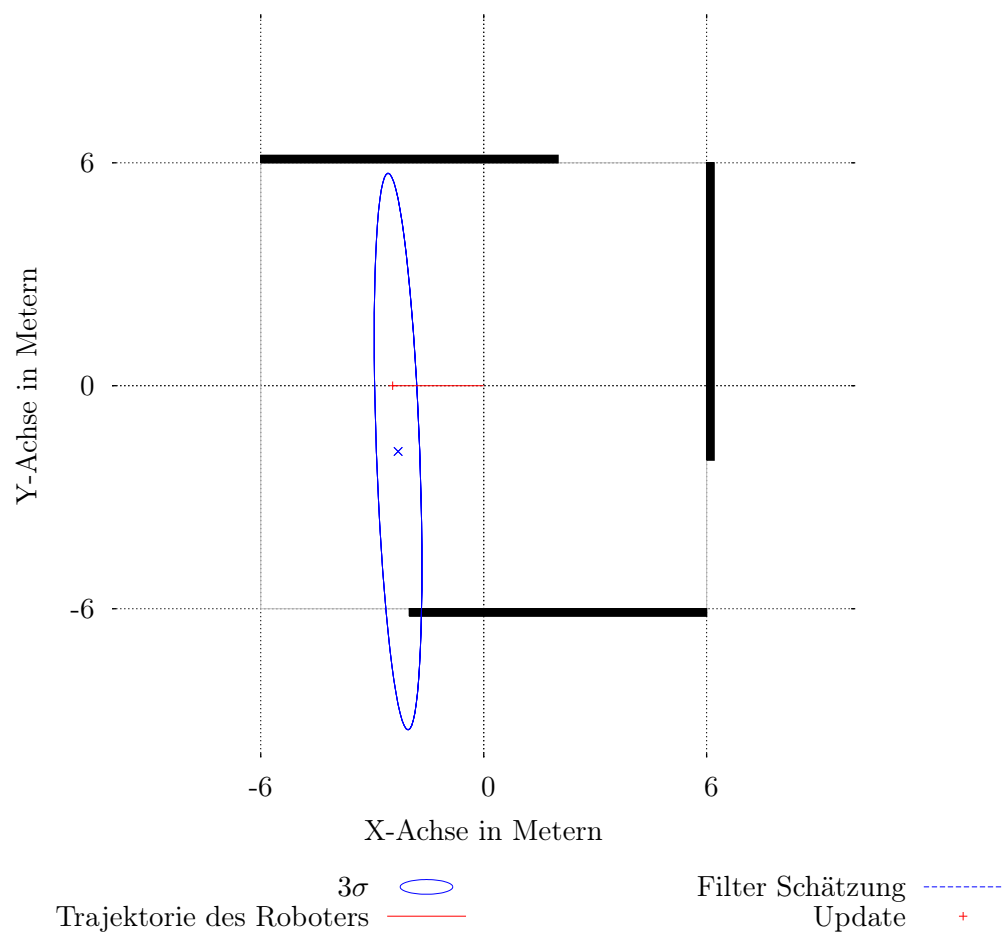


Abbildung 1.3: @@ PLATZHALTER @@

2 Grundlagen

Die Lokalisation ist eines der Grundprobleme das beim Einsatz von mobilen Robotern auftritt. Die meisten Lösungsansätze verwenden im Grunde eine Variante des Bayes-Filters wie z.B. in [Thrun et al., 2006, Seite 26] beschrieben. Abhängig vom Einsatzumfeld und der verwendeten Hardware gibt es aber noch Andere Verfahren. In [Seco et al., 2009] zum Beispiel werden, neben dem Bayes-Filter, drei weitere vorgestellt:

- Geometry-Based Methods
- Minimization of the cost function
- Fingerprint Methods

die sich in Gebäuden unter Nutzung von Elektromagnetischen Signale verwenden lassen. Je nach Sensorausstattung werden verschiedene Karten der Umgebung verwendet. Auf die sich der Roboter durch Auswertung von Sensormessungen lokalisieren muss. Dabei gibt es Karten die vorher angefertigt wurden, und dem Roboter bereits zur Verfügung stehen. Und es gibt Karten die der Roboter selbstständig während der Lokalisation erstellen muss. Letzteres wird als Simultaneous Localization and Mapping (SLAM) Problem bezeichnet zu dem es bereits viele Veröffentlichungen gibt (u.a.) [Thrun et al., 2006, Seite 309], [Thrun, 2002], [Hertzberg et al., 2012, Seite 229]

Der in Kapitel 1 genannte Partikel Filter ist eine Variante des Bayes-Filters. Weitere Verfahren findet man unter folgenden Stichwörtern: KALMAN FILTER (EKF, UKF), PARTIKEL FILTER oder SEQUENZIELLE MONTE-CARLO-METHODE, GRID-BASED-FILTER oder HISTOGRAM FILTER (CONTINUOUS SPACE), DISCRETE BAYES FILTER (DISCRETE SPACE)

In [Thrun et al., 2006, Seite 193] wird die Lokalisation in drei Teilprobleme zerlegt:

Position Tracking bezeichnet den Vorgang, bei dem mit bekannter Ausgangspo-

sition diese mit Hilfe von Sensordaten bei Bewegungen verfolgt werden kann. Dabei spielt das Dynamikmodell des Roboters sowie darin modellierte Unsicherheiten eine wichtige Rolle. Denn bewegt sich der Roboter von einer bekannten Position aus, wird mit dem Dynamikmodell seine neue Position geschätzt. Bekannte Unsicherheiten des Modells erzeugen eine Wahrscheinlichkeitsverteilung um diese neue Position, in der sich die wahre Position befinden sollte. Ohne Messungen von weiteren Sensoren die Rückschlüsse auf die Umgebung erlauben, würde die Positionsschätzung mit der Zeit immer ungenauer. Mit Hilfe eines Messmodells lässt sich beurteilen, ob eine Messung an einer bestimmten Position wahrscheinlich erscheint, oder nicht. Dadurch lässt sich die Wahrscheinlichkeitsverteilung der Position nach einer Bewegung durch eine Messung wieder eingrenzen. Auf den Bereich, in dem der Messwert des Sensors am Wahrscheinlichsten ist.

Global Localization ist das Finden der Anfangsposition des Roboters unter allen Möglichen Posen die im Szenario vorkommen können. Im Vergleich zum *Position Tracking*, bei dem es genügt die Unsicherheit um die geschätzte neue Pose zu berücksichtigen, umfasst hier der Raum möglicher Posen ein erheblich größeres Volumen. Ein Ansatz wäre alle möglichen Posen mit der selben Wahrscheinlichkeit anzunehmen, und mit den ersten Messungen und dem Messmodell diese einzugrenzen. Auf Bereiche in denen diese Messungen mit hoher Wahrscheinlichkeit auftritt.

Kidnapped Robot Problem ist eine verschärfte Variante des *Global Localization* Problems. Dabei geht man davon aus, dass sich der Roboter spontan an einem anderen Ort aufhält als vom Roboter angenommen. Nun müsste die Lokalisation des Roboters wieder im gesamten möglichen Raum erfolgen. Nur das der Roboter diesen Zustand nicht feststellen kann.

[Thrun, 2002]

Tabelle 2.1: Anzahl Probanden *pro Gruppe* nach Power und Effektstärke bei $\alpha = 0,05$

$\alpha = 0.05$										
Power	Effektstärke									
	0.10	0.15	0.20	0.25	0.30	0.40	0.50	0.60	0.70	0.80
.60	977	434	244	156	109	61	39	27	20	15
.70	1230	547	308	197	137	77	49	34	25	19
.80	1568	697	392	251	174	98	63	44	32	25
.90	2100	933	525	336	233	131	84	58	43	33
.95	2592	1152	648	415	288	162	104	72	53	41
.99	3680	1636	920	589	409	230	147	102	75	58

3 Software

Die Software die im Rahmen dieser Arbeit entwickelt wurde, ist in C++ geschrieben. Als Entwicklungsumgebung wurde Eclipse mit den C/C++ Development Tools (CDT) auf einem Linux¹ Betriebssystem verwendet.

3.1 Simulation

Grundlage der Simulation ist das 3D Grafiktoolkit Open Scene Graph (OCG)². Damit lässt sich eine 3D Szene in Form eines Graphen aufbauen und mit einem Viewer darstellen. Um den Ablauf kontrollieren zu können, lässt sich die Render-Schleife manuell aufrufen um jeden Frame einzeln berechnen zu lassen. Dies wurde als Simulationsschritt gewählt in dem alle nötigen Berechnungen durchgeführt werden können. Da die Geschwindigkeit mit der die Simulation im manuellen Modus abläuft nicht begrenzt wird, wurde eine Mindestbearbeitungszeit integriert. Denn die Geschwindigkeiten von Objekten in der Simulation, wie dem Roboter, werden durch eine zurückgelegte Strecke pro Simulationsschritt festgelegt. Bei sehr schneller Hardware ergab dies eine zu hohe Bewegungsrate um den Roboter noch manuell steuern zu können. Für automatisierte Simulationsläufe mit festgelegten Fahrprofilen könnte man diese Begrenzung wieder lösen um Zeit zu sparen.

3.1.1 Die Szene

Die Szene in der Simulation ist aus mehreren Komponenten aufgebaut, die im Folgenden näher beschrieben werden. Dabei wird ein Vergleich zu den echten Elementen auf der Bühne gezogen, um zu erläutern wie deren Attribute in der Simulation abgebildet werden können. In Abbildung 3.1 kann man alle Elemente der Szene erkennen.

¹Ubuntu 12.04 LTS

²<http://www.openscenegraph.org/>

Die Grundfläche (1) der Bühne misst 12 x 12 m. Sie wird als einfach weiße Fläche in der Szene dargestellt. Da die Bildverarbeitung nur auf den Oberen Teil des Bildes beschränkt ist, spielen Farbe und Helligkeit keine Rolle bei der Erkennung des Musters im Bild.

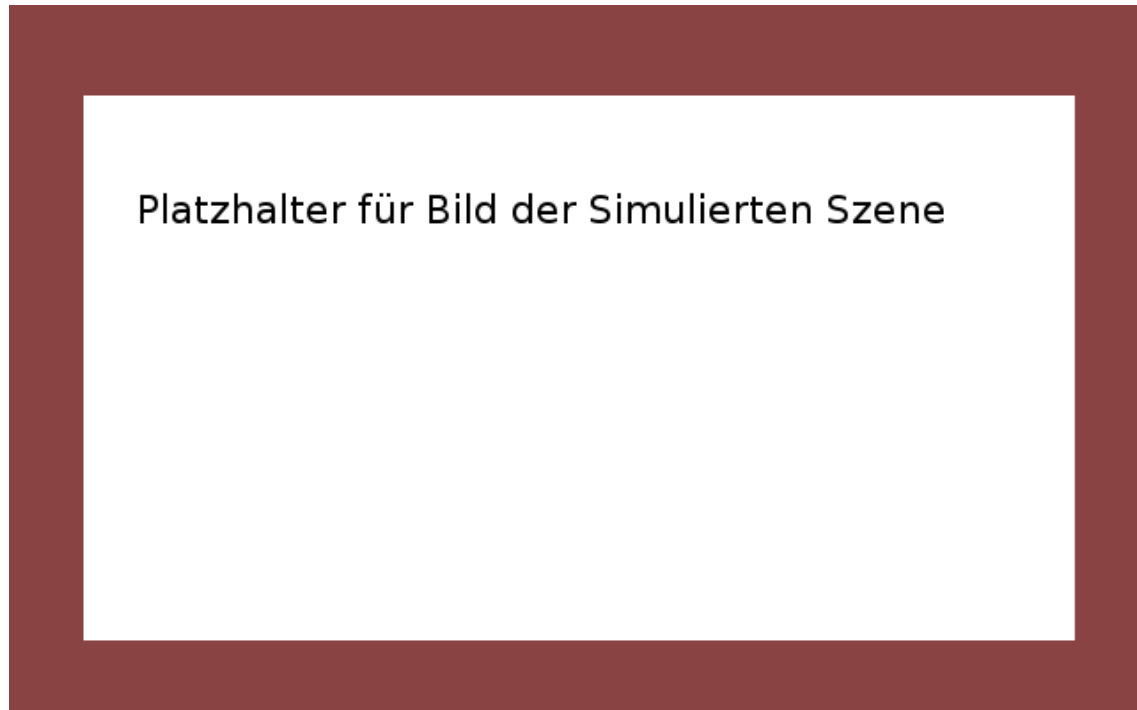


Abbildung 3.1: Bild Simulierte Szene

Die Lichtwände (2,3) sind zu drei Seiten der Grundfläche aufgestellt. Sie messen 3m in der Höhe und 8m in der Länge. Wie bereits in @@@@ beschrieben soll das Muster in der obersten Zeile der Lichtwände dargestellt werden. Hierzu können verschiedene Texturen geladen werden die das Bitmuster enthalten. In der Realität sind die Lichtwände auf der Bühne aus 1 x 1 Meter großen Segmenten zusammen gesetzt, dies wird in der Simulation nicht abgebildet. Es wäre jedoch denkbar, dies in den Aufbau der Texturen einfließen zu lassen. Also leichte Abstands Änderungen der Bits in der Textur. Auf jedem Segment sind 8 x 8 Bits unter gebracht. Das macht eine Kantenlänge von 125 mm bei jedem dieser Pixel und 64 Pixel über die ganze Länge einer Wand. Ein solches Segment besteht aus einer Milchigen Plexiglasplatte, die auf eine Struktur geschraubt wurde die für jedes Pixel ein Leuchtmittel vorsieht. Dessen Helligkeit lässt sich einstellen, wäre aber für das Bitmuster auf die Zustände: dunkel(ausgeschaltet) oder hell(ein, mit größter Helligkeit) einzustellen. Die Plexiglasplatte wird also pro Pixel von Hinten durchleuchtet. Dies führt dazu, das die Helligkeitsverteilung in einem beleuchteten Pixle inhomogen ist. In der Mitte ist die größte Helligkeit, während sie radial nach Außen etwas

abnimmt. Die Ecken der Pixle sind die dunkelsten Stellen. Auf Abbildung 1.2 auf Seite 2 ist dieser Effekt gut zu sehen. Er wird besonders stark, wenn die Leuchtmittel mit niedriger Helligkeit betrieben werden. Bei hoher Helligkeit ist der Effekt noch wahrnehmbar, aber nicht mehr so ausgeprägt. Und noch etwas fällt auf wenn man dieses Bild betrachtet, die Lichtfarbe und Helligkeit variiert leicht von Pixel zu Pixel. All diese Effekte können in der Simulation nur durch verändern der Texturen abgebildet werden.

Der Roboter (4) @@@@@@@@@@@@@@ Platzhalter @@@@@@@@@@@@@@ Er lässt sich mit den Tasten W, A, S, D (vorwärts, links, rückwärts, rechts) grob verfahren oder mit einem Gamepad mit Analogsticks auch präziser steuern. Die Steuerbefehle für den Roboter werden in Geschwindigkeit (gerade aus) und Drehrate interpretiert. Diese führen dann, in jedem Simulationsschritt zu einer Positionsänderung in Richtung der Orientierung und einer anschließenden Änderung der Orientierung um die Drehrate. <—(Implementierung möglicherweise noch ändern???) . Aus diesen Positions- und Orientierungsänderungen werden in einem Weiteren Schritt Drehwinkeländerungen der Beiden Räder berechnet. Dies ist der "wahreSZustand der Odometrie in der Simulation. Aus diesen Winkelstellungen³ der Räder wird ein Wert für die Inkrementalgeber abgeleitet und auf ganze Inkremente gerundet. Dabei gibt es einen Parameter $\sigma_{increment}$ der zu einem Gaußschen in den Inkrementen führt. <—(mehr details???)

Passanten oder Besucher (5) @@@@@@@@@@@@@@ Platzhalter @@@@@@@@@@@@@@

Partikel (6) werden zur Veranschaulichung und zu debugging Zwecken visualisiert. Sie werden durch rote kleine spitze Dreiecke dargestellt. Der spitze Winkel zeigt dabei die Orientierung an. Sie befinden sich nur auf dem Boden und beeinträchtigen die Bildverarbeitung deshalb nicht.

3.1.2 Messen in der Simulation

Die Simulation soll, neben der Visualisierung, Messungen liefern um den Lokalisationsalgorithmus testen zu können. Anders als bei Messungen an realen Experimenten hat man in der Simulation den Vorteil, alle das Messergebnis beeinflussenden Faktoren unter Kontrolle zu haben. Möchte man also, dass Messungen eine systematische Abweichung aufweisen, muss diese in der Simulation definiert werden.

³Die Winkelstellung der Räder wird nur numerisch simuliert und ist am Modell nicht sichtbar.

Genau so verhält es sich mit statistischen Abweichungen. Es lassen sich also schnell die Messunsicherheiten der Simulierten Vorgänge anpassen. Beim Debugging und Funktionstest des Lokalisationsalgorithmus wurde die Unsicherheit zum Beispiel zeitweise entfernt.

3.2 Lokalisation

Die Lokalisation ist als eine Klasse implementiert

4 Versuchsdurchführung

4.1 Theoretische Vorüberlegungen

Ideen zu Fragestellungen die Versuche nötig machen:

- Wie viele Bilder pro Zeit/Stecke sind nötig? Wie genau will man dabei noch sein? Evtl. Bildfrequenz zu Mittlerer Unsicherheit Darstellen?
- Wie wirkt es sich aus, wenn teile des Musters verdeckt sind? Ist eine Lokalisation auch bei ständiger Verdeckung möglich?
- Wie muss das Muster aufgebaut sein? Was für Fehler könne entstehen, wenn ein ungeeignetes Muster verwendet wird?
- Wie gut kommt der Filter mit Systematischen Fehlern zurecht?

4.2 Versuchsaufbau

4.3 Versuchsvorbereitung

4.4 Praktische Versuchsdurchführung

5 Versuchsergebnisse

6 Diskussion der Messergebnisse

7 Weitere Problemstellungen / Ausblick

8 Fazit

A Abkürzungsverzeichnis

CDT C/C++ Development Tools

OCG Open Scene Graph

BBM Beobachter der Bediener von Maschinen

SLAM Simultaneous Localization and Mapping

B Literaturverzeichnis

- [Hertzberg et al., 2012] Hertzberg, J., Lingemann, K., and Nüchter, A. (2012). *Mobile Roboter: eine Einführung aus Sicht der Informatik*. eXamen.press. Springer Vieweg, Berlin [u.a.]. X, 389 S. : zahlr. Ill. und graph. Darst.
- [Seco et al., 2009] Seco, F., Jiménez, A. R., Prieto, C., Roa, J., and Koutsou, K. (2009). A survey of mathematical methods for indoor localization. In *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, pages 9–14. IEEE.
- [Thrun, 2002] Thrun, S. (2002). Particle filters in robotics. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, UAI’02, pages 511–518, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Thrun et al., 2006] Thrun, S., Fox, D., and Burgard, W. (2006). *Probabilistic robotics*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Mass. [u.a.]. XX, 647 S. : Ill., graph. Darst.