



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Andrej Rull

Fahrspur- und Odometrie-basierte
Selbstlokalisierung und Kartierung (SLAM)

Andrej Rull
Fahrspur- und Odometrie-basierte
Selbstlokalisierung und Kartierung (SLAM)

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Stephan Pareigis
Zweitgutachter : Prof. Dr. Ing. Andreas Meisel

Abgegeben am 21. Dezember 2010

Andrej Rull

Thema der Masterarbeit

Fahrspur- und Odometrie-basierte Selbstlokalisierung und Kartierung (SLAM)

Stichworte

autonome Fahrzeuge, Weltmodellierung, Positionsbestimmung, Positionskorrektur, Selbstlokalisierung, SLAM

Kurzzusammenfassung

Ein autonomes Fahrzeug, das sich selbstständig auf einer Strecke orientieren soll, benötigt eine Repräsentation der Fahrbahn. Die Repräsentation erfolgt in Form einer Karte, die sich während der Fahrt aus den Umgebungsdaten aufbaut. Für den Aufbau des Weltmodells wird eine korrekte Fahrzeugposition benötigt, die vom Fahrzeug berechnet werden muss. Das Problem bei der Positionsberechnung ist die Abhängigkeit von einem korrekten Weltmodell. Ist eine Karte nicht gegeben, kann die Position nur geschätzt werden. Diese Masterarbeit beschäftigt sich mit der Problematik der gleichzeitigen Positionsbestimmung und der Weltmodellierung auf einer Fahrbahn.

Andrej Rull

Title of the paper

Lane- and odometry-based localization and mapping (SLAM)

Keywords

autonomous vehicles, environment mapping, selflocalization, position correction, SLAM

Abstract

An autonomous vehicle which should orientated itself independently on a track needs a representation of the application environment. The representation occurs in form of a map which is based on the observed data. For the construction of the world model a correct position is needed. The position must be located by the vehicle. The problem of the localization is the dependence on a correct model from the environment. Without a map the position can be only estimated. This masterthesis discuss the problem of the simultaneous localization and mapping on a track.

Inhaltsverzeichnis

Abbildungsverzeichnis	6
1 Einführung	8
1.1 Zielsetzung der Arbeit	9
1.2 Aufbau der Arbeit	9
2 Weltmodellierung	10
2.1 Aufbau eines Weltmodells	10
2.2 Das Beobachtungsmodell - measurement model	12
2.2.1 Landmarkenbasierte Ansätze	13
2.2.2 Scanbasierte Ansätze	14
2.3 Fahrzeuglokalisierung	15
2.3.1 Absolute Positionsbestimmung (Bewegungsmodell mit Karte)	16
2.3.2 Relative Positionsbestimmung (Bewegungsmodell ohne Karte)	17
3 Simultaneous Localization and Mapping	19
3.1 SLAM mit dem Extended Kalman Filter	22
3.2 SLAM mit dem Graph-SLAM Verfahren	26
3.3 SLAM mit dem Partikel Filter	28
4 Related Works	30
4.1 Visualisierte SLAM Verfahren	30
4.2 Bearing Only SLAM Verfahren	32
4.2.1 EKF-basierter Ansatz	33
4.2.2 PF-basierter Ansatz	34
4.3 FAST-SLAM Verfahren	34
4.4 Multihypothesen Ansätze	37
5 Fahrspurbasiertes SLAM	39
5.1 Einsatzfahrzeug und Sensorik	41
5.2 Problemstellung und Anforderungen	43
5.3 Topologisches Weltmodell als Grundlage	46
5.3.1 Topologische Regionen	47

5.3.2	Lokales Weltmodell	48
5.3.3	Globales Weltmodell	50
5.4	Abarbeitungsphasen	51
5.4.1	Topologische Knotenbildung (Controlling Phase)	53
5.4.2	Positionsschätzung (Prediction Phase)	58
5.4.3	Positionskorrektur und Verifikation (Update Phase)	64
6	Implementierung	71
6.1	Aufbau der Softwarearchitektur	71
6.2	Ablauf der Weltmodellierung	73
7	Auswertung der Verfahren	75
7.1	Schätzung 1	76
7.2	Schätzung 2	76
7.3	Schätzung 3	77
8	Resümee und Ausblick	78
	Literaturverzeichnis	80

Abbildungsverzeichnis

2.1	Arten der Weltdarstellung	11
2.2	Weltmodell-Hierarchie ¹	12
2.3	Landmarkenbasiertes Weltmodell	13
2.4	Scanbasiertes Modell	14
2.5	Aufteilung der absoluten Positionsbestimmung	16
2.6	Fehler der Odometrie	17
2.7	Problem der relativen Positionsbestimmung ²	18
3.1	Das allgemeine SLAM Problem	20
3.2	Das online SLAM Problem	20
3.3	Das offline SLAM Problem	21
3.4	SLAM mit dem Extended Kalman Filter ³	24
3.5	Graph SLAM Darstellung	26
3.6	Aufbau der Informationsmatrix beim GraphSLAM	27
3.7	SLAM mit dem Partikel Filter	28
3.8	Weltmodell definiert über die Menge der Partikel	29
4.1	Unterschiedliche Koordinatenframes beim vSLAM ⁴	31
4.2	Omnidirektionale Kameraansicht ⁵	33
4.3	Fast-SLAM und EKF-Slam im Vergleich ⁶	36
4.4	ATLAS Dijkstra-Projektion ⁷	37
4.5	Positionskorrektur durch benachbarte Frames ⁸	38
5.1	Die Einsatzumgebung	39
5.2	Das vollständige Slam Problem	40
5.3	Das Einsatzfahrzeug	41
5.4	Eingesetzte Sensorik	42
5.5	Softwareaufbau	42
5.6	Fehler der Odometrie	44
5.7	Fehler durch das Rauschen in der Sensorik	46
5.8	Topologische Einteilung der Einsatzumgebung	48
5.9	Lokale Weltmodellierung	49
5.10	Globales Koordinatensystem	51

5.11	Samplemenge und die Fehlerellipsen	52
5.12	Topologische Bereiche in der Einsatzumgebung	53
5.13	Die Maxima im Hough-Raum	55
5.14	Bewertung über die aktuelle Beobachtung	56
5.15	Knotenbildung in der Einsatzumgebung	57
5.16	Erstellung neuer Knoten	58
5.17	Odometrie	60
5.18	Geschätzte Pfadmenge unter Zunahme einer normalverteilten Abweichung .	63
5.19	Scanmatching zwischen dem Referenzscan r_{ref} und dem aktuellen Scan s_t .	65
5.20	Korrespondenzpartner auf einem Liniensegment	66
5.21	Referenzpunkt mit der gleichen Distanz zum Koordinatenursprung ⁹	68
5.22	Auswahl des Schätzpfades	70
6.1	Softwarearchitektur	71
6.2	Globaler Aufbau der Kartierungssoftware	72
6.3	Ablauf der Scanaufnahme	74
7.1	Schätzung des Weltmodells über die Gleichverteilung des Odometriefehlers .	75
7.2	Schätzung des Weltmodells mit fünf Schätzpfaden	76
7.3	Schätzung des Weltmodells mit zwanzig Schätzpfaden. Links die Schätzung mit einer Runde und rechts die Schätzung über mehrere Runden	77

1 Einführung

Seit Jahren gewinnen autonome mobile Systeme im alltäglichen Leben zunehmend an Bedeutung. Das Ziel der Systeme ist es den Menschen bei unterschiedlichen Aufgaben zu unterstützen und damit zu entlasten. Autonome mobile Systeme bedienen sich am Vorteil der Sensorik, die im Gegensatz zum Menschen Ermüdungserscheinungen und Emotionen stabil gegenüber steht und damit die Systeme zuverlässiger und länger arbeiten lässt.

Heutzutage wird die Technologie in mehreren Bereichen eingesetzt. So entwickelt die Firma iRobot autonome Staubsauger, die den Menschen im Haushalt unterstützen. Das Hamburger Unternehmen STILL erforscht autonome Verfahren, um Gabelstapler eigenständig fahren zu lassen, ohne dass der Mensch eingreifen muss. Ebenfalls setzt die NASA die Technologie ein um mit den beiden Erkundungsrobotern Opportunity und Spirit auf dem Mars nach Wasser zu suchen.

Auch die Automobilbranche hat die Vorteile autonomer mobiler Systeme erkannt und versucht durch verschiedene Fahrassistenzsysteme den Menschen beim Führen eines Fahrzeugs zu unterstützen. Im Gegensatz zu voll-autonomen Systemen agieren Fahrassistenzsysteme halb-autonom, indem sie nur in Gefahrensituationen in das Fahrgeschehen eingreifen oder dem Fahrzeugführer Arbeiten abnehmen. Das bekannteste Fahrassistenzsystem ist das Antiblockiersystem, das beim Bremsen das Blockieren der Räder verhindert.

Die große Herausforderung ist jedoch ein Fahrzeug zu entwickeln, das vollständig autonom fährt ohne dass der Mensch eingreifen muss. Die Universität Braunschweig erforscht zusammen mit der Volkswagen AG das autonome Fahren auf dem Braunschweiger-Stadtring. Ein mit einer Kamera und mehreren Sensoren ausgestatteter VW Passat muss sich Anhand der Kamerabilder und einer vollständigen Umgebungskarte auf dem Stadtring orientieren können ohne die Verkehrsregeln zu verletzen. Die Position in der Umgebung wird über das Global Positioning System (GPS) bestimmt.

An der Hochschule für Angewandte Wissenschaften in Hamburg wurde für den Schwerpunkt des autonomen Fahrens das Forschungsprojekt FAUST gegründet. Eines der Teilprojekte im FAUST-Kontext ist die Erforschung unterschiedlicher Fahrassistenzsystemen, wie Einpark-, Spurführungs- oder Ausweichassistent, die zusammen eine autonome Fahrt eines Modellfahrzeugs ermöglichen sollen. Dabei werden Informationen der Umgebung über Sensoren aufgenommen, fusioniert und der Regelung des Fahrzeugs zu Verfügung gestellt.

Ein Schwerpunkt in der Forschung ist die Generierung eines Weltmodells. Ein Weltmodell definiert eine Karte, die eine globale Planung der Fahrt möglich macht. Das Problem ist, dass von der Einsatzumgebung keine Umgebungskarte vorhanden ist, sodass diese vom Fahrzeug autonom während der Fahrt modelliert werden muss.

1.1 Zielsetzung der Arbeit

Das Ziel dieser Arbeit ist es ein Verfahren zu entwickeln, das eine autonome Weltmodellierung unter Berücksichtigung aller Anforderungen ermöglicht. Das Verfahren soll in der Lage sein Umgebungsdaten einer Fahrbahn in einer Indoor-Umgebung aufzunehmen und diese korrekt in einem Weltmodell zu fusionieren. Der Schwerpunkt liegt dabei im Aufbau eines korrekten Weltmodells. Als Einsatzfahrzeug wird eines an der HAW Hamburg entwickelte Modellfahrzeug im Maßstab 1:10 verwendet.

1.2 Aufbau der Arbeit

Die Arbeit ist in mehrere Abschnitte gegliedert. Zunächst wird im [Kap. 2] die unterschiedlichen Weltmodellarten vorgestellt und die Grundlagen erläutert die bei der Weltmodellierung unverzichtbar sind. Anschließend wird im [Kap. 3] das grundlegende Problem der Kartierung beschrieben und einige Ansätze vorgestellt, anhand derer sich die Problematik lösen lässt. Im [Kap. 4] werden Methoden und Verfahren erläutert die aktuell in der Forschung zu finden sind und die Unterschiede zwischen den Verfahren erkennen lassen. Das [Kap. 5] beschäftigt sich mit der Problematik dieser Arbeit. Es werden Anforderungen an die fahrspurbasierte Weltmodellierung aufgestellt und Algorithmen zur Positionsbestimmung und Kartenkorrektur entwickelt, die eine autonome Weltmodellierung ermöglichen. Die Implementierung der Verfahren und der Softwareaufbau wird in [Kap. 6] beschrieben. Zum Ende dieser Arbeit werden die Verfahren bewertet und ein Ausblick gegeben.

2 Weltmodellierung

Damit ein Fahrzeug sich selbstständig in der Umgebung bzw. auf einer Strecke orientieren kann, muss es, genau wie der Mensch, die befahrene Umgebung und deren Merkmale aufnehmen und diese richtig interpretieren können. Dazu wird eine interne Weltdarstellung aufgebaut, die als Weltmodell bezeichnet wird. Das Weltmodell muss dabei so genau aufgebaut sein, dass es allen Anforderungen, wie z.B. der Hinderniserkennung und der Wegplanung, gerecht wird.

2.1 Aufbau eines Weltmodells

Generell ist ein Weltmodell zuständig für die Verwaltung systeminterner Darstellungen der Einsatzumgebung. Das Modell kann statisch sein und zu Beginn einer Fahrt dem Einsatzfahrzeug zur Verfügung gestellt werden oder dynamisch generiert werden. Ein dynamisches Weltmodell baut sich während der Fahrt autonom auf, indem das Einsatzfahrzeug die Daten der Umgebung aufnimmt, die Daten zusammenführt und aus den fusionierten Daten eine Karte modelliert.

In der Literatur werden mehrere Modellarten unterschieden, die die Umgebung auf unterschiedlichen Abstraktionsebenen darstellen. Eine Abstraktionsebene ist ein Modell einer Karte, die je nach Anforderung unterschiedliche Informationen beinhaltet. Der Abstraktionsgrad reicht dabei von der einfachen Sensorkarte, die reine Sensorwerte in Form von Koordinatenpunkten abspeichert, bis hin zur metrischen oder topologischen Darstellungen der Karte, die für komplexe Anforderungen Sensordaten fusioniert, um mehr Aussagen über die Umgebung treffen zu können [Knieriemen (1991), S.53]. Ein Beispiel von drei Abstraktionsebenen ist in [Abb. 2.1] dargestellt.

Unter Verwendung unterschiedlicher Kartentypen, kann ein System mehrere Anforderungen erfüllen. Dabei soll die Verwaltung und die Aktualisierung der Karte einfach gehalten werden. Zwei Abstraktionsebenen (Kartentypen), die in der Literatur häufig erwähnt werden, sind das topologische und das geometrische Weltmodell. In der Regel baut sich ein topologisches Modell aus der Lage und den Anordnungen geometrischer Strukturen auf, wie Kanten von Hindernissen, Kreuzungen, Abzweigungen oder auch konvexen Freiräumen, die laut [Kuijpers (2000)] markante Punkte auf einer geometrischen Karte bilden. Die markanten Stellen

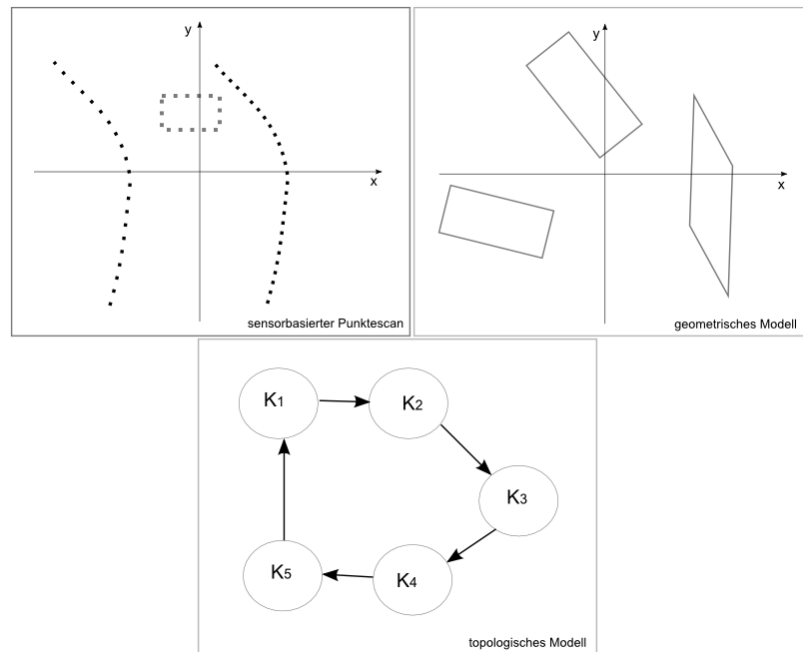
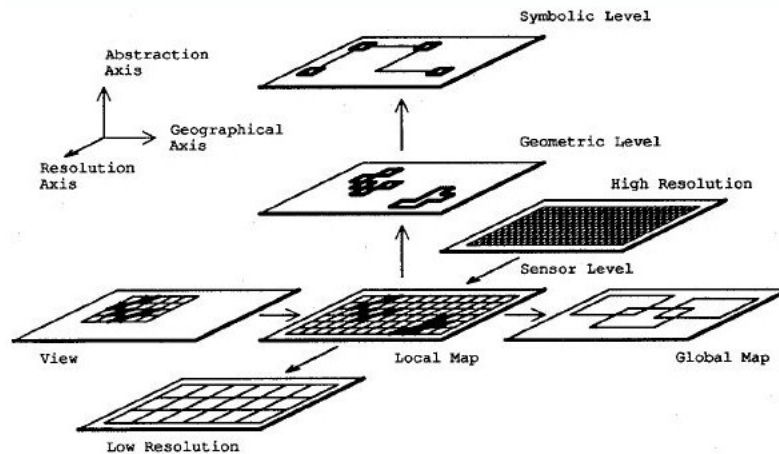


Abbildung 2.1: Arten der Weltdarstellung

definieren Knoten im Modell, die durch topologische Beziehungen (Transition / Kanten) miteinander verbunden sind. Die topologische Darstellung, in der [Abb. 2.2] als Symbolic Level Map gekennzeichnet, wird als Graph dargestellt und ist in der Modellhierarchie, wie in der Abbildung zu sehen ist, ganz oben angesiedelt.

Anders als die topologischen Karten, bündeln die geometrischen Karten die Sensordaten in Form von Merkmalspolynomen oder Rastern, um eine geometrische Struktur herzustellen. Eine genaue Beschreibung der geometrischen Struktur und die Unterschiede zwischen einer Polynomdarstellung und einer Rasterdarstellung wird in [Rull (2008)] genauer erklärt.

Die Modellierung der Weltmodelle erfolgt auf unterschiedlicher Art und Weise. In [Kuipers (2000), S.191-233] werden aus den Sensorwerten topologische Karten modelliert, denen zu einem späteren Zeitpunkt Umgebungsmerkmale bzw. lokale Umgebungsmessungen zugeordnet werden. Eine lokale Umgebungsmessung beschreibt die aktuelle Umgebung aus der Sicht des Fahrzeugs. Anders als in [Kuipers (2000), S.191-233], wird in [Erfes (1987), S.249-265] zunächst eine geometrische Abstraktionsebene aufgebaut, aus der später eine topologische Metrik gebildet wird. Dabei wird in der geometrischen Struktur nach Bereichen gesucht, die sich möglichst stark unterscheiden und aus den Bereichen topologische Knotenpunkte gebildet. Kanten zwischen den Knotenpunkten definieren den Zusammenhang der Bereiche.

Abbildung 2.2: Weltmodell-Hierarchie¹

2.2 Das Beobachtungsmodell - measurement model

Um überhaupt eine Karte modellieren zu können, muss die aufgenommene Umgebung über die externe Sensorik (Kamera, Ultraschallsensoren, Laserscanner) aufgenommen und ein Abgleich der gespeicherten Daten mit den neuen Daten hergestellt werden. In der Literatur wird der Prozess der lokalen Umgebungsaufnahme anhand des Beobachtungsmodells (measurement model) definiert [Sebastian Thrun (2008)]. Das Beobachtungsmodell gibt die bedingte Wahrscheinlichkeit der Messung zum Zeitpunkt t an.

$$p(z_t | x_t, m)$$

wo x_t die Fahrzeugposition, z_t die Messdaten zur Zeit t und m das Weltmodell der Umgebung darstellen. Der obere Ausdruck der bedingten Wahrscheinlichkeit gibt den Zusammenhang der Messung z_t zu den Daten im Weltmodell m an der Position x_t an. Damit schätzt das Beobachtungsmodell unter Berücksichtigung der aktuellen Position eine Umgebungsmessung aus der Umgebungskarte m ab. Da bei der lokalen Umgebungsaufnahme unterschiedliche Sensoren angewandt werden, kann es zu unterschiedlichen Messungen pro Zeit t kommen. Eine aktuelle Messung kann demnach über die Menge der Messdaten definiert werden.

$$z_t = z_t^1, z_t^2, \dots, z_t^n$$

¹Erfes (1987)

Die Gesamtwahrscheinlichkeit der zugeordneten Messung ($z_t|x_t, m$) ergibt sich aus dem Produkt der einzelnen Messabschätzungen.

$$p(z_t|x_t, m) = \prod p(z_t^i|x_t, m)$$

Um die Schätzung der Umgebungsmessung zu verbessern und damit die neuen Umgebungsmessungen den alten besser zuordnen zu können, werden unterschiedliche Modelle verwendet. Die Modelle basieren auf feature- und scanbasierten Ansätzen.

2.2.1 Landmarkenbasierte Ansätze

Feature- bzw. landmarkenbasierte Ansätze definieren die Umgebung über die Menge eindeutiger Landmarken l_i [?].

$$m = l_1, l_2, \dots, l_n$$

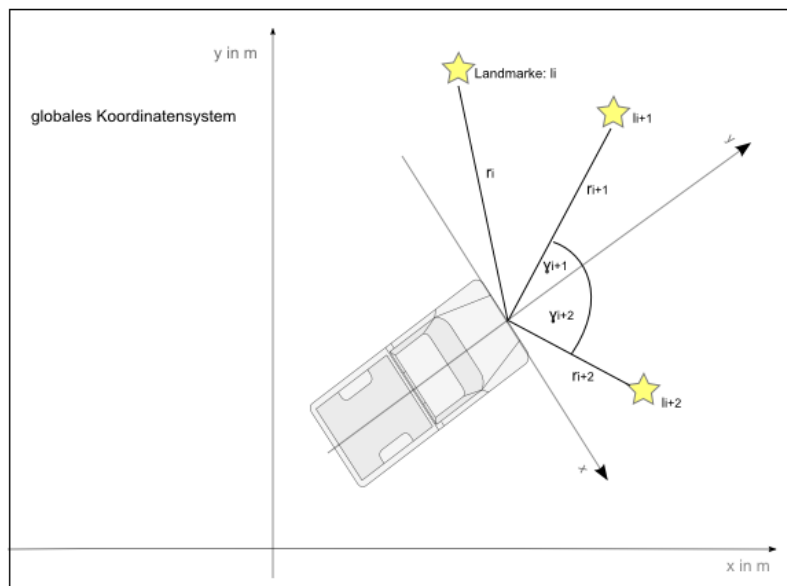


Abbildung 2.3: Landmarkenbasiertes Weltmodell

Jede Landmarke l_i wird durch den Vektor $l_i = (r_i, \gamma_i)^T$ beschrieben, wo r_i die Entfernung und γ_i die Ausrichtung zum Merkmal darstellen [Abb. 2.3].

Wird vom Fahrzeug eine neue Messung z_t aufgenommen, werden anhand einer Extraktionsfunktion $F(z_t)$ Landmarken aus der Messung abstrahiert. Die Anzahl möglicher Landmar-

ken ist pro Messung variabel, sodass eine Momentaufnahme aus einer Menge von Landmarken besteht. Je mehr Landmarken in der Messung gefunden werden, desto einfacher ist die Zuordnung der Messung.

$$F(z_t) = l_t^1, l_t^2, \dots, l_t^n$$

Das Beobachtungsmodell von vorhin ändert sich damit wie folgt:

$$p(F(z_t)|x_t, m) = \prod p(l_t^i|x_t, m)$$

Die Wahrscheinlichkeit $p(F(z_t)|x_t, m)$ nimmt zu, je besser die extrahierten Landmarken der aktuellen Messung mit den gespeicherten Landmarken im Weltmodell an der Position x_t übereinstimmen.

2.2.2 Scanbasierte Ansätze

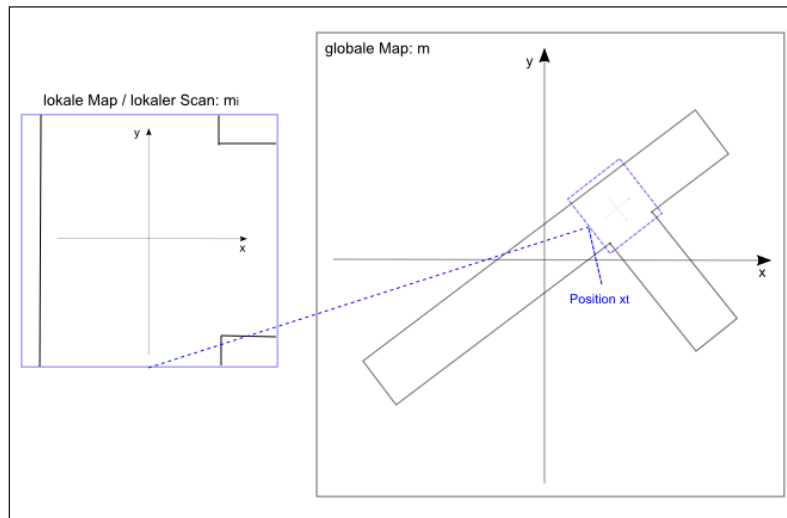


Abbildung 2.4: Scanbasiertes Modell

Andere Methoden zum Abgleichen der aktuellen Messung mit gespeicherten Daten sind scanbasierte Ansätze. Scanbasierte Verfahren beschreiben die globale Umgebungsstruktur über eine Menge lokaler Kartendarstellungen. Das globale Weltmodell wird nicht wie beim featurebasierten Ansatz durch die Menge der Umgebungsmerkmale definiert, sondern über

die Menge der aufgenommenen Umgebungskarten [Michael Bosse (2003)]. Jede Karte stellt eindeutig einen Bereich in der Umgebung dar, sodass sich die globale Struktur aus der Menge der lokalen Umgebungskarten bilden lässt.

$$m = m_1, m_2, \dots, m_n$$

Im Gegensatz zur globalen Map m , stellt eine lokale Map m_i die Umgebung relativ zu der Fahrzeugposition dar. Ist das Fahrzeug an der Position x_t , wird die lokale Kartendarstellung anhand des Positionsvektors $x_t = (x_t, y_t, \theta_t)$ transformiert und die lokalen Koordinaten in globale überführt. Anschließend wird in der globalen Umgebungskarte nach dem Bereich gesucht, mit der die transformierte lokale Karte korreliert. Die Technik des Abgleichs zweier Karten wird in der Literatur auch als "map matching" bezeichnet [Lu und Milios (1997)]. Das Verfahren lässt sich über die a-posteriori Wahrscheinlichkeit beschreiben. Je größer die Ähnlichkeit zwischen der lokalen Karte m_i und den Daten im globalen Bereich an der Position x_t ist, desto größer ist die Wahrscheinlichkeit des Zusammenhangs.

$$p(m_i | x_t, m)$$

Welches Verfahren für die Weltmodellierung benutzt, hängt von der Einsatzumgebung ab. Bietet die Einsatzumgebung nur wenige eindeutige Merkmale, die extrahiert werden können, sind scanbasierte Ansätze besser geeignet. Die Ansätze haben zwar den Vorteil, dass selbst Rohdaten miteinander verglichen werden können, können jedoch bei der Suche nach einer Korrelation im globalen Weltmodell eine längere Bearbeitungszeit verursachen.

2.3 Fahrzeuglokalisierung

Ein wichtiger Grundstein bei der Weltmodellierung ist die globale Position des Fahrzeugs. Die Position muss zu jeder Zeit bekannt sein, um die beobachteten Daten korrekt in eine metrische Karte abspeichern zu können. Ist die Position nicht gegeben, muss sie eigenständig berechnet werden. Man spricht von der Selbstlokalisierung des Fahrzeugs. Eine Lokalisierung ist eine Berechnung der Position und der Orientierung in Bezug auf das Weltkoordinatensystem und wird durch den Vektor P angegeben.

$$P = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

Dabei geben (x, y) die Positionskordinaten im Bezug auf das Weltkoordinatensystem an und θ den Orientierungswinkel des Fahrzeugs im Koordinatensystem. Im weiteren Verlauf der Arbeit wird der Positionsvektor P über den Positionszustand x_t angegeben.

In der Fachliteratur wird generell zwischen zwei verschiedenen Selbstlokalisierungsmaßnahmen unterschieden: der absolute und der relative Selbstlokalisierung [Gutmann (1996)].

2.3.1 Absolute Positionsbestimmung (Bewegungsmodell mit Karte)

Die absolute Positionsbestimmung wird in der Literatur auch als "map motion model" gekennzeichnet [Sebastian Thrun (2008)]. Das Modell kann über die bedingte Wahrscheinlichkeit des Positionszustandes beschrieben werden.

$$p(x_t | u_t, x_{t-1}, m)$$

Für die neue Positionsabschätzung x_t wird der alte Positionszustand x_{t-1} , die Fahrzeugbewegung u_t zur neuen Position und eine Darstellung der Umgebung m verwendet.

Die absolute Positionsbestimmung lässt sich auf mehreren Arten umsetzen. Die Gliederung der möglichen Lokalisierungsmaßnahmen ist in [Abb. 2.5] zu sehen.

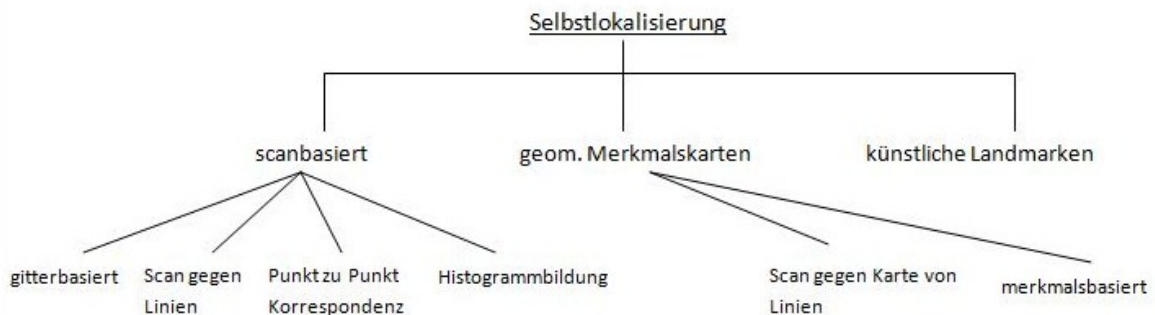


Abbildung 2.5: Aufteilung der absoluten Positionsbestimmung

Der wohl einfachste Ansatz der Selbstlokalisierung ist die Position anhand von künstlichen Landmarken abzuschätzen. In der Umgebung werden kodierte Marken aufgestellt, die eine Position im globalen Weltmodell darstellen. Detektiert das Fahrzeug während der Fahrt eine Landmarke, wird diese aufgenommen und dekodiert. Da die Positionen der Landmarken bekannt sind und die Landmarken eindeutig sind, kann die globale Fahrzeugposition über die Entfernung und Ausrichtung zu einer Landmarke bestimmt werden.

Scan- und landmarkenbasierte Verfahren stellen eine dynamische Positionsbestimmung dar, die eine fertige und fehlerfreie Umgebungskarte benötigen anhand derer das Fahrzeug autonom entscheidet wo es sich in der Umgebung befindet. Die Positionsbestimmung erfolgt über den Abgleich der aufgenommenen Daten mit den gespeicherten Kartendaten. Wird ein Umgebungsmerkmal wiedererkannt, wird die aktuelle Ausrichtung und Entfernung zum Merkmal berechnet und die globale Fahrzeugposition anhand der Karte aktualisiert.

Der Vorteil der absoluten Positionsbestimmung liegt darin, dass sich immer eine genaue Fahrzeugposition berechnen lässt, wenn Landmarken wiedererkannt werden. Ein Nachteil liegt jedoch in der eindeutigen Kennzeichnung von natürlichen Landmarken bzw. der zwingend erforderlichen globalen Karte. Zudem müssen die Landmarkenpositionen korrekt sein.

2.3.2 Relative Positionsbestimmung (Bewegungsmodell ohne Karte)

Sind keine Landmarken in der Umgebung bekannt oder sind die Positionen der Landmarken fehlerhaft muss eine andere Methode der Selbstlokalisierung angewandt werden. In der Praxis hat sich ein Verfahren etabliert, das die Fahrzeugposition nur über die Fahrzeugbewegung berechnet [Daniel Pizarro und Losad \(2010\)](#).

$$p(x_t | u_t, x_{t-1})$$

Bei der sogenannten relativen Positionsbestimmung wird die Fahrzeugposition in der Umgebung ohne eine globale Umgebungsdarstellung geschätzt. Die Schätzung erfolgt über die Auswertung der Fahrzeuggeschwindigkeit, den Lenkwinkel und der Wegmessungen. Ein Nachteil der entsteht ist die Abhängigkeit von einer fehlerfreien Fahrzeugbewegung während der gesamten Zeit. Rauschen in der Sensorik, Schlupf der Räder oder Unebenheiten in der Umgebung führen zu Fehlern in der Positionsberechnung [Abb. 2.6].

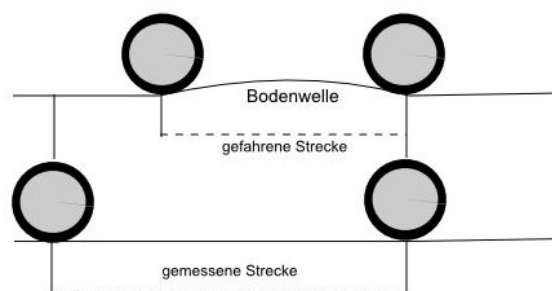


Abbildung 2.6: Fehler der Odometrie

Wird eine größere Strecke zurück gelegt akkumulieren sich die Fehler der Odometrie auf und die Positionsschätzung wird unbrauchbar. [Abb. 2.7] veranschaulicht die Problematik der odometriebasierten Positionsbestimmung.

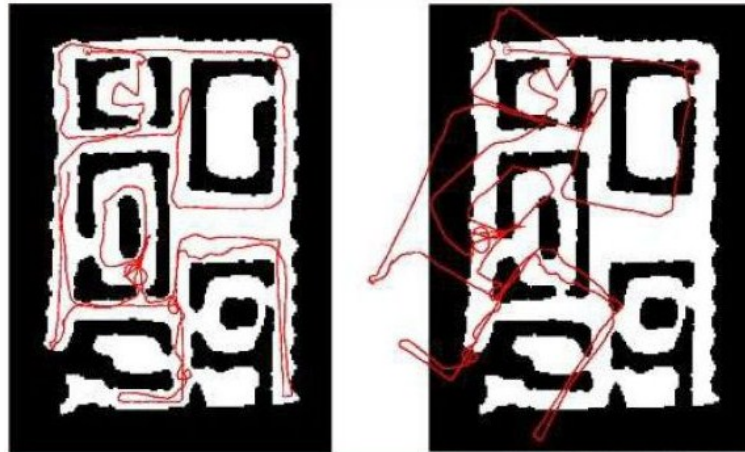


Abbildung 2.7: Problem der relativen Positionsbestimmung²

Um das Problem zu beheben, wird die relative Positionsbestimmung in zwei wesentliche Teile gegliedert. Zum Einen wird der Positionszustand mittels der Odometrie anhand des Bewegungsmodells geschätzt und zum Anderen durch Korrekturmaßnahmen gelegentlich korrigiert und aktualisiert. Für die Korrektur wird das Weltmodell m verwendet, das sich während der Fahrt aufbaut. Da keine genaue Position in der Umgebung bekannt ist, kann nur ein fehlerbehaftetes Weltmodell aufgebaut werden, was zum wesentlichen Problem der Weltmodellierung führt. In der Literatur ist das Problem auch als "simultaneous localization and mapping problem", kurz SLAM Problem, bekannt.

3 Simultaneous Localization and Mapping

In diesem Kapitel wird eines der größten Probleme bei der Weltmodellierung erläutert. "Simultaneous localization and mapping", kurz SLAM, steht für eine Umgebungskartenmodellierung und Positionsbestimmung zur gleichen Zeit. Das SLAM Problem tritt auf, wenn dem Fahrzeug zu Beginn einer Fahrt keine Umgebungsinformationen in Form einer Karte zur Verfügung stehen und das Fahrzeug keine Informationen über die eigene Position in der Umgebung hat. Die Problematik besteht darin, dass für eine genaue Positionsbestimmung eine genaue Karte m vorausgesetzt wird, anhand derer sich das Fahrzeug orientieren und die Position x_t lokalisieren kann. Ist die Karte m nicht gegeben, muss diese generiert werden. Gleichzeitig erfordert jedoch die Modellierung einer Karte eine genaue Position des Fahrzeugs in der Umgebung, was zum SLAM Problem führt.

Die Weltmodellierung hängt von den Kerngrößen $x_{0:t}$, $u_{1:t}$ und $z_{1:t}$ ab.

$$\begin{aligned}x_{0:t} &= \{x_t, x_{t+1}, \dots, x_{t+n}\} \\u_{1:t} &= \{u_t, u_{t+1}, \dots, u_{t+n}\} \\z_{1:t} &= \{z_t, z_{t+1}, \dots, z_{t+n}\}\end{aligned}$$

Dabei enthält die Menge $x_{0:t}$ die Positionszustände, die das Fahrzeug während eine Fahrt erreicht hat, $u_{1:t}$ die kinematischen Zustände(Fahrzeugbewegung, Odometrie), die das Fahrzeug während der Fahrt getätigt hat und $z_{1:t}$ die Messungen der Umgebung. Das Weltmodell m wird demnach aus der Menge der Umgebungsmessungen $z_{1:t}$ und den Positionszustandsvektoren $x_{0:t}$ gebildet.

$$m = \text{map}(x_{0:t}, z_{1:t})$$

[Abb. 3.1] veranschaulicht die SLAM Problematik. Die graugekennzeichneten Knotenpunkte, stellen gegebene Zustände, wie Fahrzeugbewegung und Umgebungsmessung dar, während die weißen Knoten für die unbekannt Zustände (Fahrzeugposition und Weltmodell) stehen. Um das Weltmodell zu generieren und die Fahrzeugposition zu ermitteln, werden die kinematischen Bewegungsabläufe $u_{1:t}$ des Fahrzeugs und die getätigte Messungen $z_{1:t}$ verwendet.

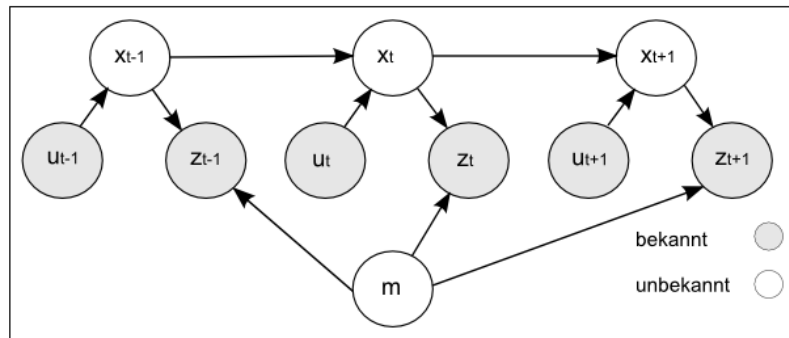


Abbildung 3.1: Das allgemeine SLAM Problem

Für die Berechnung des aktuellen Positionszustandes x_t wird die relative Position, mittels der kinematischen Zustände, bestimmt und anschließend der relative Positionszustand, über einen Abgleich der aktuellen Messungen z_t mit den zuvor aufgenommenen Messdaten m , aktualisiert.

In der Literatur wird zwischen dem dem online SLAM Problem und dem vollständigen SLAM Problem (offline SLAM Problem) unterschieden [[Sebastian Thrun \(2006\)](#)].

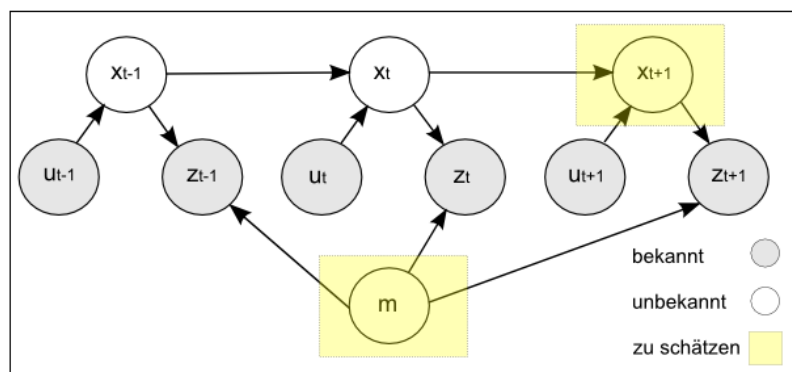


Abbildung 3.2: Das online SLAM Problem

Das online SLAM Problem wird über die a-posteriori Wahrscheinlichkeit der aktuellen Position und der Umgebungskarte beschrieben.

$$p(x_t, m | u_{t-1:t}, z_{t-1:t})$$

Für die Schätzung der Zustände werden die getätigten Messungen (u_t , z_t) verwendet, die zwischen zwei Positionsschätzungen im Zeitintervall $[t-1, t]$ durchgeführt wurden. Andere

Bewegungen und Messungen, mit denen eine Schätzung zuvor schon realisiert wurde, werden verworfen. Das online SLAM Verfahren schätzt nur den aktuellen Positionszustand x_t zur Zeit t . Die Abarbeitung des Verfahrens ist in [Abb. 3.2] zu sehen. Verlässt das Fahrzeug den aktuellen Positionszustand x_{t-1} , wird der Folgezustand x_t aus den gesammelten Messdaten $u_{t-1:t} = (u_1, u_2, \dots, u_n)$, der getätigten Umgebungsmessung $z_{t-1:t} = (z_1, z_2, \dots, z_n)$ und den gespeicherten Daten im Weltmodell m bestimmt.

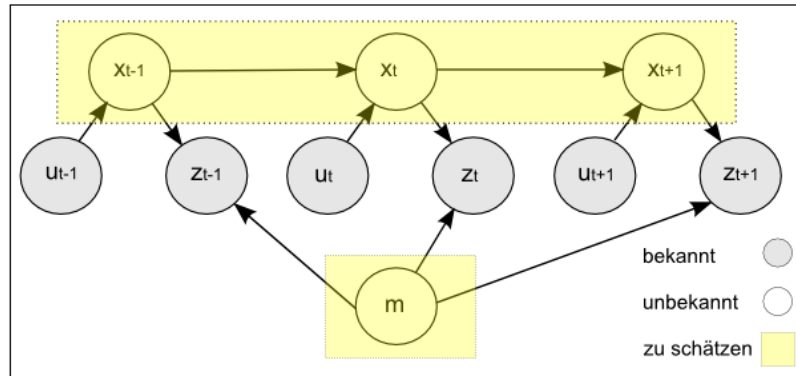


Abbildung 3.3: Das offline SLAM Problem

Anders als beim online SLAM Verfahren, wird bei einem vollständigen oder offline SLAM Problem die a-posteriori Wahrscheinlichkeit über den ganzen Positionsgraphen und der Karte berechnet [Abb. 3.3].

$$p(x_{0:t}, m | u_{1:t}, z_{1:t})$$

Durch die Schätzung des Gesamtpfades wird nicht nur der aktuelle Positionszustand x_t bestimmt, sondern auch die vorherigen Zustände $x_{0:t-1}$ aktualisiert. Laut [Sebastian Thrun (2006), S.310] ist das Ergebnis eines online SLAM Problems eine Integration über die Positionszustände des offline SLAM Problems.

$$p(x_t, m | u_{t-1:t}, z_{t-1:t}) = \int \int \dots \int p(x_{0:t}, m | u_{1:t}, z_{1:t}) dx_1, dx_2, dx_3, \dots, dx_{t-1}$$

Eine weitere Eigenschaft, die das SLAM Problem beschreibt ist die Problematik in der Schätzung. Die SLAM Schätzung beinhaltet einen diskreten und einen kontinuierlichen Ansatz [Sebastian Thrun (2006)]. Die kontinuierliche Schätzung ist die Schätzung der aufgenommenen Umgebungsmerkmale, derer Position, sowie der Fahrzeugposition. Diskrete Schätzprobleme treten bei der Bestimmung von Korrespondenzpaaren auf. Wird eine Landmarke in der Umgebung vom Fahrzeug wiedererkannt, muss das System einen Korrespondenzpartner in

der Karte finden, der mit der Landmarke korreliert. Damit muss das System in der Lage sein, den Zusammenhang zwischen neu detektierten Landmarken und zuvor aufgenommenen Merkmalen aufzustellen.

Für die Schätzung des Positionszustandes und die Lösung des kontinuierlichen und diskreten Problems, werden unterschiedliche Ansätze verwendet. Der meist verbreitetste Ansatz ist der Extended Kalman Filter (EKF), der das System als Zustandsvektor betrachtet und den Vektor kontinuierlich aktualisiert. Ein EKF Ansatz schätzt immer die aktuelle Position und die komplette Umgebung, wodurch er das online SLAM Problem löst.

Für das offline SLAM Problem werden Multihypothesenansätze bzw. Partikel Filter Verfahren verwendet. Im Vergleich zum EKF wird bei den Methoden das System durch mehrere Zustandsvektoren beschrieben, indem jeder Zustandsvektor einen Bereich in der Einsatzumgebung darstellt.

3.1 SLAM mit dem Extended Kalman Filter

Ein Kalman Filter ist ein Verfahren, das den Zustand eines linearen Systems vorhersagt. Dabei wird der geschätzte Ausgangswert mit dem tatsächlich gemessenen Ausgangswert verglichen, die daraus resultierende Differenz bzw. Abweichung linear gewichtet und zur Verbesserung der Zustandsschätzung angewandt. Für nicht lineare Systeme wird der Extended Kalman Filter verwendet, der auf einer Taylor-Näherung erster Ordnung beruht [Sebastian Thrun (2006)]. Bei einem EKF-basierten Ansatz wird das Weltmodell anhand eines Zustandsvektors y_t betrachtet. Der Vektor enthält die Fahrzeugposition $x_t = (x, y, \theta)$ und die Positionskoordinaten der Landmarken (aufgenommene Merkmale) $m_i = (x, y)$. Die Größe des Vektors ist damit $(3 + 2N)$, wobei N die Anzahl der Landmarken angibt.

$$y_t = \begin{pmatrix} x_t \\ m \end{pmatrix} = (x, y, \omega, m_{1,x}, m_{1,y}, m_{1,s}, \dots, m_{n,x}, m_{n,y}, m_{n,s})^T$$

Die Positionskoordinaten des Fahrzeug und der Landmarken werden durch Zufallsvariablen beschrieben, da sich diese während der Fahrt ändern. Da keine genau Aussage über die korrekte Position und damit über den Systemzustand gemacht werden kann, wird die Unsicherheit in der Kovarianzmatrix festgehalten. Die Matrix stellt alle Landmarken in Beziehung zu einander, sowie in Beziehung zu der Fahrzeugposition dar. Auf der Diagonalen der Matrix befinden sich die Varianzen der Zufallsvariablen. Der Systemzustand baut sich durch die Aufnahme neuer Landmarken autonom auf, indem die Landmarkenkoordinaten in Form von

Zufallsvariablen dem Vektor hinzugefügt werden. Die Erweiterung des Zustandsvektors führt gleichzeitig dazu, dass die Kovarianzmatrix quadratisch wächst. Durch den Aufbau beträgt die Größe der Matrix $(3 + 2N) \times (3 + 2N)$.

Die Abarbeitung der EKF Lokalisierung erfolgt in zwei Schritten, der Vorhersage (prediction) und der Korrektur (update). Bei der Vorhersage wird der Systemzustand y_t anhand des Bewegungsmodells aus dem alten Zustand y_{t-1} und der Fahrzeugbewegung u_t ermittelt.

$$y_t = g(u_t, y_{t-1})$$

Da der genaue Systemzustand y_{t-1} nicht vorhersagbar ist, wird der Zustand über einen Schätzzustand μ_t ermittelt. Die Schätzung erfolgt über die Taylor Annäherung erster Ordnung indem die Gleichung g über die Jacobi-Matrix G_t linearisiert wird. Die Unsicherheit der Schätzung wird in der Kovarianzmatrix festgehalten. Es wird davon ausgegangen, dass das Rauschen normalverteilt ist.

$$g(u_t, y_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t(y_{t-1} - \mu_{t-1})$$

Für den Systemzustand gilt demnach:

$$\mu_t = g(u_t, \mu_{t-1})$$

$$\Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t$$

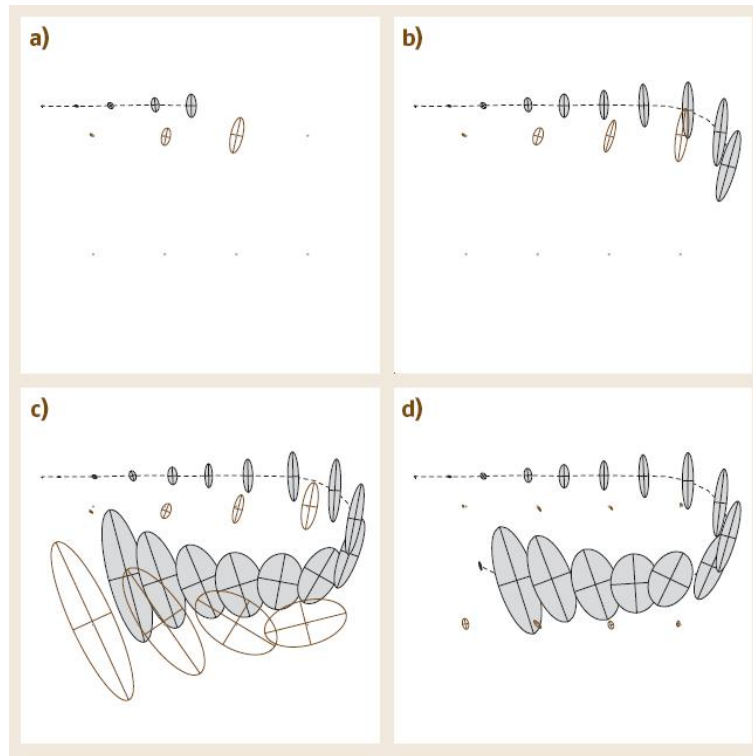
Die Kovarianz R_t ist abgeleitet aus dem Fehlermodell der Odometrie.

In der Prediction-Phase erfolgt eine Zustandsschätzung nur über das Bewegungsmodell. Damit werden nur die Fahrzeugkoordinaten und die Fahrzeugausrichtung aktualisiert. Um die Positionskoordinaten der Umgebungsmerkmale zu korrigieren wird das Beobachtungsmodell (measurement model) verwendet. Dazu werden die erwarteten Landmarken an der vorhergesagten Position x_t aus dem Zustand μ_t geschätzt und mit den Landmarken aus der aktuellen Beobachtung verglichen.

Die Schätzung der Landmarke erfolgt über das Beobachtungsmodell

$$z_t^i = h(y_t, m_i) + \mathfrak{N}(0, Q)$$

wo m_i die Position der Korrespondenzlandmarke im Systemzustand angibt, die mit der aktuellen Messung am besten korreliert. Die geschätzte Messung \tilde{z}_t wird über den geschätzten Zustandsvektor μ_t mit den Koordinaten der Landmarke $(\mu_{m_i,x}, \mu_{m_i,y})$ und der geschätzten Fahrzeugposition $(\mu_{t,x}, \mu_{t,y}, \mu_{t,\theta})$ berechnet.

Abbildung 3.4: SLAM mit dem Extended Kalman Filter³

$$\tilde{z}_t^i = \begin{pmatrix} \sqrt{(\mu_{m_i,x} - \mu_{t,x})^2 + (\mu_{m_i,y} - \mu_{t,y})^2} \\ \text{atan2}(\mu_{m_i,y} - \mu_{t,y}, (\mu_{m_i,x} - \mu_{t,x})) - \mu_{t,\theta} \end{pmatrix}$$

Da auch hier keine genau Aussage über den Systemzustand gemacht werden kann, wird die Funktion h über die lineare Funktion

$$h(y_t, m_i) \approx h(\mu_t, m_i) + H_t^i (y_t - \mu_t)$$

approximiert. H_t^i stellt die Jacobi-Matrix dar, anhand der das Beobachtungsmodell linearisiert wird. Anschließend wird mittels der Jacobian-Matrix H der Funktion h und der Kovarianzmatrix Q , die die Abweichung der Beobachtung angibt die Kalmanverstärkung berechnet und der Systemzustand μ_t mit der Kovarianzmatrix Σ_t neu berechnet.

$$K_t^i = \Sigma_t H_t^{iT} (H_t^i \Sigma_t H_t^{iT} + Q_t)^{-1}$$

³Sebastian Thrun (2008)

$$\begin{aligned}\mu_t &= \mu_t + K_t^i(z_t^i - \hat{z}_t^i) \\ \Sigma_t &= \Sigma_t(I - K_t^i H_t^i)\end{aligned}$$

Die Aktualisierung des Zustandes erfolgt für jede observierte Landmarke z_t^i in der neue Messung z_t .

Die Abarbeitung des vereinfachten EKF Algorithmus ist in der [Tab. 3.1] zu sehen.

EKF-Algorithmus
<p>Vorhersage des Systemzustandes μ_t anhand des Bewegungsmodells</p> $\mu_t = g(\mu_{t-1}, u_t)$
<p>Aktualisiere der Kovarianzmatrix unter Zunahme der Jacobi-Matrix G_t der Funktion g und der Matrix R_t, die die Abweichung in der Bewegung darstellt</p> $\Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t$
<p>for(alles observierten Landmarken $z_t^i = (r_t^i, \phi_t^i)$) in der Messung z_t)</p>
<p>Bestimme Korrespondenzlandmarke \hat{z}_t^i im Systemzustand μ_t</p> $\hat{z}_t^i = h(\mu_t)$
<p>Wurde ein Match (Korrespondenzlandmarke) gefunden, berechne den Kalman-gain K_t^i unter Zunahme der Jacobi Matrix H der Funktion h und der Kovarianzmatrix Q_t der Beobachtung</p> $K_t^i = \Sigma_t H_t^{iT} (H_t^i \Sigma_t H_t^{iT} + Q_t)^{-1}$
<p>Schätze den Systemzustand μ_t neu und aktualisiere die Kovarianzmatrix Σ_t</p> $\begin{aligned}\mu_t &= \mu_t + K_t^i(z_t^i - \hat{z}_t^i) \\ \Sigma_t &= \Sigma_t(I - K_t^i H_t^i)\end{aligned}$
<p>end for</p> <p>return μ_t, Σ_t</p>

Tabelle 3.1: EKF-Algorithmus

[Abb. 3.4] veranschaulicht die Arbeitsweise des Extended Kalman Filters. Auf Bild a), b) und c) erfolgt die Schätzung des Zustandsvektors nur über die Bewegungszustände u_t , wodurch der Fehler der Positionsschätzung im Laufe der Fahrt zunimmt. Die Zunahme wird durch die vergrößerten Fehlerellipsen deutlich. Im letzten Schritt d) wird eine Landmarke vom System

wiedererkannt und die Koordinaten des Fahrzeugs, sowie der Landmarken im Zustandsvektor aktualisiert, wodurch die Schätzungen der Fahrzeugposition und der Landmarkenposition verbessert werden.

3.2 SLAM mit dem Graph-SLAM Verfahren

Der Nachteil des Extended Kalman Filters besteht darin, dass für die Aktualisierung des Systemzustandes die benötigte Kovarianzmatrix Σ quadratisch $O(N^2)$ in Abhängigkeit der Landmarkenanzahl N wächst. Zudem muss für jeden nicht linearen Term eine Linearisierung durchgeführt werden, wodurch die Komplexität des Filters quadratisch zunimmt. Beinhaltet die Einsatzumgebung sehr viele Umgebungsinformationen (Landmarken), dauert die Zustandsaktualisierung des Systems länger als gewünscht.

Alternative Ansätze, wie der Graph SLAM Algorithmus beschleunigen die Abarbeitung der Weltmodellierung, indem die Aktualisierungshäufigkeit des Systemzustandes verringert wird.

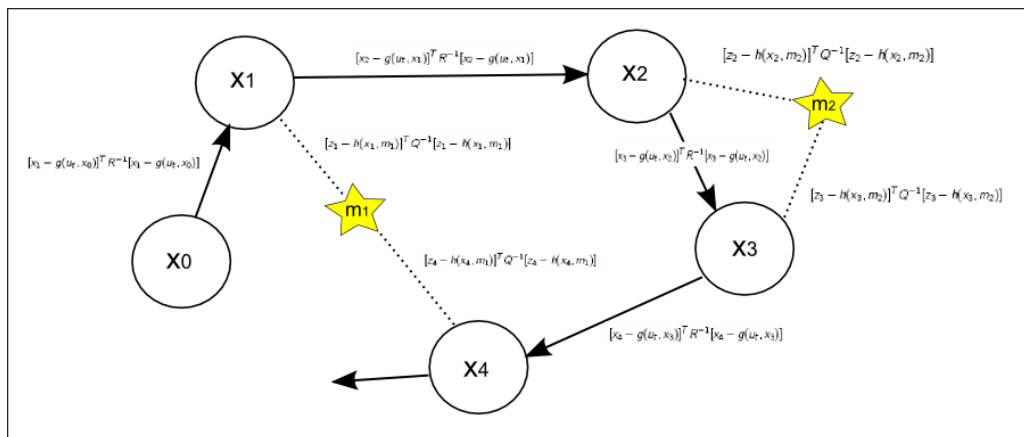


Abbildung 3.5: Graph SLAM Darstellung

Anders als ein EKF Verfahren, löst der Graph SLAM Algorithmus das vollständige Slam-Problem. Der Systemzustand μ_t wird nicht, wie beim EKF zu jedem Zeitpunkt t komplett aktualisiert, sondern zunächst anhand der Messungen und der Fahrzeugbewegungen berechnet und zur beliebigen Zeit aktualisiert. Bei der Aktualisierung wird die komplette Trajektorie des Fahrzeugs (alle Positionszustände) und alle Landmarkenpositionen neu berechnet.

Das Weltmodell des SLAM Verfahrens besteht aus Knoten und Kanten, die einen Graphen bilden [Abb. 3.5]. Ändert sich während der Fahrt der Positionszustand x_t oder wird eine Landmarke m_i in der Umgebung detektiert, werden Knoten im Graph initialisiert, die einen eigenen Umgebungszustand darstellen. Die Beziehung von einem Knoten zum Nächsten wird über eine Kante dargestellt. Laut [Sebastian Thrun (2006),S.337-339] wird bei der Bestimmung des Kantentyps zwischen den Bewegungsevents und Messdatenevents unterschieden. Während ein Bewegungsevent dazu führt, dass eine Kante zwischen zwei Positionszuständen hinzugefügt wird, führt eine neue Messung der Umgebung zu einer Kante zwischen der detektierten Landmarke m_i der Karte m und dem Positionszustand x_t .

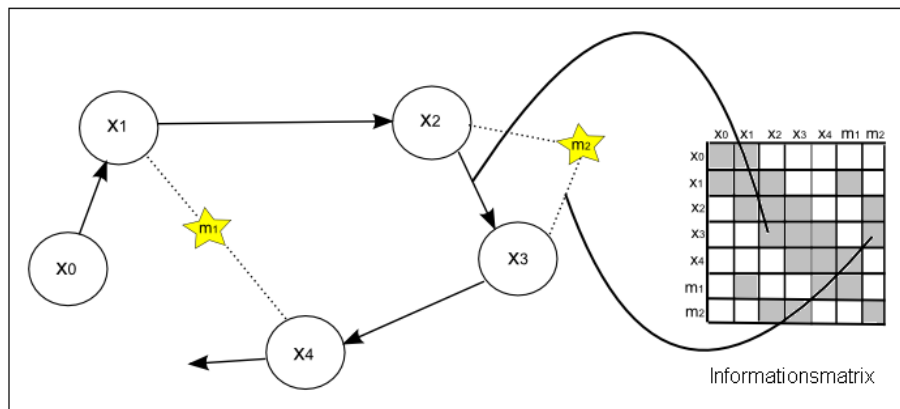


Abbildung 3.6: Aufbau der Informationsmatrix beim GraphSLAM

Jede Kante korrespondiert mit einem nicht linearen Constraint, der die Abweichungen in der Bewegung bzw. Messung angibt. Ein Constraint zwischen zwei Positionszuständen ist über die Fahrzeugbewegung u_t definiert.

$$[x_t - g(u_t, x_{t-1})]^T R^{-1} [x_t - g(u_t, x_{t-1})]$$

Wobei die Funktion $g(u_t, x_{t-1})$ das Bewegungsmodell darstellt mit dem die Fahrzeugposition x_t über die Bewegung u_t und den vorherigen Zustand x_{t-1} geschätzt wird. Die Matrix R stellt die Kovarianz des Bewegungsmodells dar. Im Vergleich dazu haben Constraints der Landmarken die Form

$$[z_t^i - h(x_t, m_i)]^T Q^{-1} [z_t^i - h(x_t, m_i)]$$

und werden anhand des Beobachtungsmodells $h(x_t, m_i)$ berechnet. Wie schon zuvor erwähnt schätzt das Modell eine gespeicherte Korrespondenzlandmarke m_i im Weltmodell m .

Der Zusammenhang der aktuellen Messung und der geschätzten Messung wird über die Kovarianz Q angegeben.

Um eine spätere Aktualisierung des Graphen zu ermöglichen, wird der Graph über eine Informationsmatrix Ω und einen Informationsvektor x_i beschrieben [Abb. 3.6]. Jede Bewegung und jede neue lokale Messung führt zu Aktualisierung der Matrix und des Vektors. Dabei werden die Constraints, die die Abweichung in der Bewegung bzw. Messung angeben in der Matrix festgehalten und die Knoten (Zustände) des Graphen im Informationsvektor.

Der Gesamtfehler des Weltmodells ergibt sich über die Summe aller Constraints.

$$J_{GraphSlam} = x_0^T \Omega x_0 + \sum_t [x_t - g(u_t, x_t - 1)]^T R^{-1} [x_t - g(u_t, x_t - 1)] \\ + \sum_t \sum_i [z_t^i - h(x_t, m, i)]^T Q^{-1} [z_t^i - h(x_t, m, i)]$$

Die Korrektur der Karte, sowie die Korrektur aller Positionszustände erfolgt über die Minimierung der Funktion J . Dazu wird die Informationsmatrix Ω , die alle Constraints enthält linearisiert und verkleinert.

3.3 SLAM mit dem Partikel Filter

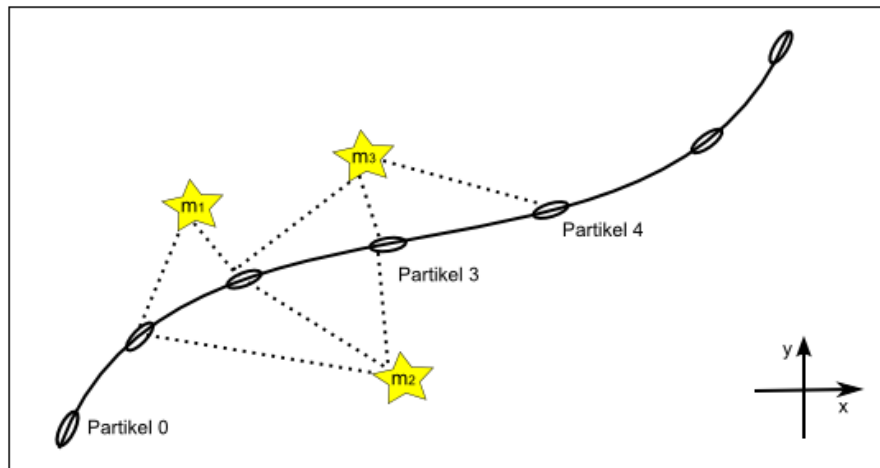


Abbildung 3.7: SLAM mit dem Partikel Filter

Ein weiterer Ansatz das SLAM Problem zu lösen ist der Partikel Filter. Im Gegenteil zum EKF- bzw. dem Graph-Algorithmus, löst ein Partikel-Filter sowohl das online als auch das offline SLAM Problem. Das Weltmodell wird über eine Menge von Partikeln definiert, die einen Bereich in der Umgebung darstellen. Jeder Partikel stellt ein eigenes Weltmodell dar, bestehend aus einem geschätzten Fahrpfad, sowie aus N klein-dimensionalen EKFs für die Schätzung der Landmarkenpositionen. Durch den Aufbau reduziert sich der Speicher- aufwand bei dem Partikel Filter auf $O(MN)$, wo N die Partikelanzahl angibt und M für die Anzahl der Umgebungslandmarken steht.

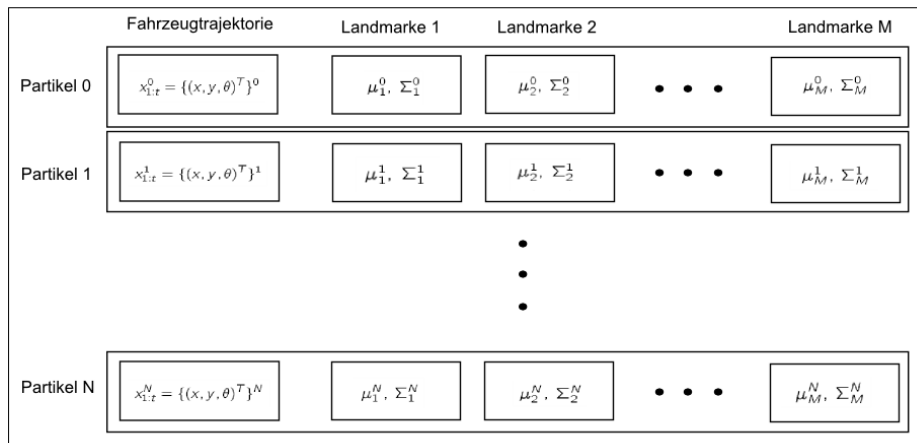


Abbildung 3.8: Weltmodell definiert über die Menge der Partikel

Jeder Partikel bewertet eigenständig die Umgebung, indem für jede neu aufgenommene Landmarke die Korrespondenzlandmarke m_i im Weltmodell bestimmt wird. Anschließend wird die neue Messung in den korrespondierenden EKF aufgenommen und der Landmarkenzustand μ_m^n , sowie die Kovarianzmatrix Σ_m^n aktualisiert. Da nur die Partikel aktualisiert werden, deren Daten mit der aktuellen Messung z_t korrespondieren, verringert sich die Komplexität des Filters auf $O(M \log N)$.

Um später eine Aussage über die bestmögliche Abschätzung zu treffen, wird jeder Partikel gewichtet. Der Partikel, dessen Landmarken die aktuelle Messung am Besten beschreiben, wird am höchsten gewichtet und beschreibt den Systemzustand y_t .

4 Related Works

Im vorherigen Kapitel wurden die unterschiedlichen SLAM-Problematiken und die grundlegenden Methoden vorgestellt mit denen das SLAM Problem gelöst werden kann. Das Kapitel soll die vorherigen Ansätze aufgreifen und einen Überblick über den aktuellen Stand in der Forschung geben. Verfahren für die Indoor- und Outdoorumgebung werden vorgestellt, die sowohl kamerabasiert als auch sensorbasiert agieren.

4.1 Visualisierte SLAM Verfahren

In [ML. Benmessaoud (2008)] wird ein EKF-Verfahren erklärt, das ein Kamerabild als Sensor verwendet. Die Umgebungsmerkmale(Landmarken) werden nicht durch Entfernungsmesssystemen bestimmt, sondern aus dem Bild extrahiert. Kanten, Ecken und Regionen im Bild, definieren robuste Landmarken. Das Extrahieren erfolgt mittels dem "Harris Corner Detection" Algorithmus.

Als Fahrzeug wird ein Roboter verwendet, auf dem eine schwenkbare Kamera (sowohl in horizontaler als auch in vertikaler Richtung) angebracht ist. Der Vorteil einer schwenkbaren Kamera besteht darin, dass mehr Landmarken in der Umgebung gefunden werden als bei einer statischen Kamera. Eine größere Landmarkenanzahl optimiert die Systemzustandesschätzung. Der Systemzustand y_t ist Abhängig von der Zeit t und ist definiert durch:

$$y_t = \begin{pmatrix} R_t \\ \bar{L}^{g.t} \end{pmatrix}$$

wo $R_t = (x_t y_t z_t)^T$ die globale Position des Fahrzeugs darstellt und $\bar{L}^{g.t} = (L_1 L_2 \dots L_N)$ ein Vektor aller globaler Landmarken $L_i^{g.t} = (L_x^{g.t} L_y^{g.t} L_z^{g.t})^T$.

Für das Bewegungsmodell wird die Fahrzeug- und die Winkelgeschwindigkeit verwendet.

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \phi_{t+1} \end{pmatrix} = \begin{pmatrix} x_t + v_k \cos \phi_t T \\ x_t + v_k \cos(\phi_t) T \\ \phi_t + \omega_t T \end{pmatrix}$$

Im oberen Bewegungsmodell gibt v_k die Translationsgeschwindigkeit und ω_t die Rotationsgeschwindigkeit in der Zeitperiode T an.

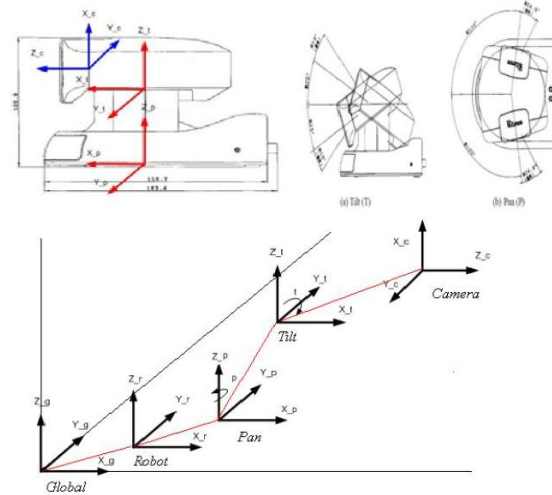


Abbildung 4.1: Unterschiedliche Koordinatenframes beim vSLAM⁴

Der Einsatz einer schwenkbaren Kamera führt zu mehreren 3-dimensionalen Koordinatenframes (Global, Roboter, Fahrzeug, Neigung, Schwenk, Kamera), dargestellt in [Abb. 4.1]. Damit Kartendaten mit den Bilderdaten abgeglichen werden können, werden die globalen Koordinaten der Landmarken $L_i^{g.t} = (L_x^{g.t} L_y^{g.t} L_z^{g.t})^T$ in das kamerabasierte Koordinatensystem $L_i^{c.t} = (L_x^{c.t} L_y^{c.t} L_z^{c.t})^T$ mittels einer Projektionsmatrix überführt. Die für die Berechnung verwendeten Projektionsmatrizen lassen sich in [ML. Benmessaoud (2008)] einsehen. Aus diesen Koordinaten werden entsprechen die Bildkoordinaten berechnet, wodurch das Beobachtungsmodell

$$I = h(R, Lg)$$

zu einer Position und den globalen 3D-Koordinaten die entsprechenden Bildkoordinaten $I = (u, v)$ liefert. Die Überführung von Bildkoordinaten in globale 3D Koordinaten, wird durch das inverse Beobachtungsmodell bestimmt.

$$L_g = h^{-1}(R, I_1, R', I_2)$$

Das inverse Modell verwendet zwei Bilderframes I_1 und I_2 , aufgenommen von unterschiedlichen Positionen R, R' .

⁴ML. Benmessaoud (2008)

Die Abarbeitung der Kartierung erfolgt in vier Schritten.

1. Initialisierung

Initialisierung des Systemzustandsvektors und der Kovarianzmatrix mit den Startkoordinaten $R_0 = (0 \ 0 \ 0)^T$ des Roboters und den am Anfang erkannten Feature $L_1 \dots L_N$.

$$y_0 = (R_0 \ L_1 \ L_2 \ \dots \ L_N)^T$$

2. Prediction

Vorhersage des erwarteten Systemzustandes y_{t+1} unter Zunahme des alten Systemzustand y_t und der Bewegung u_t und Vorhersage der neuen Kovarianzmatrix Σ_{t+1} .

$$y_{t+1} = g(y_t, u_t)$$

3. Update

Korrigieren des Systemzustandes mittels des Beobachtungsmodells. Dabei ist I_k die aktuell aufgenommene Landmarke in der Umgebung und \hat{I}_t , die vom Beobachtungsmodell ausgegebene Landmarke. Durch die Kalman Verstärkung Ka wird die Schätzung des neuen Zustandes optimiert.

$$y_{t+1} = y_{t+1} + Ka(I_t - \hat{I}_t)$$

4. Map Management

Gibt es keinen Zusammenhang zwischen einer aktuell aufgenommenen Landmarke und einer Landmarke im Systemzustandsvektor, so erweitert das Map Management den Zustandsvektor und die Kovarianzmatrix um die neue Landmarke. Die globalen Koordinaten der Landmarke werden wie oben erwähnt durch das Inverse-Beobachtungsmodell bestimmt.

4.2 Bearing Only SLAM Verfahren

Wird bei der Kartierung eine Kamera als Sensor verwendet, hat man die Problematik der fehlenden Distanzinformationen bzw. der fehlenden Tiefeninformationen im Bild. Ohne diese Informationen ist es schwierig eine Positionierung im Raum abzuschätzen und so Landmarken im Raum zu definieren. Das Bearing only SLAM Verfahren gilt in der Literatur als

mögliche Lösung für das Problem. Dabei extrahieren Verfahren spezielle Keypoints aus den Bilderdaten, die gegen Translation und Skalierung konsistent sind. Um die Tiefeninformation zu erhalten werden in [T. Lemaire (2005)] zu diesen Punkten dreidimensionale Richtungsvektoren abgebildet. Die Tiefeninformation wird laut [T. Lemaire (2005)] über eine Gleichverteilung von Gaußfunktionen entlang des Vektors repräsentiert. Durch mehrere fortlaufende Bilder überlagern sich die Vektoren mit den gleichverteilten Gaußfunktionen, sodass sich die Landmarkenpositionen approximieren lassen.

4.2.1 EKF-basierter Ansatz

Das Verfahren in [Siegfried Hochdorfern (2007)] verwendet für die Systemschätzung nur einen erweiterten Kalman Filter, der unechte Landmarken aus dem Kamerabild benutzt. Die Landmarken werden durch den Scale Invariant Feature Transform (SIFT) Algorithmus extrahiert. Eine omnidirektionale Kamera dient als Sensor. Die Kamera hat den Vorteil, dass man durch einen hyperbolischen Spiegel einen 360 Grad Blickwinkel erhält, zu sehen in [Abb. 4.2]. Ein neu extrahiertes SIFT Feature wird zunächst mit den Features in einer Datenbank abgeglichen und dadurch klassifiziert (neue Landmarke, nicht initialisierte Landmarke, initialisierte Landmarke). Der Systeminput für den Kalman-Filter wird über die ID der Landmarke, den Peilungswinkel zu der Landmarke und die getätigte Fahrzeugbewegung angegeben.

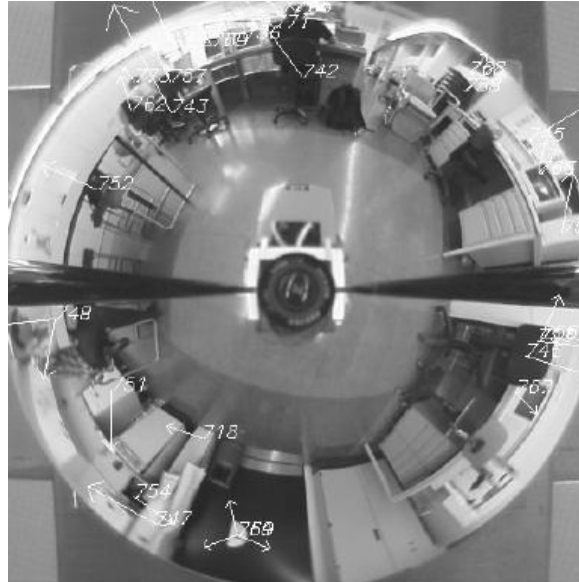


Abbildung 4.2: Omnidirektionale Kameraansicht⁵

⁵Siegfried Hochdorfern (2007)

4.2.2 PF-basierter Ansatz

Anders als [Siegfried Hochdorfern (2007)] verwendet [Hauke Strasdat (2007)] anstelle einer omnidirektionalen Kamera, eine monokulare festmontierte Kamera auf einem Roboter. Die Landmarken werden durch eindeutige Bildpunkte dargestellt. Die Bildpunkte sind eindeutig und werden über den Speeded-up visual feature (SURF) Algorithmus, der in [Herbert Bay (2008)] erläutert wird, aus dem Bild extrahiert. Für die Auffindung der Punkte dient der Fast-Hessian Detector, welcher auf der Approximation der Hessianmatrix eines Bildes basiert. Im Beobachtungsmodell werden die SURF Feature in sphärischen Koordinaten angegeben.

$$z = (\phi, \theta, \rho)^T$$

Die benötigten Winkel (Polarwinkel ϕ , Azimutwinkel θ) werden durch Pixelkoordinaten im Bild bestimmt. Die Entfernung ρ zum Feature wird zunächst ignoriert, da sie nicht berechnet werden kann. Später wird diese unter Zunahme mehrerer Bilder und einer Vektorüberlagerung approximiert. Für die Schätzung des Systemzustandes (Fahrzeugposition, sowie globale Landmarkenpositionen) wird der Partikel Filter verwendet.

Für den Abgleich der gespeicherten Landmarken, werden die globalen Landmarkenkoordinaten des Weltmodell $m = (x, y, z)$ in sphärische Kamerakoordinaten $z = (\phi \ \theta \ \rho)^T$ überführt. Das Verfahren ermöglicht damit den Abgleich der Landmarken ohne eine Distanzinformation. Die Wahrscheinlichkeit des Zusammenhangs einer gespeicherten Landmarke und einer neuen Messung, wird nur mittels der zwei Winkel berechnet. Wird die Beobachtungswahrscheinlichkeit der Übereinstimmung als zu gering bewertet, kommt es zur Neudefinition der Landmarken, indem die Tiefeninformation ρ über mehrere Bilder approximiert und die sphärischen Kamerakoordinaten in kartesische Kartenkoordinaten projiziert werden. Die Partikel des Partikel Filters werden durch die Beobachtungswahrscheinlichkeit gewichtet, wodurch die optimale Roboterposition auf dem Pfad durch den Partikel mit der höchsten Gewichtung angegeben wird.

4.3 FAST-SLAM Verfahren

Beim FAST SLAM Algorithmus der in [M. Montemerlo (2002)][M. Montemerlo (2003a)][J.Z. Sasiadek (2008)] benutzt wird, handelt es sich um einen Rao-Blackwellized Partikel Filter Algorithmus, der den kompletten Fahrpfad in einer Outdoor-Umgebung schätzt. Das System wird durch die Menge von Partikeln definiert. Die Aktualisierung des Systemzustandes (Landmarken μ und Pfad s^t) erfolgt mittels der Fahrzeugbewegung (Odometrie) u_t , den Umgebungsmessungen z_t und den observierten Landmarken m_t zur Zeit t .

$$p(\mu, s^t | z^t, u^t, m^t) = p(s^t | z^t, u^t, m^t) \prod p(\theta_\mu | s^t, z^t, u^t, m^t)$$

Ein Fast-SLAM Algorithmus wird über die a-posteriori Wahrscheinlichkeit des Fahrpfades, sowie aller Landmarkenpositionen beschrieben. Dabei gibt der erste Term das Bewegungsmodell und der zweite das Beobachtungsmodell an. Der Algorithmus wird insbesondere für große Outdoor-Umgebungen entwickelt, in denen rein EKF-basierten SLAM Verfahren wegen der hohen Landmarkenanzahl scheitern.

Die Abarbeitung ist in der [Tab. 4.1] zu sehen.

FASTSLAM-Algorithmus
<pre> for(Partikel k bis Partikel N) Berechnung des Positionsvektors über das Bewegungsmodell $x_t^k = g(x_{t-1}^k, u_t)$ Bestimmung der Landmarkenpositionen über das Beobachtungsmodell Bestimme für jede Messung z_t^i eine Korrespondenzlandmarke l_i im Partikel N Integriere z_t^i und aktualisiere den Landmarkenzustand μ_i^k und die Kovarianzmatrix Σ_i^k Gewichten des Partikels k Bestimme die Wahrscheinlichkeit w_t^k, dass von der Position x_t die Sensordaten z_t beobachtet wurden. endfor $y_t = 0$ for(Partikel k bis Partikel N) bestimme Partikel k mit der höchsten Wahrscheinlichkeit w_t^N und aktualisiere Systemzustand y_t $y_t = x_t \cup \langle x_t^k, \langle \mu_{1,t}^k, \Sigma_{1,t}^k \rangle, \dots, \langle \mu_{N,t}^k, \Sigma_{N,t}^k \rangle \rangle$ endfor return y_t </pre>

Tabelle 4.1: FASTSlam-Algorithmus

In [M. Montemerlo (2003a)] wird das FAST SLAM Verfahren benutzt um den Vectoria Park zu kartieren. Um möglichst viele Informationen aus der Umgebung zu extrahieren und um die Landmarken in der Umgebung zu detektieren, wird ein Laserscanner verwendet. [Abb. 4.3] veranschaulicht den Unterschied der Schätzung eines Fast-SLAM Verfahrens im Gegensatz zu einem EKF-basierten Verfahren. Auf der ersten Abbildung ist die reine Positionsbestimmung anhand des Bewegungsmodells zu sehen. Die anderen Abbildung stellen die Ausgabe eines EKF und eines Fast-SLAM Algorithmuses dar. Dabei wird die Schätzung des Systemzustandes mit einem kleinen und einem großen Odometriefehler durchgeführt. Der blaue Pfad stellt den Positionsgraphen dar, der anhand der GPS Daten bestimmt wurde. Der rote Pfad steht für den geschätzten Positionspfad. Der Vorteil des FAST-SLAM Verfahrens ist, dass die Position sogar komplett ohne Bewegungsmodell geschätzt werden kann, wenn die Landmarken im Weltmodell gespeichert sind.

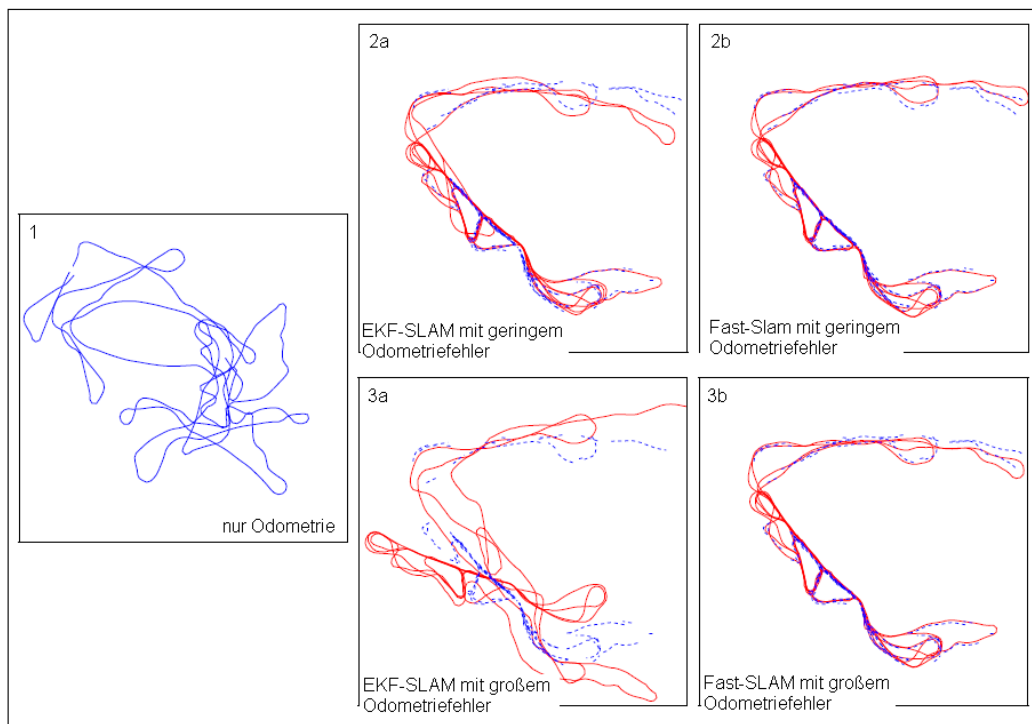


Abbildung 4.3: Fast-SLAM und EKF-Slam im Vergleich⁶

⁶M. Montemerlo (2003b)

4.4 Multihypothesen Ansätze

Ein weiterer SLAM Ansatz für große Outdoor Umgebungen wird in [Michael Bosse (2003)], [Michael Bosse (2004)], [Michael Bosse (2008)] erläutert. Zum Einsatz kommt ein Framework mit einem schmal-skaliertem Algorithmus, der sowohl feature- als auch scanbasiert arbeitet. Vergleichbar mit einem Partikel Filter handelt es sich beim so genannten ATLAS Framework ebenfalls um ein multihypothesen Ansatz. Das kartierte Weltmodell wird nicht in einem globalen Koordinatensystem, sondern in einem Graphen von lokalen Frames repräsentiert. Jeder lokale Frame implementiert ein eigenes lokales Koordinatensystem und bildet einen Knoten im Graph. Die Frames enthalten zudem neben den lokalen Messdaten die globale Frameposition und eine Hypothese über den Systemzustand.

Die Hypothese stellen den Zusammenhang zwischen Framedaten und den aktuellen aufgenommenen Messungen auf. Der Frame, der eine aktuelle Umgebungsmessung am besten beschreibt, wird aktiviert und aktualisiert. Bei der featurebasierten Lösung implementieren die Knoten dabei einen eigenständigen EKF. Scanbasierte Lösungsansätze greifen auf Scan-Match Algorithmen wie den ICP Algorithmus aus [Lu und Milios (1997)] auf, um den Fehler zwischen der aktuellen Messung und den Framedaten zu bestimmen und damit die Positionierung des Fahrzeugs zu korrigieren.

Die Frames im Weltmodell haben eine festdefinierte Größe. Ein Stammprozess entscheidet, ob in ein benachbarten Frame übergegangen oder ob ein neuer Frame erzeugt werden soll. Kanten, die die Frames miteinander verbinden, repräsentieren Transformationen zwischen den Frames. Wird ein neuer Knoten in den Graphen aufgenommen, wird nach anderen Knoten im Graph gesucht, die die Eckpunkte verknüpfen. Da mehrere Pfade zu dem jeweiligen Knoten existieren können, wird der Dijkstra Shortest Path Algorithmus verwendet um die Doppeldeutigkeit zu beheben.

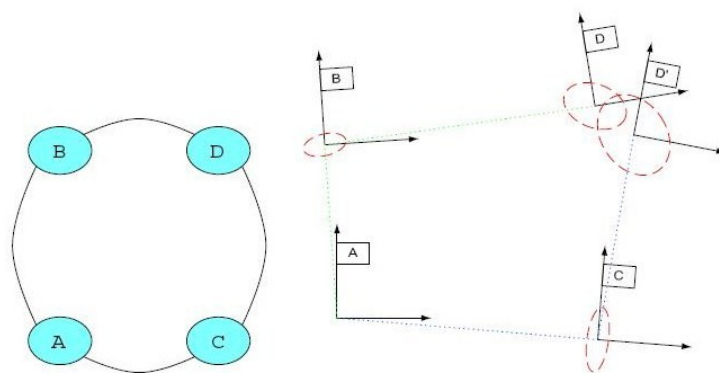


Abbildung 4.4: ATLAS Dijkstra-Projektion⁷

⁷Michael Bosse (2003)

[Abb. 4.4] zeigt ein Beispiel für eine Doppeldeutigkeit der Transformationen T_a^d von Knoten A zum Knoten D. Der kleinste Pfad ist durch den Knoten B gegeben, was durch die kleinere Fehlerellipse angezeigt wird. Das heißt Metrik ρ der Transition T_a^c ist besser als der Transition T_a^b . Für die Dijkstra-Projektion wird eine statistische Metrik ρ benutzt, die auf Wahrscheinlichkeit in den Transformationen der Kanten beruht. Die Metrik ρ ist die Determinante der Kovarianzmatrix der zusammengesetzten Transformation.

$$T_a^d = T_a^b \oplus T_b^d$$

$$\sum_{ad} = J_1(T_a^b, T_b^d) \sum_{ab} J_1(T_a^b, T_b^d)^T + J_2(T_a^b, T_b^d) \sum_{bd} J_2(T_a^b, T_b^d)^T$$

$$\rho = \det(\sum_a d)$$

Die Zusammensetzung der Transformation lässt sich mittels dem \oplus Operator berechnen. Für die Berechnung der Kovarianzmatrix \sum_{ad} werden die Jacobi Matrizen J_1 und J_2 verwendet, die die Ableitungsmatrizen der Transformationsfunktionen angeben.

Die Positionsschätzung und Aktualisierung im Frame und somit der Fahrzeugposition erfolgt bei der landmarkenbasierten Lösung durch den EKF im Frame. Bei scanbasierter Implementierung erfolgt die Aktualisierung durch benachbarte Frames. [Abb. 4.5] veranschaulicht die Aktualisierung der Position. Die aktuelle Position ist durch x_j gekennzeichnet. Der benachbarte Frame i wird aufgefordert seine Position x_i^{old} entsprechend zu aktualisieren x_i^* .

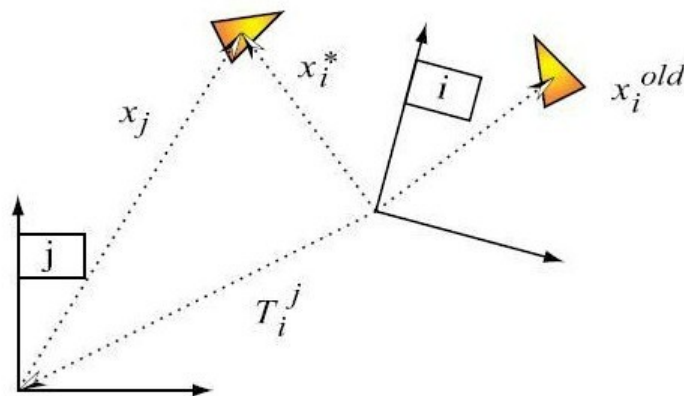


Abbildung 4.5: Positionskorrektur durch benachbarte Frames⁸

⁸Michael Bosse (2003)

5 Fahrspurbasiertes SLAM

In den vorherigen Kapitel wurde das allgemeine Problem einer gleichzeitigen Positionsbestimmung und Kartierung (SLAM) erläutert und Verfahren vorgestellt, die das "simultaneous localization and mapping" in einer Indoor- bzw. einer Outdoor-Umgebung ermöglichen. Alle Verfahren abstrahieren dabei Merkmale aus der gesamten Umgebung anhand der sich das Fahrzeug in der Umgebung wiederfindet und neu ausrichtet.

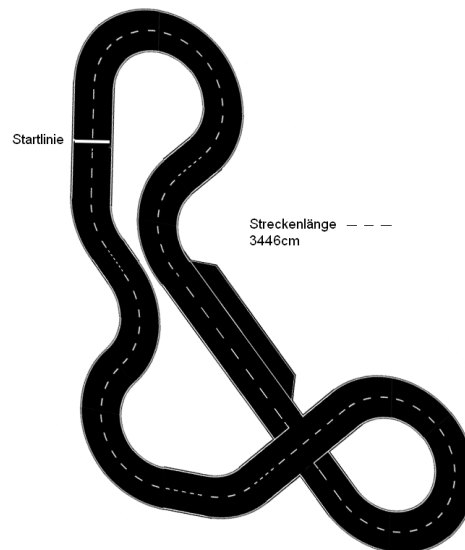


Abbildung 5.1: Die Einsatzumgebung

Bei den Verfahren werden zunehmend holonomen Fahrzeugen eingesetzt. Holonome Fahrzeuge sind in der Bewegung nicht eingeschränkt, was z.B. eine Bewegungen im Stand ermöglicht. Bei der Weltmodellierung ist das ein großer Vorteil gegenüber einem nicht holonomen Fahrzeug. Erkennt das Fahrzeug mit einer gewissen Wahrscheinlichkeit ein Merkmal wieder, kann die Wahrscheinlichkeit erhöht werden, indem sich das Fahrzeug um die eigene Achse dreht und mehrere Merkmale detektiert, die mit den gespeicherten Daten korrespondieren. Damit können die Fahrzeuge die gesamte Umgebung analysieren und eine Menge eindeutiger Umgebungsmerkmale/Landmarken abstrahieren.

Anders wie bei den vorher beschriebenen Verfahren, soll das Verfahren dieser Arbeit mit einem nicht holonomen Fahrzeug eine aufgeklebte Fahrbahn, wie in [Abb. 5.1] zu sehen, in einer Indoor Umgebung kartieren. Da nicht die komplette Umgebung kartiert wird, agiert die Kartierung fahrspurbasiert.

Im folgenden Kapitel wird zunächst das Einsatzfahrzeug mit der verbauten Sensorik vorgestellt. Daraufhin werden Anforderungen und Probleme diskutiert, die bei der fahrspurbasierte Weltmodellierung auftreten und ein offline Verfahren entwickelt, das eine gleichzeitige Lokalisierung und Weltmodellierung ermöglicht. Das Verfahren soll das vollständige SLAM Problem anhand aller Fahrzeugbewegungen $u_{1:t}$ sowie aller Umgebungsmessungen $z_{1:t}$ lösen, indem eine Verbindung zwischen den beobachteten Umgebungsdaten z_t und den Bewegungsdaten u_t aufgestellt wird. Wie bei dem ATLAS Framework aus [Michael Bosse (2008)] und dem FAST-SLAM Verfahren aus [M. Montemerlo (2003a)] wird das Weltmodell über eine Menge der erreichten Positionszustände x_t definiert, die zusammen die Bewegungstrajektorie des Fahrzeugs darstellen.

Da die Umgebung nur wenige eindeutige Merkmale bietet, beschränkt sich die Positionsschätzung des Verfahrens auf das Bewegungsmodell des Fahrzeugs.

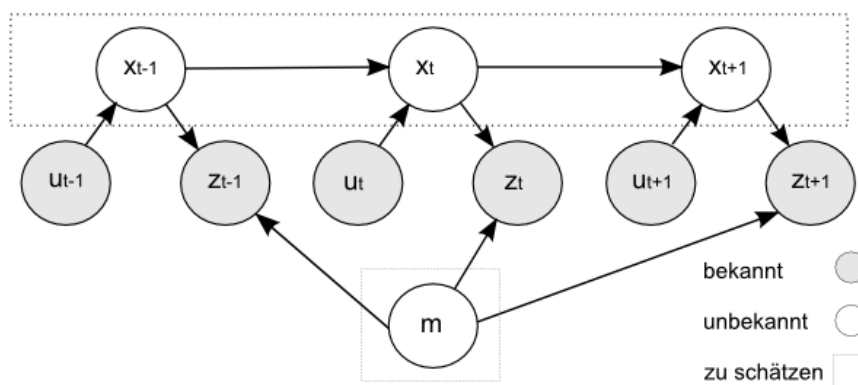


Abbildung 5.2: Das vollständige Slam Problem

5.1 Einsatzfahrzeug und Sensorik

Für die Kartierung der Fahrbahn wird ein holonomes Modellfahrzeug verwendet, das auf einem Ford F-350 Pickup-Modell im Maßstab 1:10 basiert [Abb. 5.3]. Das Fahrzeug wurde an der HAW Hamburg von Studenten entwickelt und konstruiert um im FAUS Kontext das autonome Fahren zu erforschen.



Abbildung 5.3: Das Einsatzfahrzeug

Die Sensor- und Aktorsteuerung erfolgt über drei ARM 7 Prozessoren mit je einer IO-Einheit. Die ARM Module sind über die USB Schnittstelle an einen Acer Aspire One Netbook mit einem 1.6GHz schnellen Prozessor angeschlossen und werden von diesem gesteuert. Auf dem Netbook ist ein Echtzeit Linux Betriebssystem installiert. Zusätzlich zu den ARM Prozessoren, ist eine Kamera auf dem Fahrzeug verbaut, die für die Fahrbahnerkennung zuständig ist.

Das Fahrzeug stellt folgende Sensorik bereit:

- je zwei Infrarotsensoren nach rechts und links
- je zwei Ultraschallsensoren vorne und hinten
- Inkrementalgeber an den Vorderrädern [Abb. 5.4]
- eine monokolare Kamera [Abb. 5.4]
- optische Hinderniserkennung

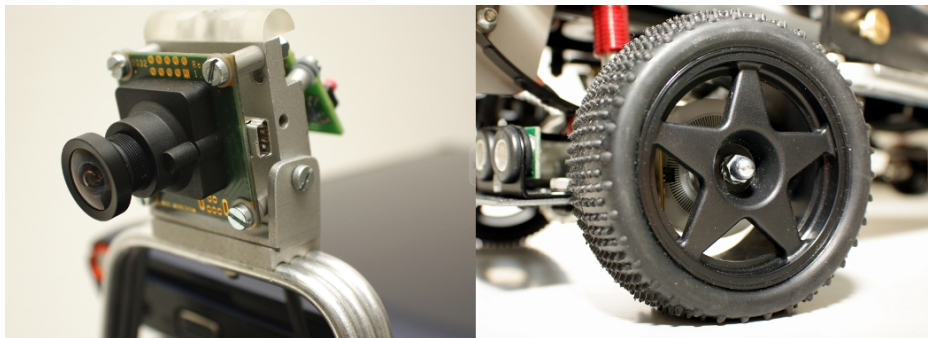


Abbildung 5.4: Eingesetzte Sensorik

Die Steuerungssoftware besteht aus einem Kernmodul und mehreren Plugins. Das Kernmodul, auch FAUSTcore genannt, übernimmt die Aufgabe der Aktivierung und Abarbeitung der einzelnen Plugins. Plugins sind Funktionsmodule, die Assistenzarbeiten durchführen, wie bspw. Fahrzeugregelung, Bildverarbeitung, Spurführung und Kartierung. Die Architektur basiert auf der Subsumption Architektur aus [Brooks (1986)] und ist in [Abb. 5.5] zu sehen.

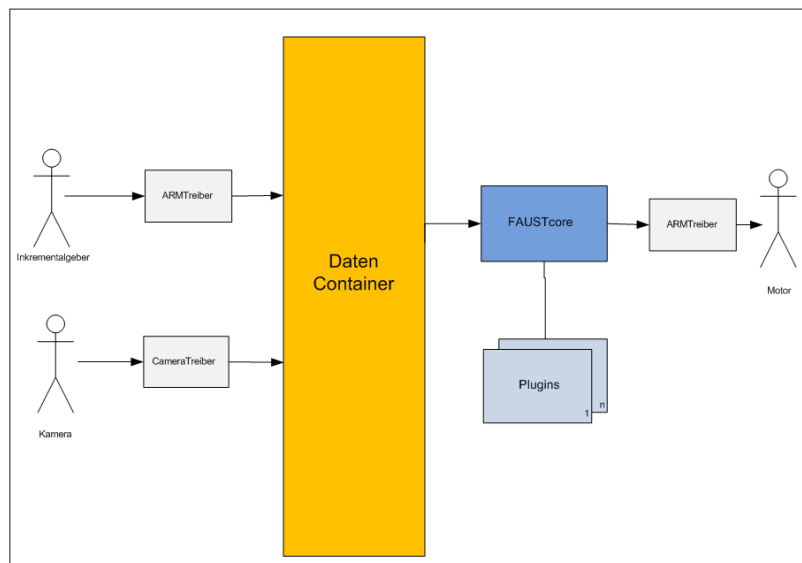


Abbildung 5.5: Softwareaufbau

Als zentrales Modul für die Fahrspurerkennung wird das POLARIS-Modul, das von [Jenning (2008)] entwickelt wurde, eingesetzt. POLARIS ist ein kamerabasiertes Fahrspurerkennungssystem, das die Fahrbahndaten aus dem Kameramodell in ein zwei-dimensionales Fahrzeugmodell transformiert und den aktuellen Streckenverlauf über zwei kubische Polynome (linke Fahrbahnmarkierung und rechte Fahrbahnmarkierung) beschreibt. Für das Bewegungsmodell werden die verbauten Inkrementalgeber beider Räder eingesetzt. Bei der

Fahrt werden die Impulse der Inkrementalgeber gezählt und der zurückgelegte Fahrweg ermittelt.

5.2 Problemstellung und Anforderungen

Ein großes Problem des Einsatzfahrzeugs ist, dass es sich nur anhand von Kameradaten und Sensordaten auf der Fahrstrecke orientiert. Dabei wird die Streckenführung wie zuvor beschrieben mittels des POLARIS Systems erfasst und der Regeleinheit in Form von Polynomen zu Verfügung gestellt. Die Regelung des Fahrzeugs funktioniert somit nur, solange die Kamera korrekte Daten liefert. Werden die Fahrbahnmarkierungen nicht richtig oder gar nicht erkannt, versagt das System und ist für eine autonome Weiterfahrt ungeeignet.

[Tab. 5.1] stellt weitere Fehlerszenarien beim Einsatzfahrzeug ohne ein Kartierungssystem dar.

Fehlerszenarium	Problematik	Konsequenz
schlechte Lichtverhältnisse	inkorrekte Streckendaten	Totalausfall d. autonomen Fahrt
Sonneneinstrahlung	fehlerhafte POLARIS Polynome, inkorrekte Streckendaten	Fahrzeugregelung in der Zeit τ nicht möglich
Kameraausfall	keine Kameradaten	Totalausfall d. autonomen Fahrt
Ultraschall-Sensorausfall	Hindernisdaten beschränkt vorhanden	Ausweichen nur mittels Linienlaserdaten möglich
Linienlaserausfall	Hindernisdaten beschränkt vorhanden	Ausweichen nur mittels Ultraschalldaten möglich
Abkommen von der Strecke	POLARIS Daten fehlerhaft	Totalausfall d. autonomen Fahrt

Tabelle 5.1: Fehlerszenarien beim Einsatzfahrzeug

Die Idee einer Fahrspurkartierung ist solche Fehler zu kompensieren, indem dem Fahrzeug Umgebungsinformationen in Form von globalen geometrischen Daten zu Verfügung gestellt werden. Das System soll in der Lage sein Kamera- bzw. POLARIS-Ausfälle auszugleichen und der Regelung die benötigten Polynome für die autonome Weiterfahrt bereit zu stellen. Die Polynome der Fahrbahn werden nicht aus einem Kamerabild berechnet, sondern aus

der erstellten Karte. Die Aufarbeitung der Daten und die Fahrzeugregelung anhand der Kartendaten sind in der Arbeit [Nowacki (2010)] beschrieben.

Das momentan implementierte Kartierungssystem auf dem Einsatzfahrzeug basiert auf einem geometrischen Weltmodell und einer odometriebasierten Positionsbestimmung aus [Rull (2008)]. Wie in den vorherigen Kapiteln schon erwähnt, ist eine rein odometriebasierte Positionsbestimmung fehleranfällig gegenüber äußeren Einflüssen, wie Unebenheiten auf der Strecke oder Schlupf während der Fahrt. Eine Kartierung ist nur bei optimalen Verhältnissen und genauer Sensorik der Wegmessung möglich. Kommt es zu Abweichungen in den Inkrementalgebern, werden Fehler in die Fahrzeugposition eingerechnet und die lokal aufgenommenen Daten fehlerhaft in das globale Weltmodell fusioniert.

[Abb. 5.6] veranschaulicht eine rein odometriebasierte Fahrbahnkartierung mit dem Einsatzfahrzeug in der Einsatzumgebung. Die Position x_0 gibt die Startposition an. Die Position x_e stellt die Endposition dar, die mit der Odometrie berechnet wurde. Im Vergleich zur realen Fahrbahn von vorhin, sieht man in der Abbildung die Ungenauigkeit, die aus der Odometrie resultiert.

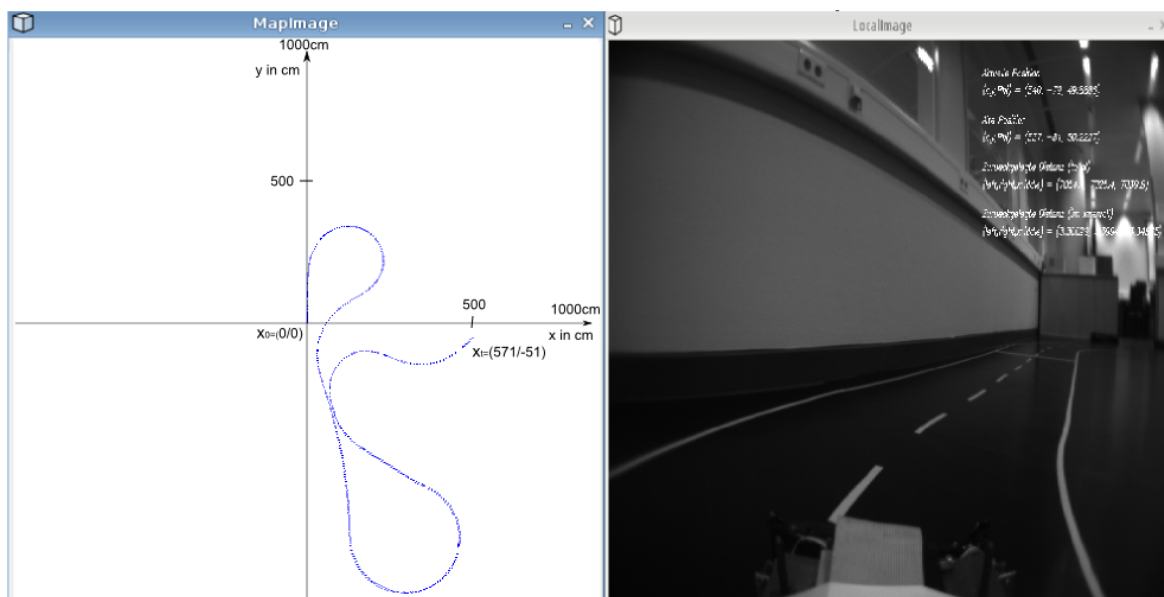


Abbildung 5.6: Fehler der Odometrie

Die ersten Tests mit dem Einsatzfahrzeug haben zudem ergeben, dass die Werte der Inkrementalgeber nicht den gewünschten Sollwerten entsprechen. Die Abweichungen resultieren aus einem Rauschen in den Inkrementalgebern, das dazu führt, dass zu viele Impulse bzw. zu wenige gezählt werden. Für ein System, das die Orientierung über die Differenz beider Wegmessungen berechnet, stellt das Problem eine Herausforderung dar.

In [Abb. 5.7] ist das Problem genauer zu sehen. Jeder Punkt im Bild definiert einen Positionszustand x_t zur Zeit t . Das Fahrzeug fährt zwar real eine Gerade, durch das Rauschen in den Inkrementalgebern ergibt sich jedoch aus den Positionszuständen eine Kurve. Unterschiedliche Verfahren [Borenstein (1996)] beheben zwar das Problem, erfordern aber zusätzliche Sensorik, wie den Gyroskop, um die globale Fahrzeugorientierung zu bestimmen. Das Verfahren dieser Arbeit soll so stabil werden, dass die Kartierung auch mit einfacher Sensorik ermöglicht wird.

Durch die Problemstellung ergeben sich mehrere Anforderungen, die bei der Weltmodellierung auf dem Einsatzfahrzeug berücksichtigt werden müssen. Wie in den anderen Verfahren, müssen zunächst Bereiche in der Umgebung definiert werden, die das Einsatzfahrzeug detektieren kann. Das Wiedererkennen von Bereichen bzw. Landmarken wird auch als loop close bezeichnet [Michael Bosse (2003)]. Je häufiger ein loop während der Kartierung festgestellt wird, desto öfter kann eine Korrektur durchgeführt werden. Auf einer Fahrbahn ist eine mehrmaliger loop close schwerer zu realisieren als in einer Umgebung mit vielen Ecken und Kanten, denn viele Streckenabschnitte korrelieren zu einander, was dazu führt, dass keine genaue Aussage über den Zusammenhang der neuen Daten mit den zuvor aufgenommenen Daten getroffen werden kann.

Die große Ähnlichkeit der verschiedenen Fahrbahnbereiche stellt bei der Kartierung eine große Schwierigkeit dar. Durch kamerabasierte Ansätze, wie in [T. Lemaire (2005)], [Siegfried Hochdorfern (2007)] oder [Hauke Strasdat (2007)], könnten zwar SIFT bzw. SURF Features im Kamerabild gefunden werden, dazu müsste aber das komplette Kamerabild analysiert und markante Bereiche extrahiert werden. Die Suche der Features würden eine hohe Rechenleistung beanspruchen, die beim Einsatzfahrzeug nicht gegeben ist.

Eine Anforderung an das System ist somit, die Wiedererkennung verschiedener Fahrbahnbereiche auch ohne Landmarken zu ermöglichen. Dazu soll die Fahrttrajektorie zunächst geschätzt und nach einer Durchfahrt korrigiert werden. Da die Korrektur der Fahrzeugposition und der Kartendaten nicht online erfolgen kann, muss ein offline SLAM Verfahren entwickelt werden, das den vollständigen Fahrpfad korrigiert. Die korrekten Daten sollen anschließend in eine globalen geometrischen Karte aus [Rull (2008)] überführt werden und eine möglichst genaue Repräsentation der Strecke darstellen.

Die Weltmodellierung muss folgende Anforderungen erfüllen.

- Streckenaufbau bestimmen
- Fahrzeugposition bestimmen
- Streckenverlauf bestimmen
- Streckenabschnitte wiedererkennen
- Fahrzeugposition korrigieren

- Streckenverlauf korrigieren

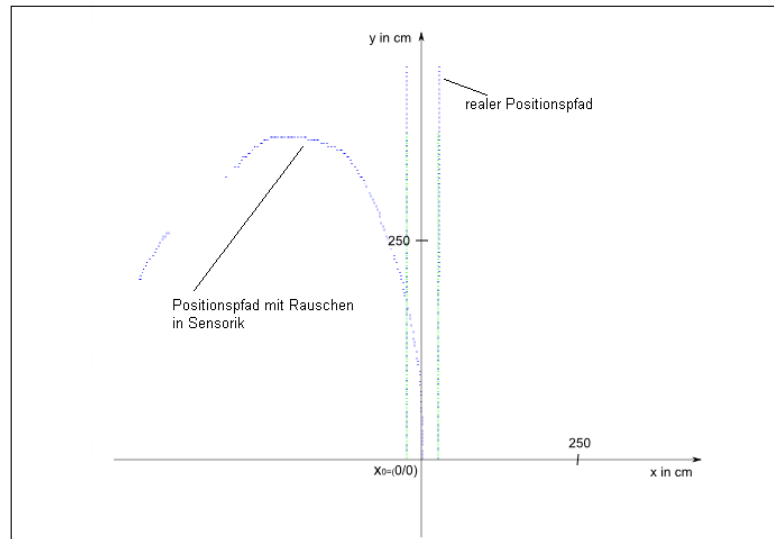


Abbildung 5.7: Fehler durch das Rauschen in der Sensorik

5.3 Topologisches Weltmodell als Grundlage

Ein geometrische Weltmodellierung in Form von Polygonen, wie in [Weber (2003)], ist in einer Umgebung mit mehreren Ecken und Kanten zwar praktisch, jedoch auf einer Fahrbahn schwierig zu realisieren. Für die Selbstlokalisierung auf einer Fahrbahn, benötigt das Einsatzfahrzeug eine Weltmodellendarstellung, mit denen sich der Fahrbahnverlauf kategorisieren lässt. Eine kategorisierte Fahrbahn hat den Vorteil, dass das Einsatzfahrzeug eine Selbstlokalisierung durchführen kann, indem Merkmale in den gespeicherten Bereichen mit den Merkmalen der aktuellen Beobachtung korrelieren.

Als Umgebungsmodell wird deshalb ein topologisches Kartenmodell entwickelt. Der Aufbau ist vergleichbar mit dem Modell aus dem ATLAS-Framework [Michael Bosse (2004)], [Michael Bosse (2004)]. Das globale Weltmodell m wird aus der Menge lokaler Karten s_t (in dieser Arbeit auch lokale Frames bzw. Scans genannt) aufgebaut. Im Unterschied zum ATLAS-Framework, wo jeder lokale Frame eine eigene Hypothese über die aktuell befahrene Umgebung aufstellt, werden die lokalen Frames dieses Verfahrens unterschiedlichen Fahrbahnbereichen zugeordnet. Die Bereiche sind für die Interpretation der Umgebung zuständig. Die Zuordnung der Frames ist notwendig, da Scans innerhalb eines Bereiches homogen zueinander sind und eine neu aufgenommene Umgebung gleich bewerten.

Die Kategorisierung der Einsatzumgebung führt zu einer topologischen Weltdarstellung. Da sich in der Einsatzumgebung mehrere Topologien gleicher Art befinden, wird die Bewertung der Topologien von einem zentralen Modul gesteuert.

5.3.1 Topologische Regionen

Die Topologie auf der Strecke definiert einen Fahrbahnbereich der Einsatzumgebung, in dem sich das Einsatzfahrzeug zur Zeit t befinden kann. In dieser Arbeit wird so ein Bereich auch als topologische Region TR oder als Knoten bezeichnet. Jeder Region werden unterschiedliche Merkmalseigenschaften zugewiesen, über die der aktuell befahrene Fahrbahnbereich bewertet wird. Die Eigenschaften sind:

- Krümmung der Fahrspurlinien
- Lenkwinkel des Einsatzfahrzeugs
- Getätigte Wegmessung
- Startlinie
- Zuvor getätigte Bewertungen

Die oberen Eigenschaften ermöglichen die Aufteilung der Einsatzumgebung in vier unterschiedliche Topologiearten. Die Fahrbahntopologien sind: Linkskurven, Rechtskurven, Geraden und Geraden mit einer Start-Stopplinie. Jedem Bereich wird eine Start- und Endposition zugewiesen, die über die Positionsvektoren $x_{k,start} = (x, y, \theta)^T$ und $x_{k,end} = (x, y, \theta)^T$ definiert werden [Abb. 5.8]. Beim der Erstellung eines Knotens k gibt die Endposition $x_{k_{i-1},end}$ des letzten Knotens k_{i-1} die Anfangsposition $x_{k_i,start}$ des neuen Knotens k_i an.

$$x_{k_i,start} = x_{k_{i-1},end}$$

Desweiteren werden zu der Anfangs- und Endposition die lokal aufgenommenen Daten in Form von lokalen Scans abgelegt. Lokale Scans sind Momentaufnahmen aus der Sicht des Fahrzeugs und stellen die Messung z_t zur Zeit t dar.

Das Weltmodell kann damit über die Menge der Zustandsvektoren betrachtet werden, in dem jeder topologischer Knoten k_i einen Zustand der Umgebung repräsentiert. Der Zustandsvektor enthält die Anfangs- und Endposition der Topologie $(x_{k_i,start}, x_{k_i,stop})$, sowie alle Scans die im Bereich aufgenommen wurden.

$$m := k_1, k_2, \dots, k_n$$

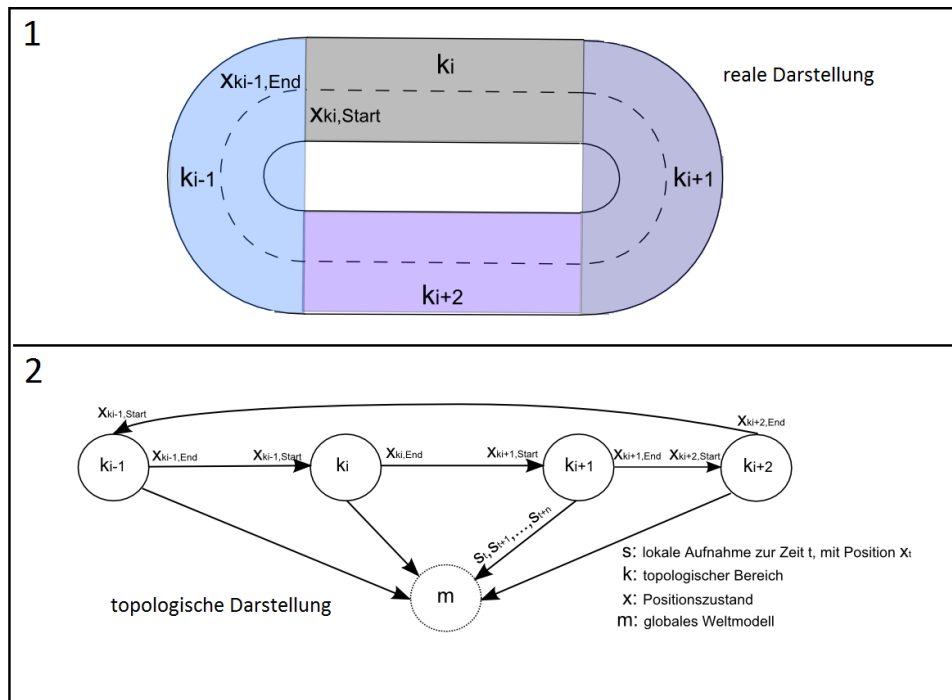


Abbildung 5.8: Topologische Einteilung der Einsatzumgebung

$$k := (x_{k,start}, x_{k,end}, s_{z_t, x_t}^t, s_{z_t, x_t}^{t+1}, \dots, s_{z_t, x_t}^{t+m})^T$$

s_{z_t, x_t}^t stellt eine lokale Umgebungsmessung z_t mit dem globalen Positionszustand x_t zur Zeitpunkt t dar.

5.3.2 Lokales Weltmodell

Das lokale Weltmodell baut sich aus den POLARIS Daten auf und wird von einem Scan dargestellt. In dem Weltmodell werden fusionierte Sensorinformationen z_t bezüglich der aktuell befahrenen Umgebung zur Zeit t abgespeichert. Das lokale Koordinatensystem orientiert sich an der Fahrzeugposition, die den Koordinatenursprung darstellt. Bei den lokalen Karte ist der Koordinatenursprung immer auf den Punkt $P(0, 0)$ festgelegt. Um die Scans später richtig zuordnen zu können, wird zusätzlich eine globale relative Fahrzeugposition x_t abgespeichert. Die Position enthält die Parameter mit denen sich ein lokaler Scan in ein globales Weltmodell abbilden lässt.

Ein lokaler Scan s_t wird über die Fahrzeugposition x_t und die aufgenommene Messung z_t zur Zeit t definiert.

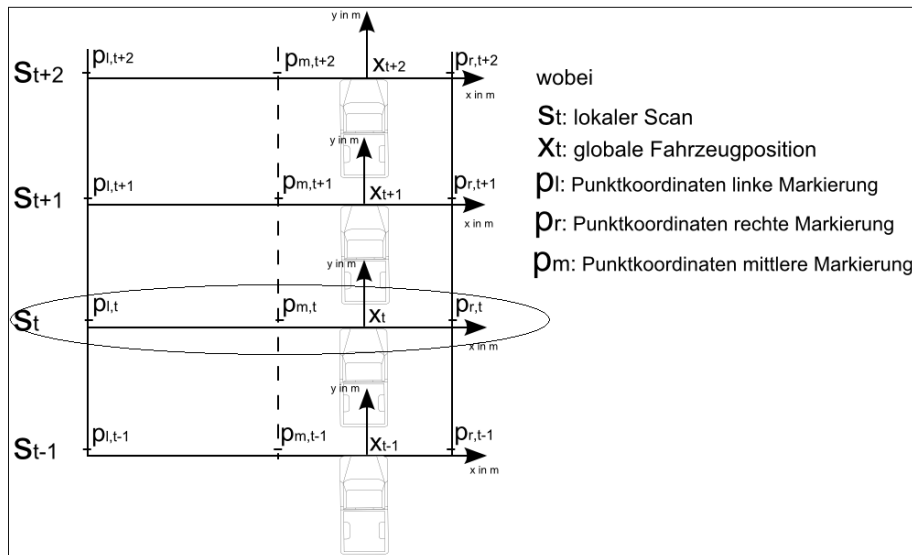


Abbildung 5.9: Lokale Weltmodellierung

$$s_t := (x_t, z_t)^T$$

Für die entwickelte Fahrspurkartierung wird eine einfache und detailarme, lokale Weltmodellierung verwendet. Pro Messung wird außer dem Positionszustand x_t nur die lokalen Koordinaten der linken, der rechten und der mittleren Fahrbahnmarkierung abgelegt, die direkt vor dem Fahrzeug liegen. Der Vektor s_t wird im weiteren Verlauf als

$$s_t := (x_t, p_{l,t}, p_{r,t}, p_{m,t})^T$$

definiert. Wobei $p_t = (x, y)$ die Koordinaten der Punkte auf der Fahrbahn widerspiegeln (linke, rechte und mittlere Fahrbahnmarkierung). Der Aufbau eines Weltmodells mit mehreren lokalen Scans ist aus der [Abb. 5.9] zu entnehmen. Die Punktkoordinaten der Fahrspur werden über die POLARIS Polynome bestimmt. Anhand der Polynome wird für einen gegebenen y Wert, der entsprechende x Wert der linken und rechten Fahrbahn berechnet. Da die Fahrbahn über drei Markierungslinien verfügt und die Fahrbahnbreite bekannt ist, können auch die Punktkoordinaten der letzten Markierung berechnet werden.

Der Detailgrad der lokalen Scans lässt sich erhöhen, indem die kompletten Fahrspurpolynome, die durch POLARIS bestimmt wurden, abgespeichert werden. Der Nachteil der dabei entsteht ist jedoch die Fehleranfälligkeit der lokalen Weltmodellierung. Schwankungen von

POLARIS und schlechte Lichtverhältnisse führen zu fehlerhaften Messung z_t , was wiederum zu einem fehlerhaften Weltmodell m führt. Jeder zusätzlich abgespeicherter Punkt erhöht zudem den Berechnungsaufwand bei der späteren Fusionierung der lokalen Daten in ein globales Weltmodell.

5.3.3 Globales Weltmodell

Anders als bei dem lokalen Weltmodell, der die aktuell befahrene Umgebung repräsentiert, beschreibt ein globales Weltmodell die gesamte Umgebung, die das Einsatzfahrzeug aufgenommen hat. Für das Weltmodell gilt die Definition:

$$m := S_t, S_{t+1}, \dots, S_{t+n}$$

wobei S_{t+i} , wie in der [Abb. 5.10] zu sehen die lokalen Scans angeben.

Der Detailgrad des globalen Weltmodells kann durch eine höhere Anzahl der aufgenommenen Scans erhöht werden, indem die Periodenzeit zwischen zwei Aufnahmen verringert wird. Zusätzlich kann der Detailgrad durch Interpolation bzw. Approximation zwischen den Punkten verbessert werden. Die Interpolation für die Verbesserung eines geometrischen Modells wird in der Arbeit [Nowacki (2010)] angewandt.

Die Zuordnung der lokalen Daten in das globale Weltmodell erfolgt anhand des gespeicherten Positionszustand x_t im Scan s_t . Da die Positionierung des Einsatzfahrzeugs immer dem Koordinationsursprung einer lokalen Karte entspricht, können die lokalen Daten des Scans $p_{l,t}^i = x_{l,i}, y_{l,i}$ durch die direkte Methode der Transformation in globale Zielkoordinaten $p_{g,t}^i = x_{g,i}, y_{g,i}$ des globalen Weltmodells überführt werden. Für die Rotation der Koordinaten gilt die Rotationsmatrix:

$$\begin{pmatrix} x_{g,i} \\ y_{g,i} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} * \begin{pmatrix} x_{l,i} \\ y_{l,i} \end{pmatrix}$$

wo θ den Orientierungswinkel des Fahrzeugs darstellt. Da die Koordinaten zusätzlich um die Positionskoordinaten $(x_{x_t}, y_{x_t})^T$ verschoben werden müssen, gilt die Transformationsmatrix:

$$\begin{pmatrix} x_{g,i} \\ y_{g,i} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} * \begin{pmatrix} x_{l,i} \\ y_{l,i} \end{pmatrix} + \begin{pmatrix} x_{x_t} \\ y_{x_t} \end{pmatrix}$$

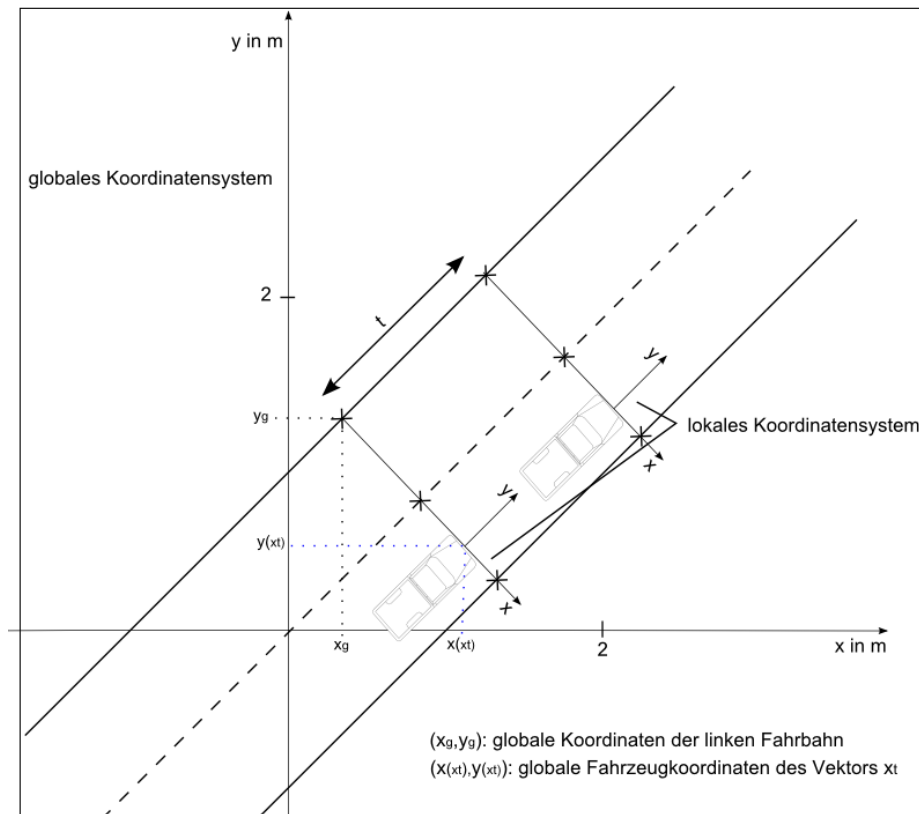


Abbildung 5.10: Globales Koordinatensystem

Bei dem Einsatzfahrzeug wird anstelle der x-Achse die y-Achse als Orientierungsursprung genommen, daher ergibt folgende Transformation:

$$\begin{pmatrix} x_{g,i} \\ y_{g,i} \end{pmatrix} = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} * \begin{pmatrix} x_{l,i} \\ x_{l,i} \end{pmatrix} + \begin{pmatrix} x_{x_t} \\ x_{x_t} \end{pmatrix}$$

5.4 Abarbeitungsphasen

Für die Realisierung des "simultaneous localization and mapping" auf einer Fahrbahn wird die Weltmodellierung in drei Phasen unterteilt. Die erste Abarbeitungsphase wird als "Controlling-Phase" bezeichnet. In der Controlling-Phase wird ein global topologischer Graph erstellt, in dem die Knoten die topologischen Bereiche k_i bilden und die Kanten die Beziehung zwischen den Bereichen in Form einer Wegmessung $u_t = (d_l, d_r)$ beschreiben. Der $u_t = (d_l, d_r)$ enthält die Wegmessung für das linke und das rechte Vorderrad, die im Zeitintervall $[t - 1, t]$ berechnet wurde.

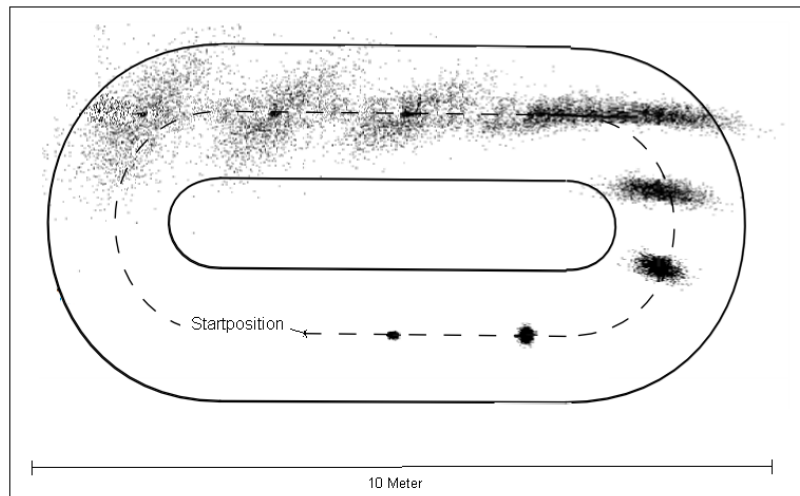


Abbildung 5.11: Samplermenge und die Fehlerellipsen

Die Aufnahme der Umgebungsdaten erfolgt in der "Prediction-Phase" in Form lokaler Punktescans. Bei der Abarbeitung wird der Zustandsvektor der Knoten k um die Scanzustände s_t erweitert. Je größer der topologische Bereich ist, desto größer ist auch der Zustandsvektor des Bereiches. Für die Lokalisierung der Scanzustände wird das Bewegungsmodell des Fahrzeugs eingesetzt. Dazu wird die Fahrzeugposition und damit auch die Position der Scans, anhand der zurückgelegten Strecke berechnet.

Um den Fehler der Odometrie aufzufangen, wird ein Sampling Algorithmus [?, S.138-140] verwendet, mit dem -für jede Positionsberechnung- mehrere mögliche Positionszustände geschätzt werden. Jede Positionsschätzung repräsentiert ein Sample in der Umgebung. Die Menge der Samples zur selben Zeit hat zur Folge, dass ein topologischer Bereich sich über eine Menge möglicher Fahrpfade definieren lassen kann.

In der [Abb. 5.11] ist die Samplermenge einer Fahrt dargestellt. Jeder Punkt in der Abbildung repräsentiert einen Scan, mit dem geschätzten Positionszustand. Mit zunehmender Fahrt, addiert sich der Schätzfehler auf, sodass die Fehlerellipsen der Scanpositionen (Samples) immer größer werden.

Die Update-Phase ist die letzte Phase der Kartierung. Bei der Abarbeitungsphase wird der bestmögliche Pfad aus der Samplermenge, die in der Prediction-Phase berechnet wurden, selektiert und der Systemzustand y_t neu berechnet. Nach der Updatephase enthält der Systemzustand nur die Scans, die den bestmöglichen Fahrpfad der Fahrbahn darstellen. Die nicht selektierten Samples werden aus den topologischen Knoten entfernt.

5.4.1 Topologische Knotenbildung (Controlling Phase)

Um eine spätere Korrektur überhaupt zu ermöglichen, muss sichergestellt werden, dass unterschiedliche Kurven und unterschiedliche Geraden auch wiedererkannt werden. Bei dem hier entwickelten Verfahren wird deshalb das globale Modell, wie zuvor schon beschrieben, in Topologien eingeteilt. Das hat den Vorteil, dass sich jeder lokale Scan einem Bereich zuordnen lässt. Jeder Bereich analysiert eigenständig die aktuelle Umgebung und stellt eine Hypothese über den Zusammenhang der aktuell aufgenommenen Daten zu den gespeicherten Daten auf. Die Region die die aktuelle Umgebung am besten darstellt, wird zur dominanten Region und definiert den Zustand der aktiven Topologie und dadurch auch den Zustand der Fahrzeugposition.

Da mehrere Regionen (bspw. zwei Geraden) zwar topologisch unterschiedlich angeordnet, in der Bewertung aber gleich sind, wird die Aktivierung der Knoten von einer zentralen Einheit gesteuert. Die zentrale Steuerung ist nötig, da eine lokale Repräsentation nichts über deren globale Anordnung in der Umgebung aussagt. So sehen zwei Scans einer Gerade gleich aus, können sich aber in unterschiedlichen Bereichen der Fahrbahn befinden.

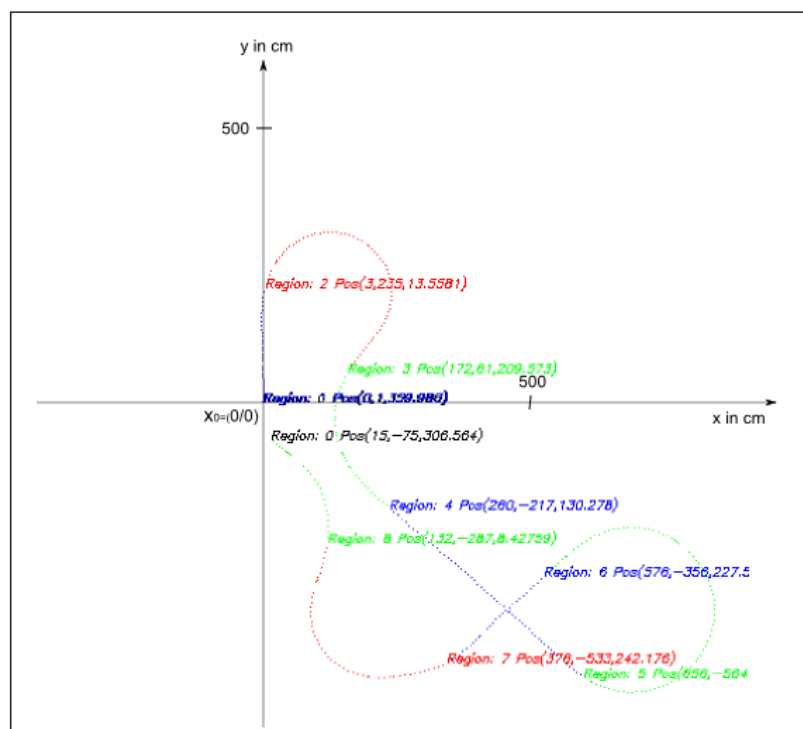


Abbildung 5.12: Topologische Bereiche in der Einsatzumgebung

Die Bildung und die spätere Aktivierung der unterschiedlichen Bereiche erfolgt über ein Be-

wertungsmodell. Die topologische Aufteilung der Bereiche in der Einsatzumgebung ist in [Abb. 5.12] zu sehen. Der erste Bereich $TR = 0$ stellt die Startregion dar, in der sich die Startlinie befindet.

5.4.1.1 Bewertung der Topologie

Das Bewertungsmodell für die Bildung und Aktivierung der Bereiche erfolgt Anhand von drei unterschiedlichen Bewertungen.

- Bewertung anhand des Beobachtungsmodells (POLARIS)
- Bewertung anhand des Bewegungsmodells (Odometrie)
- Bewertung anhand der Startlinie

Bewertung anhand der Start-Stopp Linie (Hough-Transformation)

Eine markante Stelle auf der Fahrbahn ist die Start- bzw. die Stopplinie. Der Startpunkt wird durch eine waagerechte Linie auf der Fahrbahn gekennzeichnet und bietet dem Fahrzeug eine Möglichkeit festzustellen, wenn eine Schleife (loop close) auftritt. Die Linie ist nur einmalig in der Einsatzumgebung vorhanden, dadurch hat die Bewertung anhand der Linie die höchste Priorität aller Bewertungen. Wird die Startlinie von dem Einsatzfahrzeug detektiert, ist die Wahrscheinlichkeit, dass sich das Fahrzeug in der Startregion befindet sehr hoch, wodurch die Region selektiert und der Knoten der die Startlinieninformation enthält aktiviert wird.

Die Linienenerkennung erfolgt über die Hough-Transformation, die von P.V.C. Hough entworfen und später von Duda und Hart weiterentwickelt wurde. Es ist ein Verfahren zum Detektieren von kollinearen Punkten, mit denen sich vorgegebene geometrische Strukturen in einem Bild auffinden lassen. Die gesuchte Gerade wird durch das Wertepaar (r, ϕ) beschrieben, wo r den senkrechten Abstand zum Koordinatenursprung angibt und ϕ den Winkel zwischen r und der Koordinatenachse. Der Abstand lässt sich über den Winkel ϕ und den Bildpunktkoordinaten (x, y) berechnen.

$$r = x \cdot \cos(\phi) + y \cdot \sin(\phi)$$

Um die Startlinie zu finden werden die Bildpunkt gezählt, die an derselben Geraden liegen. Die Häufigkeit des Auftretens eines gleichen Wertepaares (r, ϕ) bewertet die Wahrscheinlichkeit einer Gerade [Abb. 5.13]. veranschaulicht den Hough-Raum für die Startlinie in der Einsatzumgebung. Im optimalen Fall müssen nur Geraden mit dem Winkelwert $\phi = 90$

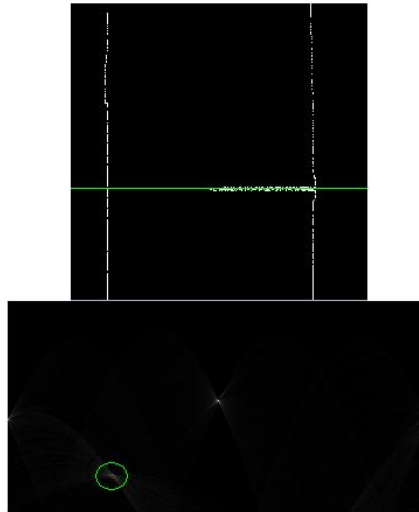


Abbildung 5.13: Die Maxima im Hough-Raum

betrachtet werden, da diese genau waagrecht auf dem Bild zu sehen sind. Da die Start-Stopplinie aber auch quer den dem Fahrzeug angefahren werden kann, wird der Winkelbereich von 80 bis 100 analysiert.

Zusätzlich zu der Startlinie muss die Fahrspur bewertet werden. Wird eine durchgehende Fahrspur erkannt, so handelt es sich um eine Startlinie. Wird jedoch eine Lücke auf der linken bzw. rechten Fahrbahn erkannt, handelt es sich um eine Kreuzung, wodurch die Umgebung nicht den Startbereich repräsentiert.

Bewertung anhand des Beobachtungsmodells (POLARIS)

Die zweite Bewertungsfunktion agiert Anhand des Beobachtungsmodells. Für jede lokale Messung z_t , die durch die Polynome der Polarisdaten repräsentiert wird, werden 20cm der Fahrbahn vor dem Fahrzeug analysiert. Für jeden Abstand $d = 2\text{cm}$ zwischen zwei Punkten wird die Steigung m bestimmt und das Arithmetische Mittel der Steigungen gebildet [Abb. 5.14]. Da die Polarispolynome den Fahrbahnverlauf vor dem Fahrzeug darstellen bewertet des Beobachtungsmodell die Umgebung vorausschauend.

$$\bar{m} = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}} \right)$$

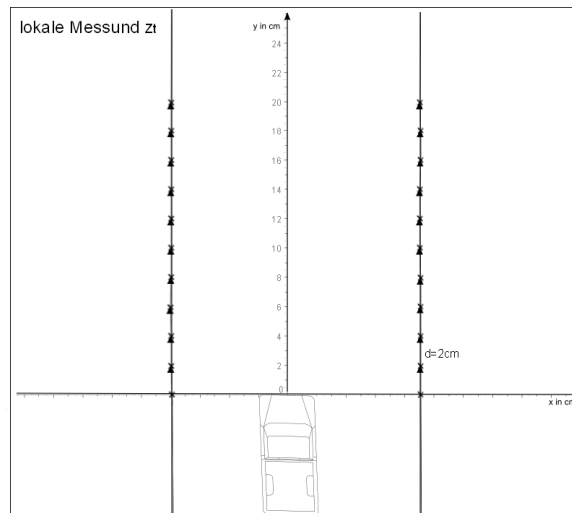


Abbildung 5.14: Bewertung über die aktuelle Beobachtung

Um das System stabiler gegenüber Polarisfehlern zu machen (fehlerhafte POLARIS-Daten), werden die letzten fünf Bewertungen zusammengefasst. Damit ist die Bewertung b definiert über das arithmetische Mittel der Summe aller vorherigen Bewertungen $b_{t-1}, b_{t-2} \dots b_{t-n}$.

$$b = \frac{1}{n} \sum_{i=0}^{n-1} \bar{b}_i$$

Ist das Arithmetische Mittel über einem erlaubtem Schwellwert $0 + \epsilon$, definiert das System die aktuelle Umgebung als Rechtskurve. Beim Unterschreiten des Schwellwert $0 - \epsilon$, wird der Bereich als Linkskurve interpretiert. Bewertungen innerhalb des Intervalls $[0 - \epsilon, 0 + \epsilon]$ repräsentieren eine Gerade.

Bewertung anhand des Bewegungsmodells (Lenkwinkel)

Die letzte Bewertungsfunktion benutzt, für die Bewertung der aktuelle Umgebung, das Bewegungsmodell des Fahrzeuges. Für die Hypothese werden die letzten zehn Lenkwinkel l des Fahrzeuges betrachtet und analysiert. Damit wird, anders wie zuvor beschrieben, keine vorhersagende Bewertung durchgeführt, sondern der aktuelle Streckenverlauf analysiert. Das Ergebnis der Bewertungsfunktion setzt sich aus dem arithmetischen Mittel der letzten zehn Lenkwinkel zusammen.

$$\bar{l} = \frac{1}{n} \sum_{i=0}^n l_i$$

Ist der berechnete Wert der Umgebung über einem Schwellwert $0 + \epsilon$, so wird die Umgebung als eine Rechtskurve betrachtet. Wird der Schwellwert $0 - \epsilon$ unterschritten, wird der Bereich als eine Linkskurve gekennzeichnet. Liegt die Bewertung innerhalb des Bereichs $[0 - \epsilon, 0 + \epsilon]$ wird die Umgebung als eine Gerade bewertet.

Zusätzlich zu dem Lenkwinkel werden die Wegmessungen des linken und rechten Rades bewertet. Unterschreitet bzw. überschreitet die Differenz der beiden Messungen den Schwellwert von $[0 - \epsilon, 0 + \epsilon]$, so wird die aktuelle Umgebung, die während der Zeitperiode t befahren wurde, als Links bzw. Rechtskurve interpretiert.

5.4.1.2 Knotenbildung und Aktivierung

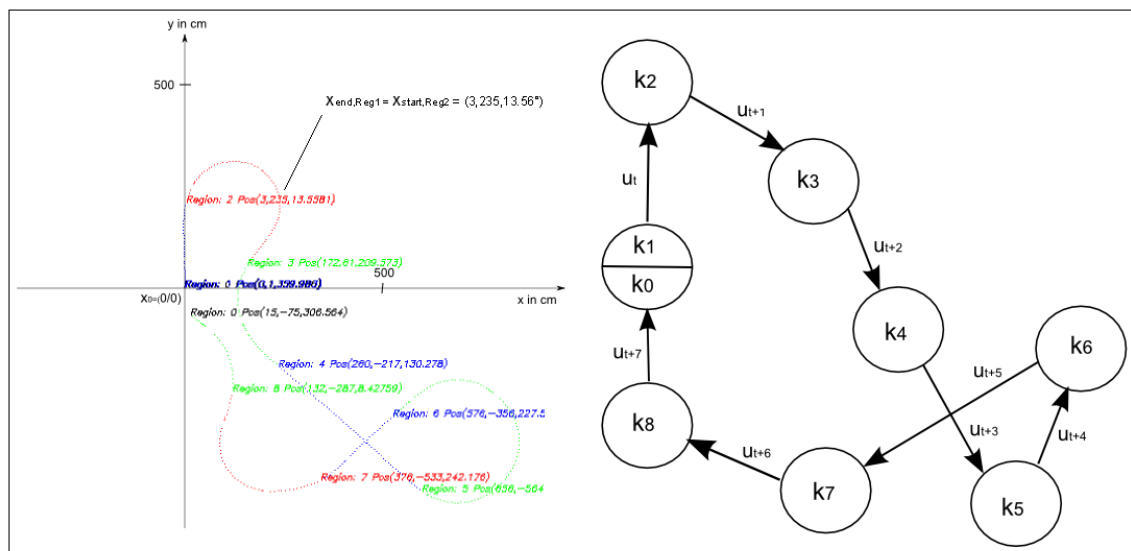


Abbildung 5.15: Knotenbildung in der Einsatzumgebung

Hat das Fahrzeug zu Beginn einer Fahrt keine Umgebungsinformation gespeichert, werden die topologischen Knoten bei der ersten Durchfahrt der Strecke generiert. Der erste Knoten repräsentiert immer die Start- und Stoppreigion und somit den Anfangsbereich des topologischen Graphen. Dazu wird das Einsatzfahrzeug so positioniert, dass die Startlinie aufgenommen werden kann. Stellt das Fahrzeug anhand des Bewertungsmodells fest, dass der aktuell aktive Knoten die Umgebung schlechter beschreibt als ein anderer Knotentyp, wird ein neuer Knoten mit dem neuen Typ generiert und als aktiv gesetzt. Der alte Knoten wird deaktiviert und die Liste der Knoten um den neuen Knoten erweitert.

Eine Neugenerierung des Knotens wird erst durchgeführt, wenn im System kein Folgeknoten gespeichert ist. Stellt das Fahrzeug bei der Fahrt fest, dass die aktuelle Umgebungsmessung

z_t die Startlinie enthält, wird ein loop close ausgeführt. Ein loop close schließt den Trajektorienpfad ab, sodass die aktuelle Region der Startregion zugewiesen und der Startknoten aktiviert wird. Zudem wird der Index, der die Identifizierungsnummer des Knotens angibt, zurückgesetzt.

[Abb. 5.15] veranschaulicht die Knotenbildung in der Einsatzumgebung. Ist ein topologischer Graph schon gegeben, wie bei der zweiten Durchfahrt, wird ein neuer Knoten nur dann erstellt, wenn der erwartete Knoten die aktuelle Umgebung schlechter bewertet als ein neuer Knoten von einem anderen Typ. [Abb. 5.16] stellt die Aufnahme eines neuen Knotens k_3 in das Weltmodell dar. Da k_3 zur Zeit t die aktuelle Umgebung besser beschreibt als der Knoten $k_3 \Rightarrow k_4$, wird die direkte Verbindung zwischen k_2 und $k_3 \Rightarrow k_4$ entfernt. Demnach wird der Knoten $k_3 \Rightarrow k_4$ erst über die Zustandsübergänge u_{t+1} und u_{t+2} erreicht. Wird die aktuelle Umgebung von dem Folgeknoten besser bewertet, so wird der Folgeknoten aktiviert.

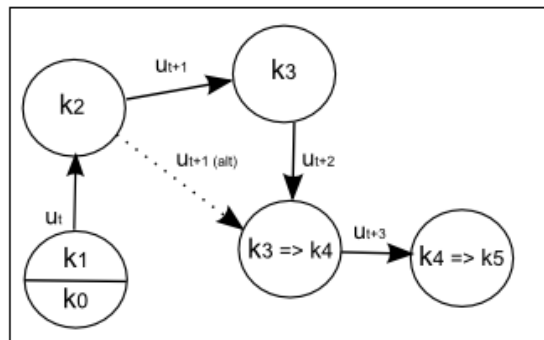


Abbildung 5.16: Erstellung neuer Knoten

5.4.2 Positionsschätzung (Prediction Phase)

Die Predictionphase ist die Abarbeitungsphase, in der die relative Fahrzeugposition x_t zur Zeit t vorhergesagt wird. In der Abarbeitungsphase werden die Umgebungsinformationen in Form von lokalen Scans abgelegt und über die Fahrzeugposition x_t global zugeordnet.

Wie zuvor schon erwähnt wird der Positionszustand durch den Vektor

$$x_t := \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

beschrieben. Die Positionskordinaten (x, y) und der globale Orientierungswinkel θ geben die Parameter für die Transformationsmatrix an, mit der die lokalen Scandaten in global Daten überführt werden. Der Positionszustand eines Scans s_{x_t} entspricht genau der Fahrzeugposition x_t ($x_t := s_{x_t}$).

5.4.2.1 Das odometriebasierte Bewegungsmodell

Zu Beginn der Fahrt, wo keine Karte gegeben ist, wird der Positionszustand über das Bewegungsmodell des Einsatzfahrzeugs berechnet.

$$p(x_t | u_t, x_{t-1})$$

x_t stellt den relativen Positionszustand dar und u_t, x_{t-1} die Zustände, über die der Zustand x_t vorhergesagt werden kann.

Das Einsatzfahrzeug hat das Problem, dass für die Bestimmung der Orientierung keine interne Sensorik vorhanden ist. Für die Positionsbestimmung wird deshalb eine Methode angewandt, die die Orientierung nur über die getätigte Wegmessung $u_t = (d_{r,t}, d_{l,t})$ berechnet. Die Berechnung der Wegmessung erfolgt über die eingebauten Inkrementalgeber. Dazu wird der Reifendurchmesser D durch die Gesamtanzahl N der detektierten Impulse pro Umdrehung U geteilt.

$$\delta d = \frac{D}{N}$$

δd beschreibt den zurückgelegte Weg zwischen zwei Impulsmessungen. Um die Wegmessung im Zeitintervall $[t - 1, t]$ zu bestimmen, wird der Weg zwischen zwei Impulsen δd mit der Gesamtanzahl der Impulse im Zeitintervall $[t - 1, t]$ multipliziert.

$$d_l = \delta d \cdot \sum_i i_{links} \quad d_r = \delta d \cdot \sum_i i_{rechts}$$

Für die Berechnung der Orientierung wird die gefahrene Wegstrecke d_l des linken Rades der Messung des rechten Rades d_r gegenüber gestellt und die Differenz der Messungen betrachtet. Ist die Differenz im Zeitintervall $[t - 1, t]$ positiv, fährt das Fahrzeug laut des Bewegungsmodells eine Linkskurve. Ist der Differenzbetrag negativ fährt das Fahrzeug eine Rechtskurve [Abb. 5.17]. Da der Kreisradius beider Messungen unterschiedlich ist, kann die Orientierungsänderung $\delta\theta$ über den Reifenabstand q und den Differenzbetrag ermittelt werden.

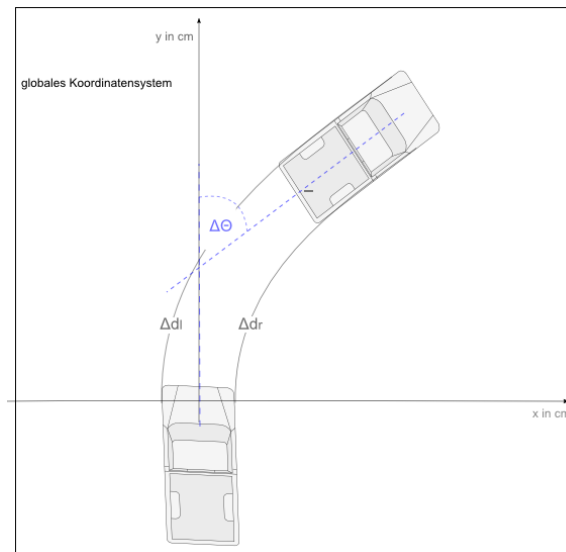


Abbildung 5.17: Odometrie

$$\delta\theta = \frac{d_l - d_r}{q}$$

Die Berechnung ist möglich, da die Wegmessung beider Räder sich bei einer Kurvenfahrt nur durch den Reifenabstand d unterscheidet. Für das rechte Rad gilt demnach:

$$2\pi(r_{rechts}) = 2\pi(r_{links} - q)$$

wo r_{rechts} den Kreisradius für das rechte und r_{links} den Radius für das linke Rad angeben. Die Wegstrecken d_l und d_r , sowie der zurückgelegte Weg d_m im Mittel lassen sich über den Kreisumfang berechnen.

$$d_l - d_r := 2\pi(q)$$

$$d_l := 2\pi(r + q)$$

$$d_r := 2\pi(r)$$

$$d_m := \frac{(d_l + d_r)}{2}$$

Durch die oben beschriebene Erkenntnis werden die Orientierung θ und die Positionskor-
 dinaten (x, y) des neuen Positionszustandes x_t durch die Orientierungsänderung $\delta\theta$, der
 Fahrzeugbewegung u_t , sowie den Positionsdaten des alten Zustandes x_{t-1} bestimmt.

$$x_t = x_{t-1} + \begin{pmatrix} d_{m,t} \sin(\theta_{x_t} + \delta\theta) \\ d_m \cos(\theta_{x_t} + \delta\theta) \\ \delta\theta \end{pmatrix}$$

5.4.2.2 Bewertung der Ungenauigkeit

Wie in der [Abb. 5.6] zu erkennen hat die reinodometriebasierte Positionsbestimmung den
 Nachteil, dass sie mit zunehmender Fahrt ungenauer wird. Die Ungenauigkeit resultiert aus
 den Fehlern in der Wegmessung u_t und dem ungenauen Positionszustand x_{t-1} . Um die
 Positionsschätzung zu verbessern, wird nach der ersten Streckendurchfahrt und einem er-
 folgreichen loop close eine Referenzabweichung ermittelt, die sich aus den Bewegungszu-
 ständen $u_{tr=g,0:t}$ einer Geraden g (topologische Region) berechnet. In $u_{tr=g,0:t}$ sind alle
 Wegmessungen des linken und rechten Rades enthalten, die auf allen Geraden der Fahr-
 bahn gemessen wurden. Messungen anderer topologischer Regionen (Kurven) werden igno-
 riert.

Für einen optimalen Fall gilt für alle Wegmessungen $u_{tr=g,0:t} = (d_{l,0:t}, d_{r,0:t})$:

$$d_{l,0:t} = d_{r,0:t}, \text{ wobei } \{d_{l,0:t}, d_{r,0:t}\} \in u_{tr=g,0:t}$$

Da die Summer aller Wegmessungen des rechten rades auf den Geraden genau mit den
 Wegmessungen des linken rades übereinstimmen muss, gibt die Differenz der Summen den
 Gesamtfehler in der Bewegung an.

$$E(u_{tr=g,0:t}) = \sum_{t=0}^n (d_{r,t} - d_{l,t}), \text{ wenn } TR = \text{Gerade}$$

Um die Positionsschätzung zu optimieren, wird der Gesamtfehler auf die einzelnen Mes-
 sungen des linken und des rechten Rades gleichverteilt. Die Verteilung des Fehlers wird im
 Fehlervektor $E(u_t) = (\epsilon_{d_l}, \epsilon_{d_r})$ festgehalten, der den konstanten Odometriefehler für das
 linke und rechte Rad darstellt.

$$E(u_t) := \left(1 + \frac{E(u_{tr=g,0:t})}{2 \cdot \sum(d_{l,t})}, 1 + \frac{-E(u_{tr=g,0:t})}{2 \cdot \sum(d_{r,t})} \right)^T$$

Die obere Definitionen gilt, da $E(u_{0:t})$ genau der Summer der Einzelabweichungen der Wegmessungen $d_{r,t}$ und $d_{l,t}$ entspricht. Wird der Fehler komplett auf das linke Rad oder das rechte Rad verteilt ergibt sich die Berechnung

$$E(u_{r,t}) = \sum(d_{l,t} * \epsilon_{d_l} - d_{l,t})$$

$$E(u_{l,t}) = \sum(d_{r,t} * \epsilon_{d_r} - d_{r,t})$$

Durch die gewonnene Betrachtung wird der Bewegungszustand, um die Abweichung, erweitert. Beide Streckenmessung werden einander angepasst, sodass sie idinetisch sind. Für den neuen Bewegungsvektor gilt:

$$\tilde{u}_t = \begin{pmatrix} d_r \\ d_l \end{pmatrix} * E(u_t)$$

Da die Abweichung in u_t nur relativ vorhergesagt werden kann, wird die Positionsbestimmung unter Zunahme einer normalverteilten Abweichung durchgeführt. Für jede lokale Messung z_t wird der neue Positionszustand unter Zunahme einer zufälligen Abweichung mehrmals geschätzt. Damit wird ein Scan s_t mit unterschiedlichen Positionen zur Zeit t abgespeichert. Die Menge der Positionszustände gibt die Samplenumenge an. Das Bewegungsmodell ändert sich damit wie folgt:

$$p(s_t^i | x_{t-1}, u_t)$$

Für jede Schätzung des Positionzustades x_t wird eine Samplenumenge $s_t^i, s_t^{i+1}, \dots, s_t^{i+n}$ geschätzt, in der jeder Sample ein Scan mit einer möglichen Position $s_{x,t}^i$ darstellt.

Im Weltmodell entstehen durch den Aufbau der Samplenumenge eine Menge möglicher Positionspfade. Jeder Pfad J_t beschreibt einen möglichen Trajektoriengraphen der Fahrt.

$$J_i = s_t^i, s_{t+1}^i, \dots, s_{t+n}^i$$

Zur Lokalisierung einer Scanposition $s_{x,t}^i$, wird die odometriebasierte Berechnung um eine normalverteilte Abweichung mit der Varianz σ und dem Erwartungswert μ erweitert.

$$\tilde{u}_t = \begin{pmatrix} d_r \\ d_l \end{pmatrix} + \begin{pmatrix} \epsilon_{\alpha d_r} \\ \epsilon_{\alpha d_l} \end{pmatrix}$$

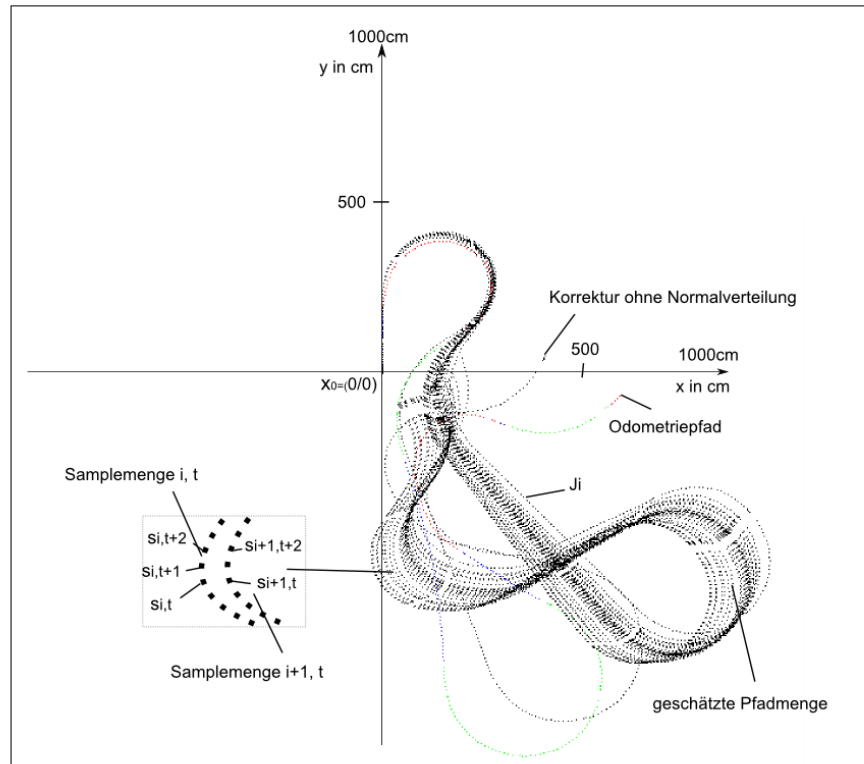


Abbildung 5.18: Geschätzte Pfadmenge unter Zunahme einer normalverteilten Abweichung

Die zufällige Abweichung $\epsilon_{\alpha d_r}$, $\epsilon_{\alpha d_r}$ stellt ein zufälliges Rauschen in der Odometrie dar und wird durch den Vektor

$$E(u_t) := \begin{pmatrix} \epsilon_{\alpha s_r} \\ \epsilon_{\alpha s_l} \end{pmatrix}$$

definiert. Die Abweichungen ϵ_{s_r} und ϵ_{s_l} werden über die Zufallsabweichung z den Erwartungswert μ und die Varianz σ beschrieben. Dabei ist z normalverteilt, wenn die Dichte von z durch:

$$f(z) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(z - \mu)^2}{2\sigma^2}\right)$$

gegeben ist [Schlittgen (2008)]. Für die Berechnung der normalverteilten Abweichung wird die Box-Muller Methode verwendet, die aus zwei Standardzufallszahlen u_1, u_2 die rechteck-

verteilt sind, durch Transformation eine normalverteilte Zufallszahl z generiert. Für die normalverteilte Zufallszahlen gilt die Definition:

$$z := r \cdot \cos(\varphi)$$

Die Standardzufallszahlen werden in Polarkoordinaten r und φ über die Inversionsmethode transformiert.

$$r = \sqrt{-2 \ln u_1}$$

$$\varphi = 2\pi u_2$$

Die erzeugte Zufallszahl z ist standardnormalverteilt mit dem Erwartungswert $\mu = 0$ und der Varianz $\sigma = 1$. Für die Normalverteilung wird die Abweichung um die Varianz und den Erwartungswert angepasst [[Schlittgen \(2008\)](#)].

$$\epsilon_s = \sigma z_i + \mu$$

Wobei bei der Berechnung von einem Erwartungswert $\mu = 0$ ausgegangen wird.

Die mögliche Abweichung wird pro Pfad zu Beginn der zweiten Fahrt abgespeichert und bei der Berechnung des Positionsvektors berücksichtigt. Die Berechnung der Fahrzeugorientierung ändert sich wie folgt:

$$\tilde{\delta\theta} = \frac{(d_l + \epsilon_{d_l}) - (d_r + \epsilon_{d_r})}{d}$$

In der [Abb. 5.18] ist die komplette Predictionphase dargestellt. Bei der Abarbeitung werden durch die normalverteilte Abweichung n Pfade erzeugt. Jeder Pfad in der Abbildung stellt einen geschätzten Trajektoriengraphen dar, der über die Menge der Scanzustände (Samplemenge) definiert ist. Wie in der Abbildung zu erkennen, verbessert sich bei manchen Pfaden die Schätzung durch Zunahme der normalverteilten Abweichung. Die optimale Schätzung wird in der Update-Phase bestimmt.

5.4.3 Positionskorrektur und Verifikation (Update Phase)

Die Updatephase ist die letzte Abarbeitungsphase der Kartierung. Während in den zuvor beschriebenen Abarbeitungsphasen, die Karte vorbereitet, die lokale Umgebung abgespeichert und die Fahrzeugposition geschätzt wird, ermittelt die Updatephase die bestmögliche

Schätzung und aktiviert aus der Gesamtpfadmenge den Pfad, der die gefahrene Fahrzeugtrajektorie am besten projiziert.

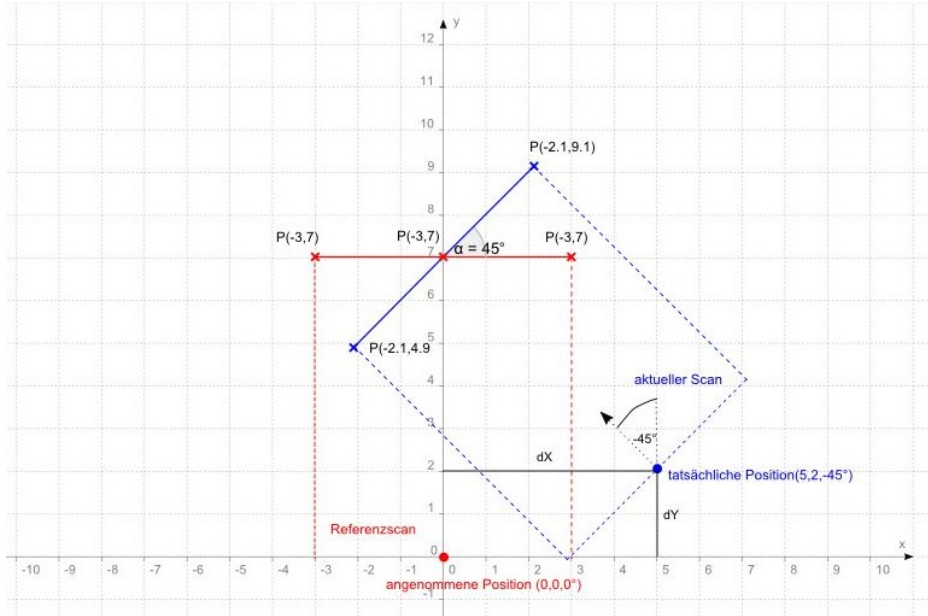


Abbildung 5.19: Scanmatching zwischen dem Referenzscan r_{ref} und dem aktuellen Scan s_t

Um die bestmögliche Schätzung zu ermitteln, wird die Start- und Stopplinie im Anfangsbereich verwendet. Dazu wird die Startlinie in einem lokalen Referenzscan in Form von Punktkoordinaten abgespeichert und der Referenzscan im Anfangsbereich abgelegt. Der Vorteil des Referenzscans besteht darin, dass die Scanposition zu Beginn der Fahrt genau ist. Korreliert eine neu aufgenommene Messung z_t mit dem Referenzscan s_{ref} , kann die aktuelle Fahrzeugposition x_t über die Position des Referenzscans $s_{x,ref}$ berechnet werden.

Die Positionsbestimmung erfolgt über eine paarweise Überdeckung zwischen dem aktuell aufgenommenen Scan (mit der Startlinie) und dem Referenzscan [Abb. 5.19]. In der Literatur spricht man von einem sogenannten Scanmatch [Lu und Milios (1997)]. Ein Match wird beschrieben durch die Funktion $match$, die zwei Scans miteinander vergleicht und einen Fehlervektor ausgibt, der den Fehler in der Verschiebung und Verdrehung enthält.

$$(\delta x, \delta y, \delta \theta)^T = match(s_t, s_{ref})$$

Die Funktion implementiert den in [Lu und Milios (1997)] entwickelten "Iterative Dual Correspondence [IDC]" Algorithmus. Der Algorithmus gleicht die Punkte der Scans paarweise ab und bestimmt den Fehlervektor $E(\delta T, \delta \theta)$, wobei T die Translationsparameter $T = (\delta x, \delta y)^T$ enthält.

Die Ermittlung der Verdrehung und Verschiebung erfolgt über die Bildung von Korrespondenzpaaren der beiden Scans. Dabei werden die Punkte des aktuellen Scans $p_i \in s_t$ den Punkten des Referenzscans $p'_i \in s_{ref}$ zugewiesen. Für die Zuordnung verwenden [Lu und Miliotis (1997)] zwei unterschiedliche Methoden, die Iterative Closeded Point Methode (ICP) und die Find Matching Range Point (FMRP) Methode.

Der Unterschied der beiden Algorithmen liegt in der Zuordnung der Punkte. Während der ICP Algorithmus nach einem Punkt im Referenzscan sucht, der einem Punkt im aktuellen Scan am nächste ist, ordnet der FMRP Algorithmus einem aktuellen Punkt, einen Punkt im Referenzpunkt zu, der möglichst die gleiche Entfernung zur Aufnahmeposition besitzt. Laut [Lu und Miliotis (1997)] eignen sich die Korrespondenzpaare des ICP Algorithmus besser für die bestimmung der Verschiebung und die Paare des FMRP Algorithmus besser für die Bestimmung der Verdrehung.

5.4.3.1 ICP Algorithmus

Der ICP Algorithmus ist in [Alg. 1] dargestellt. Zur Bestimmung der Korrespondenzpaare $K_i = (p_i, p'_i)^T$ aus dem aktuellem Scan und dem Referenzscan wird zwischen allen Punkten des Referenzscans interpoliert, sodass alle Scanpunkte durch kurze Liniensegmente verbunden sind. Für jeden Scanpunkt des aktuellen Scans wird anschließend der Abstand zum Liniensegment zwischen zwei aufeinanderfolgenden Referenzpunkten berechnet. Das Liniensegment mit dem kleinsten Abstand zum aktuellen Punkt beinhaltet den Korrespondenzpartner \hat{p}_i für p_i . Damit kann ein Korrespondenzpartner \hat{p}_i auf dem Liniensegment zwischen zwei Referenzpunkten p'_i und p'_{i+1} liegen und muss nicht direkt einem Referenzpunkt entsprechen [Abb. 5.20].

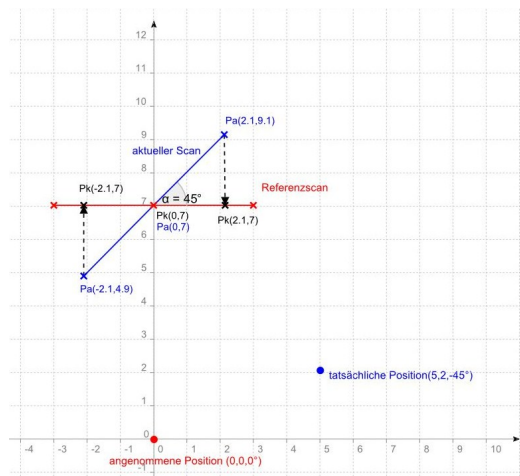


Abbildung 5.20: Korrespondenzpartner auf einem Liniensegment

Der Korrespondenzpunkt lässt sich über die Geradengleichung $y = mx + b$ der Referenzpunkte p'_i und p'_{i+1} berechnen. Aus der Geradengleichung wird die Steigung m_{G1} und der y-Achsenabschnitt b_{G1} der Gerade bestimmt.

$$m_{G1} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

$$b_{G1} = y_i - m_{G1} * x_i$$

Um den Punkt \hat{p}_i zu ermitteln wird nach einer Geraden zwischen dem aktuellen Punkten p_i und dem Korrespondenzpartner \hat{p}_i gesucht. Da die gesuchte Gerade y_2 senkrecht zu der ersten Gerade steht [Abb. 4.17], gilt für die Steigung m_{G2} der zweiten Gerade:

$$m_{G2} = \frac{-1}{m_{G1}}$$

Der y-Achsenabschnitt wird mittels

$$b_{G2} = x_a * m_{G2} + y_a$$

berechnet. Durch das Gleichsetzen beider Geraden können die Punktkoordinaten des Referenzpunktes ermittelt werden.

Algorithm 1 ICP Algorithmus

Require: Referenzscanpunkte: P_i , aktueller Scanpunkt P_a

Ensure: Korrespondenzpunkt P'

$d_{New} \leftarrow 0$

$d_{Old} \leftarrow -1$

for $i = 1$ to $n - 1$ **do**

$d_{New} \leftarrow getDistanceToReferenceLine(P_i, P_{i+1})$

if $|d_{New}| < |d_{Old}|$ or $d_{Old} = -1$ **then**

$d_{Old} \leftarrow d_{New}$

$P_1 \leftarrow P_i$

$P_2 \leftarrow P_{i+1}$

end if

$P' \leftarrow getPointOfLine(P_1, P_2)$

end for

return P'

5.4.3.2 FMRP Algorithmus

Beim FMRP Algorithmus erfolgt die Bestimmung der Korrespondenzpaare über die gleiche Entfernung zur Aufnahmeposition [Abb. 5.21]. Anstelle der Aufnahmeposition wird in dieser Arbeit der Koordinatenursprung des Scans, als Referenz genommen. Bei der Abarbeitung wird für jeden Punkt p_i des neuen Scans s_t und für jeden Punkt p'_i des Referenzscans die Distanz q zum Koordinatenursprung berechnet. Der Referenzpunkt, der die gleiche Distanz wie der aktuelle Punkt aufweist, bestimmt den Korrespondenzpartner. In [Lu und Milios (1997)] wird das Verfahren noch erweitert, indem auch Bereichen zwischen zwei Referenzpunkten analysiert werden. Da sich die Bearbeitungszeit des Algorithmus dadurch erhöht, werden für das Korrekturverfahren nur die gegebenen Punkte im Referenzscan analysiert.

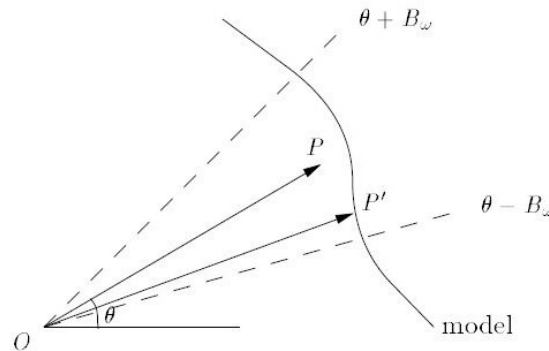


Abbildung 5.21: Referenzpunkt mit der gleichen Distanz zum Koordinatenursprung⁹

5.4.3.3 Abarbeitung der Korrektur und Pfadauswahl

Wurden die Korrespondenzpaare der beiden Scans gebildet, erfolgt die Bestimmung der Verschiebung und Verdrehung. Der Fehlervektor wird über die Methode der kleinsten Abstandsquadrate berechnet. Dazu wird nach einem Vektor gesucht mit dessen Transformationsparameter $(\Delta T, \Delta\theta)$ ein Scan an den anderen angepasst wird, sodass die Summe der Abstandsquadrate möglichst klein wird. Ist die Summe 0, ergibt sich eine vollständige Überdeckung beider Scans.

$$E_{fit}(\Delta\alpha, \Delta t) = \sum_{i=1}^n |R(\Delta\alpha)P_i + \Delta T - P'_i|^2$$

$$= \sum_{i=1}^n \left((x_i \cos(\Delta\alpha) - y_i \sin(\Delta\alpha) + \Delta T_x - x'_i)^2 + (x_i \sin(\Delta\alpha) + y_i \cos(\Delta\alpha) + \Delta T_y - y'_i)^2 \right)$$

⁹Lu und Milios (1997)

Durch Umstellung erhält man die Rotationsmatrix:

$$R(\Delta\alpha) = \begin{pmatrix} \cos(\Delta\alpha) & -\sin \Delta\alpha \\ \sin(\Delta\alpha) & \cos \Delta\alpha \end{pmatrix}$$

Die obere Formel lässt sich nach [Lu und Milios (1997)] minimieren, sodass sich der Rotationswinkel und die Verschiebung der Scans über

$$\Delta\alpha = \arctan \frac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}}$$

$$\Delta T = \begin{pmatrix} \tilde{x}' \\ \tilde{y}' \end{pmatrix} - \begin{pmatrix} \cos(\Delta\alpha) & -\sin(\Delta\alpha) \\ \sin(\Delta\alpha) & \cos(\Delta\alpha) \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$$

berechnen lassen. $S_{xy'}$, $S_{yx'}$, $S_{xx'}$, $S_{yy'}$ stellen die Summen der Kovarianzen der Referenzpunktkoordinaten und der Koordinaten des neuen Scans dar. Mit den ermittelten Parametern wird anschließend die absolute Endposition des Fahrzeuges $x_{t,new}$ berechnet, indem die im Referenzscan gespeicherte Position $s_{x,ref}$ um die Translationsparameter $\Delta T = (\delta x, \delta y)^T$ verschoben und um den ermittelten Rotationswinkel $\Delta\theta$ verdreht wird.

$$x_{t,new} = \begin{pmatrix} x_{ref} \\ y_{ref} \\ \theta \end{pmatrix} + \begin{pmatrix} \Delta T_x \\ \Delta T_y \\ \Delta\alpha \end{pmatrix}$$

Die ermittelte Position gibt die Referenzposition für die Pfadselektierung an. Bei der Selektierung eines Pfades wird die euklidische Distanz von den Endpositionen des Pfades zur Referenzposition ermittelt und ein Vergleich aufgestellt. Der Pfad mit der kleinsten euklidischen Distanz zur Referenzposition (dessen Position am nächsten an der Referenzposition ist) wird selektiert. Andere Pfade werden entfernt und vom System nicht mehr betrachtet.

Da die Umgebungsstruktur aus mehreren topologischen Bereichen besteht, wird der aktive Pfad pro Bereich mit der Startposition des Bereichs und der Endposition des vorherigen Bereichs selektiert. Am Ende der Updatephase bleibt nur ein Pfad im System übrig, der die am besten geschätzte Trajektorie der Fahrt darstellt [Abb. 5.22].

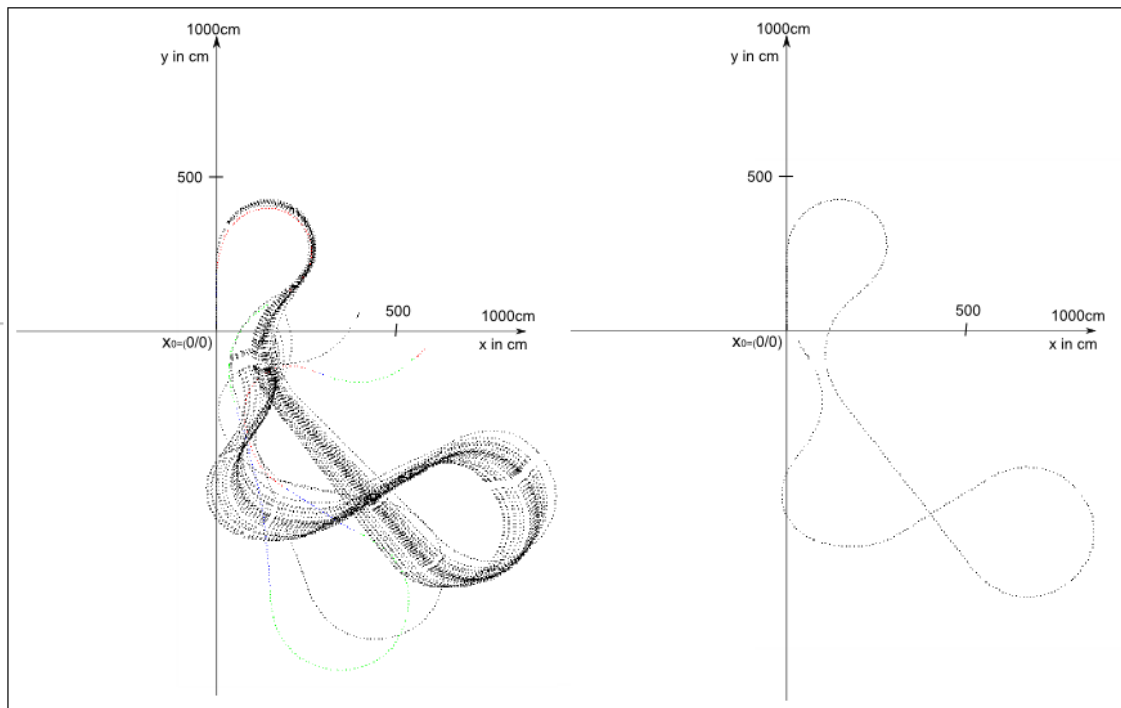


Abbildung 5.22: Auswahl des Schätzpfades

6 Implementierung

In diesem Kapitel wird die entwickelte Softwarearchitektur der Weltmodellierung vorgestellt. Der Aufbau der Kartierung wird anhand von UML-Klassendiagrammen erläutert und das Zusammenspiel, der unterschiedlichen Module erklärt. Auf die Implementierung der geometrischen Kartierung und der Kartenfusionierung wird nicht eingegangen, da diese in der Arbeit [Rull (2008)] ausführlich erläutert ist.

6.1 Aufbau der Softwarearchitektur

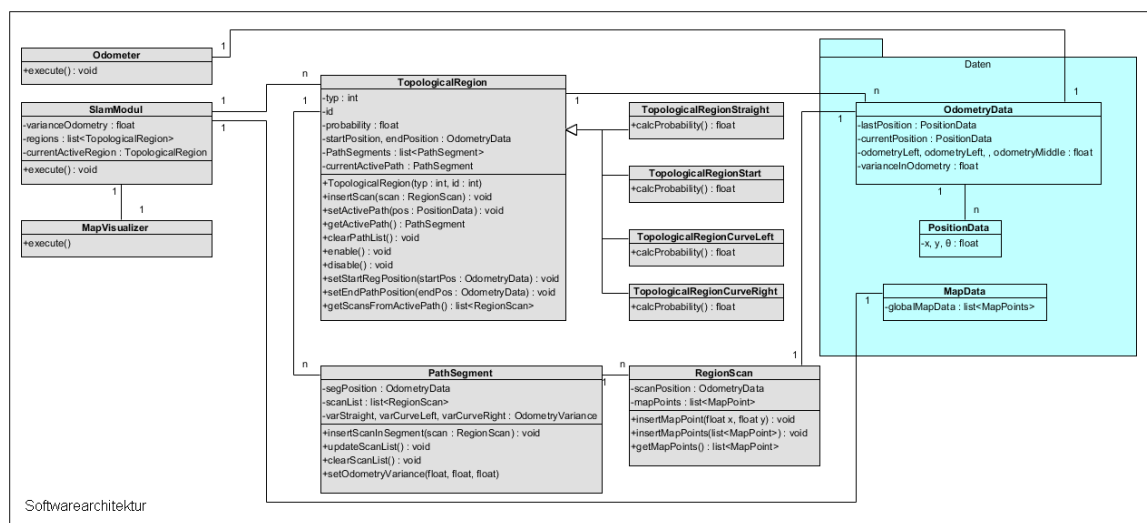


Abbildung 6.1: Softwarearchitektur

Wie schon im vorherigen Kapitel erwähnt, basiert die Softwarearchitektur des Einsazufahrzeuges auf der Subsumption Architektur. Um die Kartierung in die Software zu integrieren, wird ein Plugin-Modul implementiert, das die Verfahren der Weltmodellierung enthält. Während der Fahrt wird es dem dem Kernmodel (FAUSTcore) aktiviert, sodass die Umgebung kartiert wird. Die Softwarearchitektur des SLAM Verfahrens setzt sich auch drei Bausteinen zusammen.

Der erste Baustein ist der Odometer, der das einfache Bewegungsmodell implementiert, wo die Fahrzeugposition anhand der Odometriedaten berechnet wird. Die relative Position wird in einem Datenobjekt gespeichert und im Datenspeicher abgelegt, sodass alle Pluginmodule auf die Position zugreifen können. Das sogenannte OdometryData-Objekt beinhaltet alle Informationen zwischen zwei Positionszuständen x_t und x_{t+1} , wie Wegmessung des linken Rades, des rechten Rades, die Orientierungsänderung, sowie die Gesamtmessung des zurückgelegten Weges.

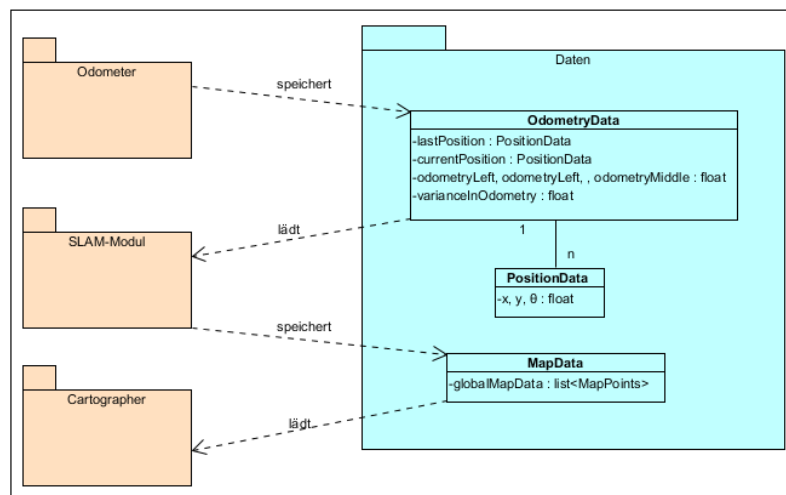


Abbildung 6.2: Globaler Aufbau der Kartierungssoftware

Die Verfahren der SLAM-Problematik werden von dem zweiten Baustein bearbeitet, das sich aus dem SLAM-Modul, den topologischen Regionen, den Pfaden und den Scans zusammensetzt. Das SLAM-Modul stellt die zentrale Steuereinheit dar. Im SLAM Modul wird hauptsächlich die Beobachtung durchgeführt und die Topologien abgefragt in denen sich das Fahrzeug befindet. Die Bewertung der Umgebung in den TopologicalRegion-Objekten implementiert. Da jeder topologische Bereich eine eigenständige Bewertung implementiert, werden die Objekte der unterschiedlichen Bereiche aus dem TopologicalRegion-Objekt abgeleitet. Die Ableitung hat den Vorteil, dass verschiedene Bereiche unterschiedlich behandelt werden können.

Damit eine globale Umgebungszuordnung der Bereiche erfolgen kann, wird jedes TopologicalRegion-Objekt mit einer Start- und Endposition (als OdometryData-Objekt) abgespeichert. Das OdometryData-Objekt hat den Vorteil, dass (im Vergleich zum PositionData-Objekt) die gesamten Odometriedaten zwischen der Anfangs- und Endposition hinterlegt sind. Zudem enthält jedes TopologicalRegion-Objekt eine Menge möglicher Fahrpfade (PathSegments). Die PathSegment-Objekte stellen eine mögliche Trajektorie des Fahrzeuges dar. Das PathSegment-Objekt implementiert die vorhersage eines neuen

Positionszustandes unter Zunahme der Varianzen in der Odometrie. Da unterschiedliche Topologien unterschiedliche Varianzen der Odometrie aufweisen, werden Abweichungen für gerade Strecken, Linkskurven und Rechtskurven abgelegt.

Die Trajektorie der Fahrt wird über die Menge gespeicherter RegionScan-Objekte beschrieben, die in einer Liste abgelegt sind. Jedes RegionScan-Objekten speichert eine Messung z_t^i mit einem geschätzten Positionszustand s_{x_t} zur Zeit t ab.

Der Austausch der gespeicherten Positionsdaten und Kartendaten erfolgt über das globale Kommunikationsdatenobjekt, MapData. Das Objekt wird durch das SLAM Modul mit den korrigierten Karten aus dem aktiven Pfad initialisiert, sodass die Daten der geometrischen Kartierung weitergereicht werden können. Der globale Aufbau ist in [Abb. 6.2] zu sehen.

Die Kartendaten können auch in das XML-Format exportiert werden, sodass die Karte zu einem beliebigen Zeitpunkt geladen werden kann. Für die Darstellung der Karte ist der Map-Visualizer verantwortlich. Das Objekt fragt das SLAM Modul nach den aktuellen Daten ab und stellt diese visuell dar.

6.2 Ablauf der Weltmodellierung

Der Ablauf der Kartierung startet immer mit dem Odometer, der den relativen Positionszustand x_t berechnet und im OdometryData-Objekt abspeichert. Wird die Strecke zum ersten Mal befahren, bewertet das Slam-Modul die aktuelle Umgebung anhand der Eigenschaften der vier unterschiedlichen Bereiche. Der Bereich der mit der aktuellen Bewegung und Beobachtung am besten korreliert, definiert die aktuelle Topologie.

Hat sich die Topologie im Vergleich zur letzten Bewertung nicht verändert, wird die neue Messung in der zuvor erstellten Topologie gespeichert. Hat sich die Topologie verändert, wird ein neues TopologicalRegion-Objekt angelegt und die Messung in einem neuen Objekt abgelegt. Sind im System schon alle Topologien vorhanden, wird die aktuelle Bewegung und Beobachtung nur mit den Eigenschaften der aktuell aktiven und der nächstmöglichen Topologie bewertet.

Die neu aufgenommenen Daten werden über ein RegionScan-Objekt an die aktive Topologie gegeben. Enthält die Topologie mehrere Pfadsegmente, wird der Scan an jedes Segment weiter gereicht. In den Segmenten wird die Position des Scan $s_{x,t}$ unter Zunahme des normalverteilten Fehlers in der Wegmessung neu berechnet und aktualisiert.

[Abb. 6.3] veranschaulicht einen vereinfachten Ablauf einer Scanaufnahme. Die Abbildung stellt die erste Streckendurchfahrt dar, wo zur Zeit $t + n$ der Anfangsbereich erkannt und der Fahrpfad geschlossen wird (loop close).

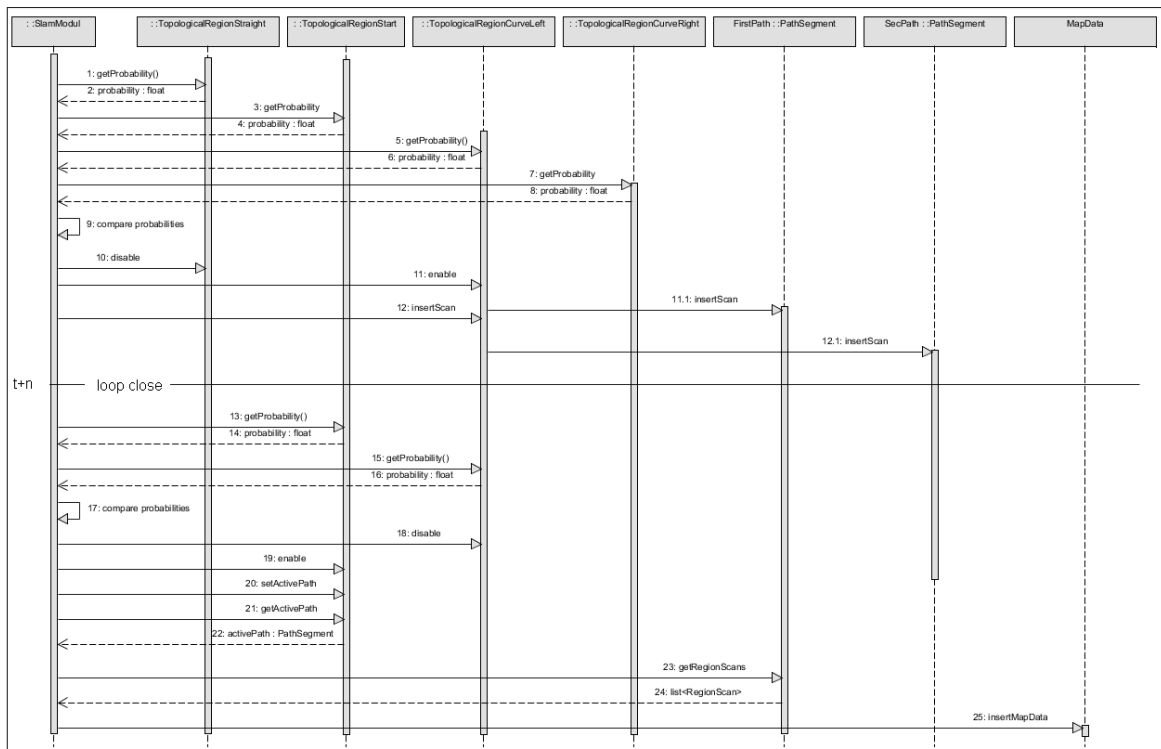


Abbildung 6.3: Ablauf der Scanaufnahme

7 Auswertung der Verfahren

In diesem Kapitel werden die Ergebnisse des zuvor beschriebenen Verfahrens der Weltmodellierung vorgestellt. Dabei soll der reale Positionspfad der Strecke [Abb. 5.1] optimal geschätzt und der fehlerhafte Odometripfad [Abb. 5.6] korrigiert werden. Die Visualisierung der Ergebnisse erfolgt über den Positionspfad. Bei den Tests wurde das Fahrzeug direkt vor der Startlinie positioniert, sodass der Anfangsbereich bekannt ist. Der Positionszustand der ersten Aufnahme beträgt $s(x, 1) = (0, 0, 360^\circ)$. Die reale Endposition wurde nach einer Streckendurchfahrt unter Zunahme der Startposition abgemessen und beträgt $x_{end} = (0, -40, 320^\circ)$. Die Endposition x_{end} gibt damit die Sollposition an.

Das Verfahren der Weltmodellierung wird unter Zunahme mehrere Aspekte betrachtet, dabei soll der Fehler, der durch die Odometrie verursacht wird ausgeglichen werden. Wie in der [Abb. 5.6] zu erkennen weicht die berechnete Endposition $s(x)^n = ()$, die durch das Bewegungsmodell des Fahrzeug berechnet wurde um den Vektor $E(\Delta T_{x,y}, \delta\theta) = (-571, 11, -89^\circ)$ von der Soll-Position ab.

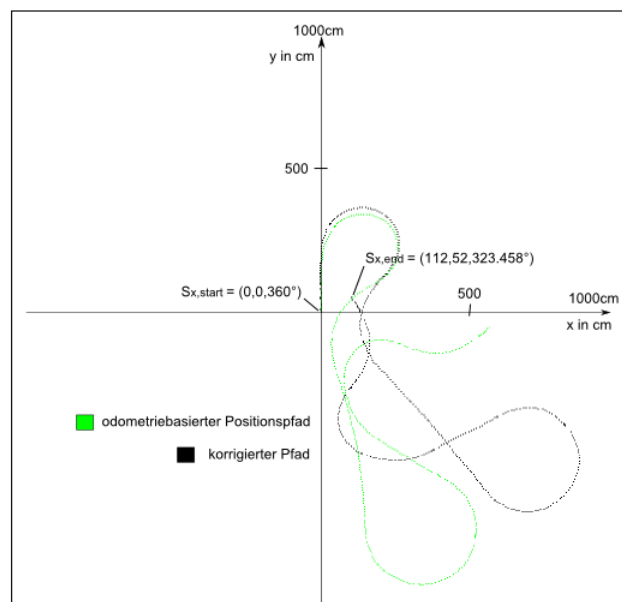


Abbildung 7.1: Schätzung des Weltmodells über die Gleichverteilung des Odometriefehlers

7.1 Schätzung 1

Bei der ersten Schätzung wird die Standardabweichung in den Wegmessungen der Geraden hinzugezogen. Die Abweichung beträgt pro Zeitintervall $[0ms, 50ms]$ für das linke Rad $E(d_l) = -0,339\%$ und das rechte Rad $E(d_r) = +0,339\%$. Die Schätzung ist in [Abb. 7.1] zu sehen.

Wie aus der Abbildung zu entnehmen ist, verbessert sich die Schätzung des Positionspfades. So sind im Vergleich zum odometriebasierten Positionspfad Geraden, als Geraden zu erkennen. Die Endposition verbessert sich auf $s_{x,end} = (112, 52, 323.46^\circ)$.

7.2 Schätzung 2

Für die zweite Schätzung wird die normalverteilte Zufallsabweichung hinzugezogen. Die Varianz der Abweichung beträgt 0,3 des Erwartungswerts $-0,339$ bzw. $+0,339$. Die Schätzung wird mit fünf Pfaden durchgeführt. Durch Zunahme der normalverteilten Abweichung und der fünf unterschiedlichen Schätzungen verbessert sich die Endposition auf $s_{x,end} = (56, 0, 322.26^\circ)$. Die Abweichung zur Sollposition beträgt nur noch $E(\Delta T_{x,y}, \delta\theta) = (-56, -40, 2.26^\circ)$.

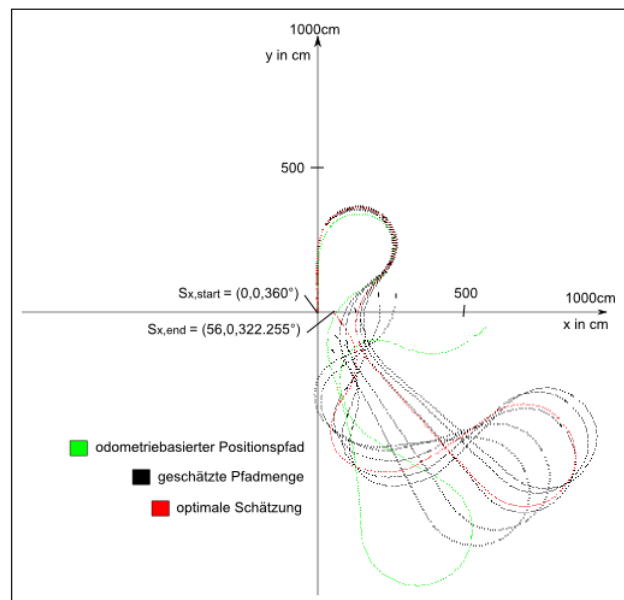


Abbildung 7.2: Schätzung des Weltmodells mit fünf Schätzpfaden

7.3 Schätzung 3

Bei der dritten Schätzung wird die Menge der Pfade auf zwanzig erhöht und die normalverteilte Abweichung mit der Varianz 0,001 betrachtet. Die Zunahme der neuen Pfade erhöht die Wahrscheinlichkeit den optimalen Pfad zu finden. Die Abweichung zur Sollposition beträgt nur noch $E(\Delta T_{x,y}, \delta\theta) = (-40, -8, 0.652^\circ)$.

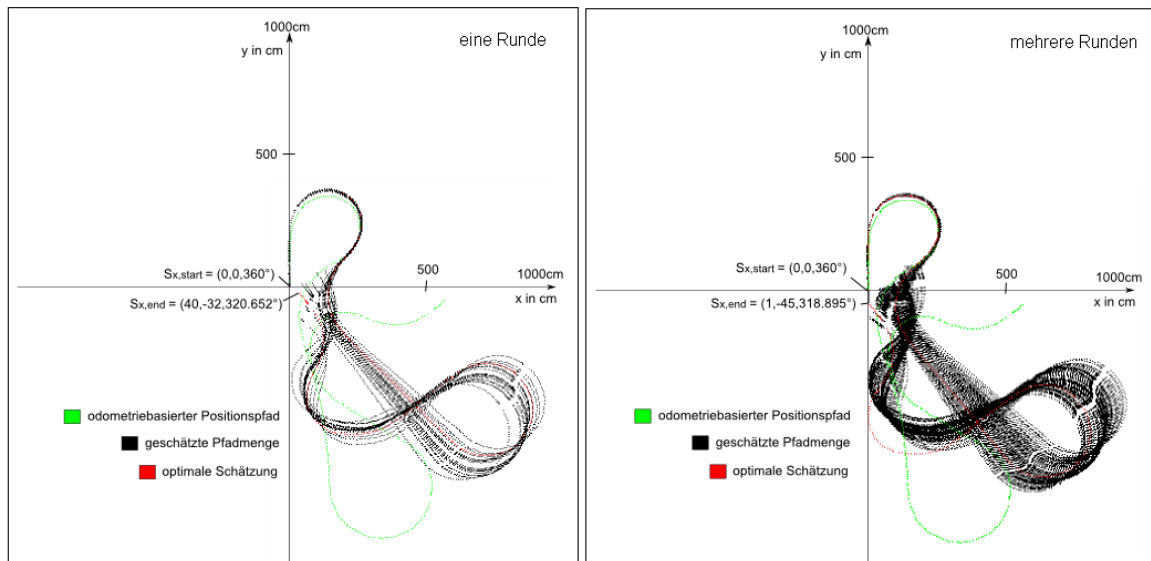


Abbildung 7.3: Schätzung des Weltmodells mit zwanzig Schätzpfaden. Links die Schätzung mit einer Runde und rechts die Schätzung über mehrere Runden

Lässt man nun die Schätzung über mehrere Runden durchführen, wird diese weiter optimiert. Eine Schätzung über fünf Runden mit zwanzig Positionspfaden je Runde ist in [Abb. 7.3] zu sehen. Die letzte Schätzung gibt auch die optimale Positionsschätzung von $S_{x,end} = (-1, 5, 1.11^\circ)$ an

8 Resümee und Ausblick

Ein korrektes Weltmodell ist die Grundlage für die globale Bahnplanung. Hat das Fahrzeug die Möglichkeit auf eine globale Sicht der Umgebung, ist es für das Fahrzeug einfacher Entscheidungen zu treffen. Zudem hat das globale Weltmodell den Vorteil, dass das Fahrzeug die eigene globale Position besser abschätzen kann, indem die Kartendaten mit der aktuellen Beobachtung abgeglichen werden. Hat das Fahrzeug keine Umgebungskarte, muss diese während der Fahrt generiert werden.

Für die Modellierung der Karte wird eine korrekte Position benötigt, was das wesentlichen Problem der Weltmodellierung darstellt. Ohne ein korrektes Weltmodell ist es schwierig eine genau Position zu berechnen. Wiederum ist es ohne eine genaue Position schwierig ein korrektes Weltmodell zu erzeugen. Die Problematik wird als SLAM bezeichnet. Bei dem SLAM wird zwischen dem online und dem offline SLAM Verfahren unterschieden. Während beim online SLAM Problem nur die aktuelle Position und das Weltmodell berechnet werden muss, wird beim offline SLAM Problem der gesamte Positionspfad mit dem Weltmodell berechnet.

Für die Weltmodellierung dieser Arbeit, wurde eine offline Verfahren entwickelt, das das Weltmodell über mehrere mögliche Positionspfade schätzt. Die Anzahl der Pfade ist variable einstellbar. Die Schätzung erfolgt über eine normalverteilte Abweichung in der Fahrzeugbewegung.

Damit eine Korrektur überhaupt möglich ist, wird die Fahrbahn in Topologien eingeteilt, wo jede Topologie einen Bereich der Fahrbahn darstellt. Um eine Korrektur durchzuführen wird die Startlinie im Anfangsbereich verwendet. Die Startlinie definiert das Merkmal für einen geschlossenen Kreis, was als loop close bezeichnet wird. Wird die Startregion wiedererkannt, erfolgt eine Auswahl des optimalen Pfades aus der Menge der geschätzten Positionspfade. Der Positionspfad, der die bestmögliche Schätzung der Fahrbahn darstellt wird selektiert.

Wie in den Beispielen gezeigt funktioniert das entwickelte Verfahren in der Praxis und ermöglicht eine offline Positionsbestimmung und eine offline Weltmodellierung der Fahrbahn. Das Verfahren ist in der Lage, die Fahrbahn in mehrere Umgebungsbereiche zu gliedern, die Bereiche wiederzuerkennen und einen globalen Positionspfad zu schätzen anhand dessen

die lokalen Beobachtungen in eine globale Sicht transformiert werden können. Das Ergebnis des Verfahrens ist zufriedenstellend, denn mittels der modellierten Karte, lassen sich anschließend Kameraausfälle auf dem Einsatzfahrzeug kompensieren [Nowacki (2010)].

Die Schwäche des Verfahrens ist, dass es nur auf Schätzung anhand der Fahrzeugbewegung beruht. Die Bestimmung der Varianzen in den Wegmessungen, wird nur einmal pro Runde durchgeführt, da nur die Startlinie als markante Stelle genutzt wird. Für die Verbesserung der Schätzung können zusätzliche Topologien (wie Kreuzung) definiert werden, in denen die Varianzen der Wegmessungen genau wie an der Startlinie Neuberechnet werden. Zudem verbessert sich die Schätzung, wenn das Fahrzeug die Fahrbahn öfters durchfährt, sodass die Schätzposition der Sollposition über mehrere Schätzzyklen angenähert wird.

Literaturverzeichnis

- [Arnaud Doucet 2000] ARNAUD DOUCET, Kevin Murphy Stuart R.: Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. (2000)
- [Borenstein 1996] BORENSTEIN, Feig L.: Gyrodometry: a new method for combining data from gyros and odometry in mobile robots. In: *IEEE International Conference on Robotics and Automation* (1996), S. 423–428
- [Brooks 1986] BROOKS, R.: A Robust Layered Control System for a Mobile Robot. In: *IEEE Journal of Robotics and Automation* (1986)
- [Daniel Pizarro und Losad 2010] DANIEL PIZARRO, Enrique Santiso Marta Marron David Jimenez Santiago C. ; LOSAD, Cristina: Localization of Mobile Robots Using Odometry and an External Vision Sensor. In: *Sensors 2010* 10 (2010), S. 3655 – 3680
- [Diosi und L.Kleeman 2005] DIOSI, A. ; L.KLEEMAN: Laser scan matching in polar coordinates with application to SLAM. In: *IEEE International Conference on Intelligent Robots and Systems* (2005), Aug, S. 3317–3322
- [Durrant-Whyte und Bailey 2006] DURRANT-WHYTE, Hugh ; BAILEY, Tim: Simultaneous Localization and Mapping: Part I. In: *IEEE Robotics and Automation Magazine* (2006), S. 99–108
- [Ebert 2008] EBERT, Michael: *Aktives Mapping und Positionsbestimmung eines autonomen Modellfahrzeugs*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008
- [Erfes 1987] ERFES, Alberto: Sonar-Based Real-World Mapping and Navigation. In: *IEEE Journal of Robotics and Automation* 3 (1987), S. 249–265
- [Gutmann 1996] GUTMANN, Jens-Steffan: *Vergleich von Algorithmen zur Selbstlokalisierung eines mobilen Roboters*, Universität Ulm, Diplomarbeit, 1996
- [Hauke Strasdat 2007] HAUKE STRASDAT, Maren Bennewitz Wolfram B.: Visual Bearing-Only Simultaneous Localization and Mapping with Improved Feature Matching. In: *Autonome Mobile Systeme 2007, 20. Fachgespräch, Kaiserslautern* (2007), S. 15–21

- [Herbert Bay 2008] HERBERT BAY, Tinne Tuytelaars Luc Van G.: SURF: Speeded Up Robust Features. In: *Computer Vision and Image Understanding (CVIU)* 110 (2008), S. 346–359
- [Jenning 2008] JENNING, Eike: *Systemidentifikation eines autonomen Fahrzeugs mit einer robusten, kamerabasierten Fahrspurerkennung in Echtzeit*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2008
- [J.Z. Sasiadek 2008] J.Z. SASIADEK, D. N.: Navigation of an autonomous mobile robot using EKF-SLAM and FastSLAM. In: *Mediterranean Conference on Control and Automation* 16 (2008), Juni, S. 517–522
- [Knieriemen 1991] KNIERIEMEN, Thomas: *Autonome Mobile Roboter*. BI-Wiss.-Verlag, 1991. – ISBN 3411150319
- [Kuipers 2000] KUIPERS, Benjamin: The spatial semantic hierarchy. In: *Artificial Intelligence* 119 (2000), Februar, S. 191–233
- [Lu und Milios 1997] LU, Feng ; MILIOS, Evangelos: Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In: *Journal of Intelligent and Robotic Systems* 18 (1997), S. 249–275
- [M. Montemerlo 2002] M. MONTEMERLO, D. Koller B. W.: FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: *Proceedings of the AAAI National Conference on Artificial Intelligence (2002)*, S. 593–598
- [M. Montemerlo 2003a] M. MONTEMERLO, D. Koller B. W.: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: *Proceeding of International Joint Conferences on Artificial Intelligence (2003)*, S. 1151–1156
- [M. Montemerlo 2003b] M. MONTEMERLO, S. T.: Simultaneous localization and mapping with unknown data association using FastSLAM. In: *IEEE International Conference on Robotics and Automation (2003)*, S. 1985–1991
- [Michael Bosse 2003] MICHAEL BOSSE, John Leonard Seth T.: An atlas framework for scalable mapping. (2003), S. 1899–1906
- [Michael Bosse 2004] MICHAEL BOSSE, John Leonard Seth T.: Simultaneous localization and map building in large-scale cyclic environments using the Atlas Framework. 23 (2004), S. 1113–1139
- [Michael Bosse 2008] MICHAEL BOSSE, John Leonard Seth T.: Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM. 27 (2008), S. 667–691

- [ML. Benmessaoud 2008] ML. BENMESSAOUD, K. Nemra AK. S.: Single-Camera EKF-vSLAM. In: *International Journal of Intelligent Systems and Technologies* 3 (2008)
- [N. M. Kwok 2008] N. M. KWOK, G. D.: Bearing-only SLAM in Indoor Environments Using a Modified Particle Filter. (2008)
- [Nowacki 2010] NOWACKI, Filip: *Überwachung und Führung autonomer Fahrzeuge basierend auf einer geometrischen Umgebungskarte*, Hochschule für Angewandte Wissenschaften Hamburg, Masterarbeit, 2010
- [Rull 2008] RULL, Andrej: *Sensorbasierte Umgebungskartierung mit lokaler Positionskorrektur für autonome Fahrzeuge*, Hochschule für Angewandte Wissenschaften Hamburg, Bachelorarbeit, 2008
- [Schlittgen 2008] SCHLITTFEN, Rainer: *Einführung in die Statistik*. Oldenbourg Verlag München, 2008. – ISBN 9783486587746
- [Sebastian Thrun 2006] SEBASTIAN THRUN, Dieter F.: *Probabilistic Robotics*. The MIT Press, 2006. – ISBN 9780262201629
- [Sebastian Thrun 2008] SEBASTIAN THRUN, John J. L.: Simultaneous Localization and Mapping. In: *Handbook for Robotics* (2008), S. 871–889
- [Siegfried Hochdorfern 2007] SIEGFRIED HOCHDORFERN, Christian S.: Bearing-Only SLAM with an Omnicam Robust Selection of SIFT Features for Service Robots. In: *Autonome Mobile Systeme 2007, 20. Fachgespräch, Kaiserslautern* (2007), S. 8–14
- [T. Lemaire 2005] T. LEMAIRE, J. S.: A practical 3D bearing-only SLAM algorithm. In: *International Conference on Intelligent Robots and Systems* (2005), S. 2449–2454
- [Weber 2003] WEBER, Joachim: *Globale Selbstlokalisierung für mobile Service Roboter*. Universität Kaiserslautern, 2003. – ISBN 9783925178917

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 21. Dezember 2010

Ort, Datum

Unterschrift