



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung: Anwendung 1

Torben Becker

Biologisch motivierter SLAM Algorithmus

Inhaltsverzeichnis

1	Einführung	2
1.1	Einleitung	2
1.2	SLAM Algorithmen	3
1.3	Gliederung	3
2	Motivation	5
2.1	Problem gäniger SLAM-Algorithmen	5
2.2	Szenario Campusbot	5
2.3	Szenario Carolo-Cup Fahrzeug	7
3	RatSLAM	10
3.1	Die Zelltypen	10
3.2	Funktionsweise	11
4	Ziel	13
5	Risiken	15
5.1	Risiken beim CampusBot	15
5.2	Risiken bei Carolo-Cup Fahrzeugen	16
6	Ausblick	17

Zusammenfassung

Diese Ausarbeitung beschäftigt sich mit dem biologisch motivierten SLAM Algorithmus RatSLAM. Dabei werden mögliche Szenarios des Algorithmus im Forschungsprojekt FAUST beschrieben sowie die angestrebten Ziele, aber auch mögliche Risiken. Außerdem wird auf die grundlegende Funktion von RatSLAM eingegangen und ein Ausblick dargestellt.

1 Einführung

1.1 Einleitung

Eine große Aufgabe in der heutigen autonomen Robotik besteht darin, dass sich die Roboter selbstständig orientieren können. Dabei besteht die Orientierung aus zwei wesentlichen Aspekten. Der eine Teil besteht aus der Erstellung einer Karte der Umgebung, der andere Teil aus der Positionsbestimmung innerhalb der generierten Karte. Für diese Aufgaben sind bereits diverse Ansätze und Verfahren entwickelt worden. Mit Hilfe dieser Verfahren kann eine Karte der Umgebung erstellt sowie die Position innerhalb dieser Karte bestimmt werden.

Solche Verfahren sind vor allem in Krisensituationen interessant, in denen eine zuvor erstellte Karte keinen Nutzen mehr hat. Dies kann an verschiedenen Gründen liegen, wie z.B. verschütteten Gängen oder blockierten Zufahrten. Darüber hinaus ist es auch wichtig, dass ein autonom agierender Roboter sich selbst in einer unbekanntem Umgebung orientieren kann. Noch wichtiger wird dieser Punkt, wenn keinerlei Hilfe in Form von GPS- oder bald auch GALILEO-Signalen zur Verfügung steht. Und auch andere Möglichkeiten der Ortung, wie z.B. die Dreiecks-Ortung durch Funkmasten oder WLAN-Accesspoints versagen.

Solche Verfahren nennt man SLAM Algorithmen. Dabei bedeutet SLAM 'Simultaneous Localization and Mapping'. Wie schon oben angesprochen, erstellt ein solcher Algorithmus eine Karte seiner Umgebung anhand von Sensor Daten. Diese Sensor Daten können von unterschiedlichen Sensoren stammen, wie z.B. ein Laser-Scanner mit einer 360 Grad Rotation. Mögliche Alternativen stellen Ultraschall- sowie Radar-Sensoren dar. Bei einem SLAM Algorithmus wird die Position nicht exakt bestimmt, sondern nur geschätzt. Diese Schätzung erfolgt in den meisten Fällen durch Odometrie Daten. Diese Daten können von den unterschiedlichsten Sensoren stammen, wie z.B. der Umdrehungszahl des Motors, Hallgeber am Getriebe oder Inkrementalgebern an den Rädern.

Dabei bietet der biologisch motivierte SLAM Algorithmus RatSLAM (J.Milford u. a.

(2004)) einen komplett neuen Ansatz, um die Aufgabe der gleichzeitigen Lokalisierung innerhalb einer Karte und die Erstellung selbiger zu lösen. Dieser Algorithmus ist aus der Erforschung von Neurowissenschaftlern mit Ratten und deren Orientierung im Raum entstanden.

1.2 SLAM Algorithmen

Um das Problem von SLAM zu beherrschen, sind im Laufe der Zeit viele Verfahren entwickelt worden, die alle unterschiedliche Ansätze verfolgen (Siciliano und Khatib (2008)), dabei sind die bekanntesten Ansätze mittels (Extended-) Kalman-Filter, Partikel-Filter oder graphbasierende Ansätze realisiert. Dabei arbeiten aber so gut wie alle genannten Ansätze nach dem gleichen Prinzip. Sie verarbeiten die Sensor Daten inkrementell und bauen kontinuierlich eine Karte mit diesen Daten auf. Dabei wird die Position des Roboters zur Startzeit des Algorithmus als Ursprung des Koordinatensystems definiert, wodurch auch die erste Position des Roboters bekannt ist. Durch die Bewegung des Roboters wird die Karte immer erweitert und der Algorithmus versucht, schon bekannte Teile nicht erneut auszuwerten. Ein Beispiel einer so erstellten Karte ist in Abbildung 1 dargestellt.

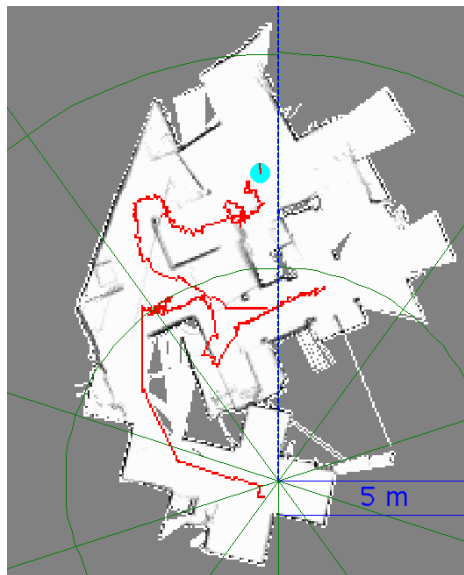


Abbildung 1: Erstellte Karte und Positionierung eines autonomen Roboters

1.3 Gliederung

Im nächsten Kapitel wird auf die Motivation für diese Arbeit eingegangen sowie mögliche Szenarios vorgestellt für zwei der drei vorhandenen Fahrzeugplattformen. In Kapitel drei wird der biologisch motivierte Algorithmus RatSLAM näher beschrieben und etwas über

seine Funktionsweise erzählt. Das vierte Kapitel beschäftigt sich mit den Zielen, die mit dieser Arbeit erreicht werden sollen. Die möglichen Risiken und deren Auswirkungen werden in Kapitel fünf beschrieben und in Kapitel sechs wird ein kurzer Ausblick dargelegt, wie an diese Arbeit angeknüpft werden kann.

2 Motivation

2.1 Problem gängiger SLAM-Algorithmen

Obwohl die meisten SLAM Verfahren bereits sehr gut erforscht sind und in den meisten Fällen sehr gut funktionieren, haben sie trotzdem jeweils ihre Vor- und Nachteile (Aulinas u. a. (2008)). Diese variieren zwischen einer enorm ansteigenden Komplexität bzw. einer extremen Verlangsamung des Algorithmus mit der Größe der Umgebung oder der Voraussetzung von exakten Sensoren. Dabei gibt es ein Problem, das viele Verfahren in der Vorverarbeitung besitzen. Dieses Problem stellt die Merkmalsextraktion dar, um so Überlappungen in der erstellten Karte zu finden. Denn viele Verfahren zur Merkmalsextraktion liefern zum einen keine 100 prozentige Wahrscheinlichkeit der Übereinstimmung von definierten Merkmalen und zum anderen müssen auch geeignete Merkmale definiert werden.

Die Aufgabe der korrespondierenden Merkmale wird noch um eine zusätzliche Aufgabe erweitert. Diese Aufgabe ist zwar nicht ganz so wichtig, da sie von den meisten Verfahren mittlerweile gut beherrscht wird, aber ein paar gängige Verfahren haben immer noch Probleme mit der Datenverwaltung. In größeren Umgebungen werden die zu verwaltenden Datenmengen, die die Sensoren produzieren, zu groß und können nicht mehr in so geringer Zeit bearbeitet werden, dass der Roboter sowohl die Erstellungen der Karte als auch die Positionsbestimmung gleichzeitig durchführen kann. Ein noch größeres Problem stellen freie Umgebungen dar, also Umgebungen, die nicht in einem geschlossenen Bereich liegen, wie z.B. Räume oder Straßen in Großstädten. In freien Umgebungen liefern viele Sensorarten nicht mehr definierbare Ergebnisse zurück, da sie in den seltensten Fällen auf einen Gegenstand treffen und somit der Sensor kein Signal zurück bekommt. Dies hat zur Folge, dass auch innerhalb kürzester Zeit die zu verwaltende Datenmenge ebenfalls eine abnorme Größe annehmen und der SLAM Algorithmus kein befriedigendes Ergebnis mehr liefern kann.

2.2 Szenario Campusbot

Die Fahrzeugplattform CampusBot, die ursprünglich auf dem VolksBot XT (VolksBot) basiert, wird dazu entwickelt, um mit Menschen auf dem Campus Berliner Tor der Hochschule für Angewandte Wissenschaften Hamburg zu interagieren. Um diese Interaktion zu erreichen, ist die Plattform mit verschiedenen Sensoren ausgestattet, um diverse Aufgaben erfüllen zu können. Der CampusBot verfügt neben einer normalen Kamera, die zur Zeit abgeschaltet ist, über die Kinect von Microsoft zur Erkennung sowie Verfolgung von Personen und Hindernissen. Darüber hinaus ist ein vier Ebenen Laser-Scanner der Firma Sick installiert und es sind Inkrementalgeber an den Rädern vorhanden. Zusätzlich zu dieser Sensorik soll noch ein Greifarm installiert werden, damit der CampusBot auch selbstständig mit seiner Umgebung interagieren kann, wie z.B. den Fahrstuhl zu bedienen oder Gegenstände zu greifen.

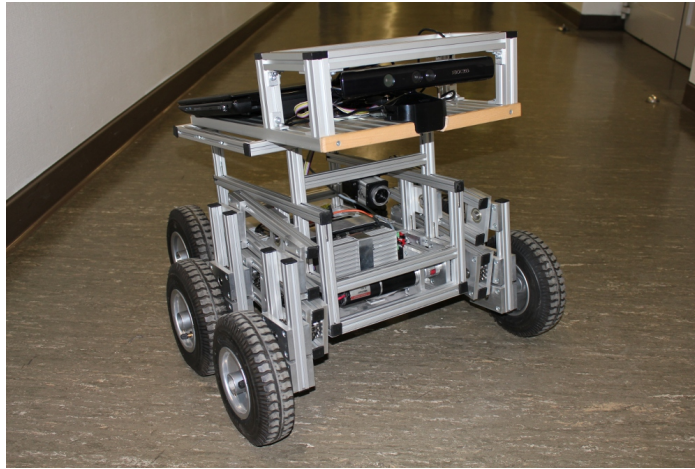


Abbildung 2: Der CampusBot

Bisher ist auf dem CampusBot mittels des Laser-Scanners ein FastSLAM realisiert worden. Dabei benötigt zum einen der Laser-Scanner nicht gerade wenig Energie, zum anderen unterliegt FastSLAM ebenfalls dem schwerwiegendem Nachteil, dass die Datengröße ab einem gewissen Zeitpunkt nicht mehr gut vom Algorithmus auf Übereinstimmungen durchsucht werden kann. Ein weiterer Nachteil neben dem hohen Energieverbrauch des Laser-Scanners ist auch die enorme Rechenleistung, die benötigt wird, um in recht kurzer Zeit verwertbare Daten zu erhalten. Um genau zu sein, belegt die Berechnung der Daten vier Threads auf dem Intel Core i7 Mobile von acht möglichen Threads.

Darüber hinaus liefert FastSLAM keine perfekte Karte, die auf einen Gebäudeplan gemappt werden könnte. Mit wachsender Entfernung, die der CampusBot vom Startpunkt aus zurücklegt, leidet die Genauigkeit der erstellten Karte und ein genaues Mapping wäre nicht mehr möglich. Durch diesen Umstand kann auch keine Routenplanung anhand der Daten durchgeführt werden, indem auf einen bestimmten Bereich der Karte getippt wird und der CampusBot sich selbstständig dorthin bewegt.

RatSLAM, ein biologisch motivierter SLAM Algorithmus, kann anstelle der herkömmlichen SLAM Algorithmen zur Kartierung des Campus Berliner Tor der HAW Hamburg eingesetzt werden. Die Kartierung des Campus könnte noch dahingehend erweitert werden, dass zusätzlich die auf dem Campus befindlichen Gebäude Stockwerk für Stockwerk mit erfasst werden. Da allein die Daten der Karte des Campus ohne Gebäude schon enorm sein sind, kann kein normaler SLAM Algorithmus eingesetzt werden. RatSLAM ermöglicht eine recht genaue Karte, die sogar auch sehr gut in ein Gebäudeplan übertragen werden könnte.

Es sollen zwei mögliche Arten zur Erstellung einer Karte des Campus zur Verfügung stehen. Die eine Methode ist geführt, d.h., dass der Benutzer mittels der Tablet Fernbedienung dem CampusBot zeigt, wie es auf dem Campus aussieht. Dabei müsste der Benutzer zusam-

men mit dem CampusBot alle Wege des Campus sowie alle Gänge und Flure der Gebäude durchlaufen. Die andere Möglichkeit besteht darin, den CampusBot selbstständig das Gelände und die Gebäude ergründen zu lassen. Im Gegensatz zur ersten Variante müsste zwar kein Benutzer anwesend sein um dem CampusBot die gesamte Umgebung zu zeigen, allerdings ist die zweite Variante deutlich zeitintensiver, da die Möglichkeit besteht, dass er mehrmals die gleichen Orte durchfährt. Dies könnte bei einer geführten Kartierung vermieden werden.

Dieser Ansatz kann dadurch erweitert werden, dass einem Benutzer die Möglichkeit gegeben wird, den CampusBot vollständig autonom auf dem Campus Berliner Tor fortzubewegen. Diese autonome Bewegung könnte dadurch erreicht werden, dass dem Benutzer auf der Tablet-Fernbedienung der Campus Berliner Tor angezeigt wird. Soll sich der CampusBot nur außerhalb der Gebäude bewegen, so würde der Benutzer auf einen bestimmten Punkt auf der Karte tippen und der CampusBot würde sich anhand seiner internen Karte zu dieser Stelle bewegen. Klickt der Benutzer auf eines der Gebäude, so wird ihm eine Auswahl von Stockwerken zur Verfügung gestellt und der Benutzer kann eine entsprechende Etage auswählen. In der Karte des selbigen Stockwerks wählt der Benutzer einen Punkt aus, zu dem der CampusBot dann fährt.

2.3 Szenario Carolo-Cup Fahrzeug

Die Carolo-Cup Fahrzeuge bestehen mittlerweile aus zwei Generationen von Fahrzeugen. Die erste Generation besteht aus den Fahrzeugen Onyx und Rubin. Beide Fahrzeuge sind baugleich und bauen auf dem 1:10 RC-Modellauto Tamiya Ford F-350 auf. Anstelle der Ladefläche wurde eine IO-Box installiert, die aus zwei Mikrocontrollern (NXP LPC24xx), der Stromversorgung für die Mikrocontroller sowie der Sensorik und Erweiterungsboards für die Mikrocontroller besteht. Als Sensorik verwenden diese Fahrzeuge vorne und hinten jeweils zwei Ultraschall Sensoren. An jeder Seite befinden sich jeweils zwei Infrarot Sensoren zur Distanzmessung von Sharp (GP2Y0A21YK0F). Die zurückgelegte Strecke wird mittels Inkrementalgebern an beiden Vorderrädern entnommen. Die Bildbearbeitung und Sensor Auswertung übernimmt dabei ein Aspire One von Acer, das zu der frühesten Generation gehört. Dabei ist das Netbook mit einer Solid-State-Disk ausgestattet, um auch größere Erschütterungen aushalten zu können.

Die zweite Generation von Carolo-Cup Fahrzeugen basiert auf dem 1:10 RC-Modellauto HB TCX von hpi racing. An diesem Fahrzeug werden zwei Arten von Infrarot Sensoren zur Distanzmessung von Sharp verwendet. Zum einen der Sharp GP2Y0A02YK, der einen Erfassungsbereich von 20 bis 150 Zentimetern hat und zum anderen der Sharp GP2Y0A41SK0F, der einen Messbereich von vier bis 30 Zentimetern hat. Um die zurückgelegte Strecke messen zu können, soll ein Laser-Maus-Sensor von Avago (ANDS-9500) verwendet werden, mit dem es voraussichtlich auch möglich sein wird, ein seitliches Driften des Fahrzeuges zu detektieren. Während auf der ersten Generation noch Netbooks eingesetzt

werden, ist auf der zweiten Generation ein Embedded PC von Axiomtek (Pico820VGA) verbaut, der die komplette Bildverarbeitung und Sensor Auswertung übernimmt. Der Embedded PC arbeitet dabei mit einer CompactFlash Karte, um ebenfalls Erschütterungen widerstehen zu können.



Abbildung 3: 1. Generation



Abbildung 4: 2. Generation

Die Herausforderung während des studentischen Wettbewerbs Carolo-Cup (Carolo-Cup (b)) ist neben dem Einparken und der Hinderniserkennung auch die Spurerkennung. Dabei ist laut Reglement vorgeschrieben, dass bis zu einem Meter der Fahrbahnmarkierung von zwei Linien gleichzeitig fehlen dürfen (Carolo-Cup (a)). Diese fehlenden Teilstücke muss der verwendete Algorithmus selbst ausgleichen und darf dabei nicht die vorgegebene Strecke verlassen oder seitlich Ausbrechen. Dabei könnte die Spurerkennung und -führung durch einen SLAM Algorithmus unterstützt werden.

In den vergangenen Semestern sind viel Zeit und Überlegung in diverse Ansätze für ein mögliches Mapping Verfahren zur Kartierung der Fahrstrecke investiert worden. Dabei ist allerdings nie ein echtes Mapping Verfahren entstanden, was den Anforderungen gerecht wurde. Die Hauptaufgaben, die dabei immer wieder aufgetreten sind, sind die Synchronisierung und die verwendete Hardware. Die Synchronisierung sollte stattfinden, wenn das Fahrzeug die Strecke einmal verlassen sollte, dabei ist es aber schwer, einen neuen Synchronisierungspunkt zu finden und so das Fahrzeug in der erzeugten Karte neu zu positionieren, außer es fährt über die Startlinie. Dies ist eine viel zu lange Strecke. Werden die Verfahren zum Mapping aufwändiger, kommt die verwendete Hardware der PCs schnell an ihre Grenzen und kann die Datenverarbeitung nicht mehr in Echtzeit bewältigen.

Da RatSLAM die zu verarbeitende Datenmenge im Vergleich zu herkömmlichen SLAM Verfahren sehr gering hält und auch die Fahrstrecke begrenzt sowie nicht sehr lang mit ca. 43 Metern ist, kann es möglich sein mit diesem biologisch motiviertem Algorithmus eine gute Alternative gefunden zu haben, um eine perfekte Karte zu erzeugen. Ein weiterer Vorteil von RatSLAM gegenüber einem Mapping Verfahren ist die sofortige Neu-Synchronisierung des

Fahrzeuges innerhalb der Karte ohne aufwändige Synchronisationspunkte, da RatSLAM sehr schnell wieder Übereinstimmungen mit der gespeicherten Karte finden kann.

3 RatSLAM

Das biologisch motivierte SLAM Verfahren RatSLAM (J.Milford u. a. (2004)) ist eine Entwicklung die aus den Erkenntnissen der Neurowissenschaft mit Ratten basiert. Dabei konnte der Algorithmus nur entwickelt werden, weil die Neurowissenschaften verschiedener Universitäten in den letzten Jahren diverse Zelltypen in dem Gehirn von Ratten entdeckt haben, die den Orientierungssinn beschreiben. Mittels dieses Orientierungssinnes können die Ratten verschiedene Merkmale aus der Umgebung extrahieren und sich so in einem Raum orientieren. Die Informatiker Milford, Wyeth und Prasser, die zu diesem Zeitpunkt alle an der University of Queensland in Australien ansässig waren, haben diese Zelltypen zwar nicht exakt nachgebaut, sondern nur die besten Eigenschaften genommen. Mithilfe dieser Eigenschaften haben sie ein neuronales Netz entwickelt, was einen solchem Orientierungssinn nachempfunden ist, allerdings auch mittels Computer berechnet werden kann. Dabei basiert der Algorithmus von RatSLAM im Wesentlichen auf einer optischen Wahrnehmung. Es werden also in der Regel keine Tiefensensoren benötigt, wie z.B. Laser-Scanner, Ultraschall- oder Infrarot-Sensorik. Zur Stabilisierung der Positionsbestimmung können Daten von Sensoren wie Inkrementalgebern oder Hall-Sensoren an dem Getriebe- rad entnommen werden, dies ist allerdings nicht unbedingt notwendig.

3.1 Die Zelltypen

Insgesamt sind bisher sechs Arten von Gehirnzellen identifiziert worden, die für den Orientierungssinn einer Ratte zuständig sind. Interessant dabei ist, dass einige von diesen Zellarten auch im menschlichen Gehirn nachgewiesen worden sind. Von diesen sechs Arten sind drei Arten der Grundbaustein für die Orientierung und die Funktionsweise dieser Zelltypen sind in dem Kern von RatSLAM eingeflossen.

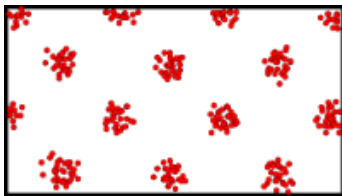


Abbildung 5: Grid Cells

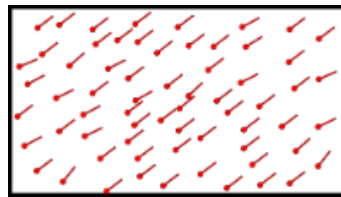


Abbildung 6: Head Direction Cells

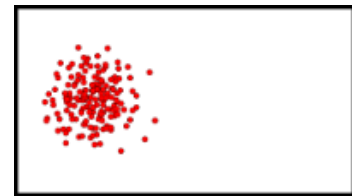


Abbildung 7: Place Cells

Die Grid Cells, die in Abbildung 5 dargestellt sind, sind regelmäßig über den Raum verteilt. Diese Zellarten sind an verschiedenen Orten in einem Raum aktiv und wurden 2005 an der NTNU (Norwegen) entdeckt. In Abbildung 6 sind die Head Direction Cells abgebildet. Diese Zelltypen zeigen nicht den Aufenthaltsort einer Ratte in einem Raum an, sondern die Ausrichtung des Kopfes. Dabei orientiert sich die Ausrichtung des Kopfes an einem globalen Koordinatensystem und funktioniert somit wie eine Art innerer Kompass. Die Striche,

die von den Punkten wegführen, zeigen theoretisch die momentane Ausrichtung des Kopfes an. Place Cells, abgebildet in Abbildung 7, reagieren im Allgemeinen nicht auf Orientierung bzw. Blickrichtung einer Ratte in einem Raum, sondern aktivieren sich nur in der näheren Umgebung eines bestimmten bzw. prägnanten Ortes.

3.2 Funktionsweise

Um eine Trennung zwischen der biologischen Sicht (Kapitel 3.1) und der Sicht aus der Informatik zu bekommen und Verwechslungen vorzubeugen, wird eine neue Struktur eingeführt (J.Milford und Wyeth (2008)), die aus den sogenannten Local View Cells, Pose Cells und der Experience Map besteht. Diese neue Struktur ist in Abbildung 8 abgebildet. Dabei gibt es zwei Arten von Signalen. Auf der einen Seite sind das die External Cues und auf der anderen die Self Motion Cues. Dabei werden diese Signale reinweg aus dem Bild der Kamera gewonnen. Dieses System ist anders als RatSLAM nicht biologisch motiviert.

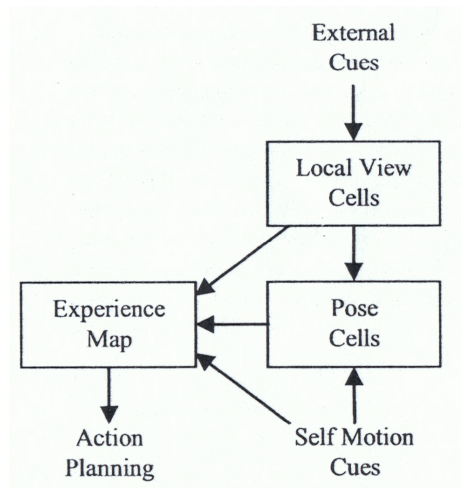


Abbildung 8: Prinzipieller Aufbau von RatSLAM

Die Pose Cells stellen die drei Freiheitsgrade (DOF) des Roboters dar und benutzen dafür CAN. CAN steht für Continuous Attractor Networks und ist ein spezieller Typ von neuronalen Netzwerken. Sie haben ein Array von neuronalen Einheiten mit gewichteten Kanten. Mit diesen speziellen neuronalen Netzwerken werden sehr häufig die Place, Head Direction und Grid Cells modelliert (Samsonovich und McNaughton (1997)). Die dreidimensionale Struktur der Pose Cells ist untereinander verbunden, indem jede Fläche mit der gegenüberliegenden Fläche mittels sogenannten wraparound connections verbunden ist. Befindet sich das System in einer großen Umgebung und bewegt sich gleichzeitig, umlaufen die Aktivitäten die Pose Cell Struktur sehr häufig, was in aktivierten bzw. feuernenden Feldern resultiert. Damit ist gemeint, dass mehrere einzelne und zusammenhängende neuronale Einheiten auf einer Fläche aktiviert werden. Dieses Verhalten ähnelt dem von Grid Cells.

Die Pose Cells werden durch die Self Motion Cues aktualisiert und durch eine Local View kalibriert. Dabei werden die Self Motion Cues dazu benutzt, um eine Integration des Pfades in die Pose Cells zu erlangen, während die Local View Cells von den External Cues aktualisiert werden und über ein assoziatives Lernen Einfluss auf die Pose Cells ausüben.

Die Experience Map soll die Informationen der Zellen und der Eigenbewegungssignale (Self Motion Cues) in Erfahrungen umwandeln und organisieren. Dabei wird eine Erfahrung durch die Verbindung von zwei Codes definiert: dem Pose Code und dem Local View Code. Der Pose Code wird von den Pose Cells erzeugt und der Local View Code von den Local View Cells. Sollte sich einer der Codes oder beide verändern, bedeutet dies, dass eine neue Erfahrung angelegt werden muss, die mit der vorherigen Erfahrung über die aktuelle Distanz, die durch die Eigenbewegungssignale gewonnen worden ist, verbunden wird. Sollte der Roboter bereits bekannte Umgebungen erneut durchfahren, merkt er dies durch die gleichen Codes, die bereits in Erfahrungen gespeichert sind. Die Korrektur der Erfahrungskarte (Experience Map) wird durch die Ausrichtung der Lokalisation und Orientierung von Erfahrungen mit den Informationen der Transitionen, die zwischen den Erfahrungen gespeichert sind, durchgeführt.

Eine Local View Cell wird aktiviert, wenn die zuvor gesehene Umgebung, die mit dieser Zelle verbunden ist, große Ähnlichkeiten mit der aktuellen Umgebung aufweist. Dabei besitzt jede Local View Cell ein Template, das aus einem Scanline Intensity Profil erzeugt wird. Sollte nun eine Übereinstimmung gefunden werden zwischen dem aktuellen Scanline Intensity Profil und einem gespeicherten Template, wird die Local View Cell aktiv. Kann keine Übereinstimmung gefunden werden, wird eine neue Local View Cell mit einem entsprechenden Template angelegt. Die Scanline Intensity Profile werden dabei von den Bildverarbeitungsalgorithmen erzeugt und sind ein eindimensionaler Vektor, der durch die Summierung der Intensitätswerte in jeder vertikalen Pixel-Reihe generiert wird. Nach der Bildung des Vektors wird dieser noch normiert. Dabei werden diese Profile neben der Verwendung in den Local View Cells auch für die Rotation des Roboters sowie zur Geschwindigkeitsbestimmung genutzt (Campbell u. a. (2005)).

4 Ziel

Das Ziel dieser Arbeit soll zunächst die Implementierung des biologisch motivierten SLAM Algorithmus RatSLAM in die bestehende Software Umgebung des Forschungsprojektes FAUST sein. Dabei muss zunächst der Quell-Code für Matlab, den die University of Queensland bereitstellt, analysiert werden und als ein Plugin für den FAUSTcore umgeschrieben werden in der Programmiersprache C++. Darüber hinaus muss evaluiert werden, inwieweit die verwendeten Variablen innerhalb des Matlab-Codes effizient in C++ umzusetzen sind und ob es möglicherweise effizientere Variablentypen oder Datenstrukturen gibt, die genutzt werden können.

Wenn die Implementierung gelungen ist, soll der Algorithmus um eine Komponente erweitert werden, die von der Technischen Universität Chemnitz entwickelt wird und RatSLAM eine bessere Merkmalsextraktion in dynamischen Umgebungen liefert. Dieses Ziel ist besonders wichtig, wenn sich der CampusBot autonom über den Campus bewegen soll, denn es kann nicht immer davon ausgegangen werden, dass die Wege, die der CampusBot entlang fährt, menschenleer sind. Gleiches gilt auch für die Flure und Räume innerhalb der Gebäude. Denn auch diese können ihre Struktur verändern, z.B. durch Pappkartons oder Menschen, die sich in den Gängen bewegen. Die Kartierung der Strecke während des Carolo-Cups würde davon ebenfalls profitieren, denn während des Trainingstages sitzen kaum bis keine Menschen um die Fahrbahn herum. Bei dem Wettbewerb ist die komplette Fahrbahn mit Menschen besetzt und trotz dieser Veränderung muss sich das Fahrzeug fehlerfrei orientieren können.

Wie schon in dem Szenario (Kapitel 2.2) beschrieben worden ist, wäre ein mögliches weiteres Ziel die Entwicklung einer Android App. oder die Erweiterung einer bestehenden Android App. mit der Funktion zur Anzeige der erstellten Karte, in der dann auch durch die verschiedenen Stockwerke eines Gebäudes angezeigt werden können. Dieses Konzept einer Android App. kann noch dadurch verfeinert werden, wenn ein Gebäude Plan auf die von RatSLAM erstellte Karte projiziert wird. Darüber hinaus soll es möglich sein, mittels der Tablet Fernbedienung dem CampusBot ein Ziel vorzugeben, welches dann autonom von dem CampusBot angefahren wird. Eine mögliche Alternative dafür ist, sofern der Benutzer es wünscht, dem Benutzer den Weg von seiner aktuellen Position zum angegebenen Ziel zu zeigen und interaktiv auf der Karte den Weg zu zeigen.

Da die Software FAUSTcore unter einem Real-Time Linux Kernel läuft, der hart echtzeitfähig ist, sollte das biologisch motivierte SLAM Verfahren RatSLAM ebenfalls eine solche Eigenschaft aufweisen in Form von einer sehr schnellen Bearbeitung sowie Berechnung aller Vorgänge. Sollte sich in den Tests zur Evaluierung der Echtzeitfähigkeit von RatSLAM herausstellen, dass der Algorithmus deutlich länger braucht zur Abarbeitung, gibt es zwei Möglichkeiten dieser Aufgabe entgegenzuwirken. Die erste und trivialste Möglichkeit ist die Erhöhung der Scheduler Zeit. Diese Erhöhung ist auf dem CampusBot deutlich unkritischer als auf den Carolo-Cup Fahrzeugen, da sich der CampusBot mit einer deutlich

geringeren Geschwindigkeit fortbewegt. Dann ist eine Implementierung auf der Fahrzeugplattform Carolo-Cup nicht mehr möglich. Die zweite, aber nicht leichter zu realisierende Möglichkeit ist die Erforschung einer Echtzeitfähigkeit, um RatSLAM auf beiden Fahrzeugplattformen zu implementieren und zu verwenden. Diese kann im Falle einer Notwendigkeit nur durch besonders schnelle Mathematik- und Such-Algorithmen realisiert werden.

5 Risiken

Die auftretenden Risiken müssen bei RatSLAM in zwei Kategorien eingeteilt werden, da sie sich unterschiedlich verhalten. Das sind zum einen die Risiken, die entstehen können, wenn RatSLAM auf dem CampusBot implementiert wird. Zum anderen sind es die Risiken, die bestehen, wenn eine Implementierung bzw. Portierung von RatSLAM auf die Carolo-Cup Fahrzeuge durchgeführt wird.

5.1 Risiken beim CampusBot

Das größte Risiko, das bei der Fahrzeugplattform CampusBot existiert, ist der Umgang mit dynamischen Umgebungen. Es gibt zwar eine gewisse Erweiterung für RatSLAM, um mit einer Umgebung zurechtzukommen, die ihre Merkmale verändert, allerdings ist in dieser Veröffentlichung nicht erwähnt, in welchem Umfang diese Erweiterung diesen dynamischen Aspekt herausrechnen kann. Denn in dieser Veröffentlichung (Suenderhauf u. a. (2010)) wird die Erweiterung nur auf bereits vorhandenen Videos, die die Wissenschaftler der University of Queensland aufgenommen haben, angewendet. Dabei ist die Dynamik nicht besonders hoch, da in diesem Stadtteil, in dem das Video entstanden ist, kein besonders hohes Verkehrsaufkommen zum Zeitpunkt der Aufnahme herrschte.

Es ist allerdings wünschenswert, dass sich der CampusBot nicht nur durch ein oder zwei Menschen, die sich zur Zeit der Fahrt des CampusBots auf dem Campus befinden, hindurchfahren kann, sondern auch größere Gruppen oder gar größere Menschenmassen dem Algorithmus RatSLAM keine Probleme bereiten würden. Sollte diese Erweiterung in solch einem Falle nicht funktionieren, muss eine Alternative gefunden werden. Sollte keine Alternative zu finden sein, besteht das Risiko im Wesentlichen darin, dass die verwendete Lösung nicht in jedem Fall mit einer dynamischen Umgebung umgehen kann. In solch einem Fall muss dann eine Notfallstrategie entwickelt werden, wie der CampusBot sich zu verhalten hat, um nicht unkontrolliert in eine solche Menschenmasse hineinzufahren.

Zurzeit verwendet der CampusBot als einzige Kamera eine Microsoft Kinect. Mittels dieser Kamera erkennt der CampusBot auf der einen Seite Gegenstände um diesen auszuweichen, auf der anderen Seite kann er mittels der Tiefensensorik auch Personen verfolgen. Das Problem an der Microsoft Kinect ist, dass sie eigentlich für den stationären Einsatz, also z.B. auf einem Fernseher oder einem Tisch konzipiert ist und nicht für ein sich bewegendes Objekt. Die Frage ist allerdings, ob das normale Kamerabild der Microsoft Kinect ausreicht und schnell genug sowie überhaupt abgreifbar ist, um dieses Bild mit RatSLAM zu verarbeiten oder ob eine neue, zusätzliche Kamera installiert werden muss, damit RatSLAM funktionieren kann. Dies ist zwar kein so großes Risiko, dass eine Implementierung scheitern kann, allerdings wird diese Maßnahme die Energiebilanz erneut verschlechtern.

5.2 Risiken bei Carolo-Cup Fahrzeugen

Bei der Fahrzeugplattform Carolo-Cup besteht das Risiko auf der einen Seite in der doch sehr stark beschränkten Hardware. Auf diesen arbeitet nur ein Embedded PC bzw. ein Netbook und beide sind nur mit einem Intel Atom ausgestattet. Dabei ist das Problem, dass in keiner Veröffentlichung, die RatSLAM als Thema haben, erläutert wird, auf was für einer Hardware Plattform RatSLAM getestet wird und inwieweit es generell eine Echtzeitfähigkeit erreichen kann. Die Echtzeitfähigkeit ist aber unbedingt notwendig, da jeder Task nur 25 Millisekunden zur Verfügung stehen, um alle notwendigen Berechnungen durchzuführen. Sollte die Grenze von 25 Millisekunden überschritten werden, wird der Task der Prozessorkern entzogen und eine neue Task gestartet. Im Falle von RatSLAM hieße dies, dass eine Karte niemals oder niemals korrekt erstellt werden kann sowie die Positionsbestimmung innerhalb der Karte nicht durchgeführt werden kann oder die Positionsbestimmung enorm fehlerträchtig ist.

Wie es schon bei dem CampusBot erwähnt worden ist (Kapitel 5.1), besteht auch auf den Fahrzeugplattformen Carolo-Cup das Problem von dynamischen Umgebungen. Um die gesamte Fahrbahn sind in einem Rechteck Stühle für die Zuschauer aufgestellt. Bei der Erstellung einer Karte am Trainingstag sind diese Stühle in der Regel nicht besetzt und verursachen somit keine größeren Veränderungen im Bild und damit in der Merkmalsextraktion. Am Wettbewerbstag sind sämtliche Stühle und auch die Reihen hinter den Stühlen mit Personen besetzt. Durch den sehr großen Sehbereich des verwendeten Objektivs, das ein Fish-Eye Objektiv mit einem Blickwinkel von 185 Grad ist, sieht die Kamera auch weite Teile des Publikums, das sich stetig verändert und das in einem nicht geringen Umfang. Es könnte nun passieren, dass der biologisch motivierte Algorithmus RatSLAM mit dieser Situation überfordert ist und keine Übereinstimmungen mehr erkennen kann und somit augenscheinlich immer in einer ihm unbekanntem Umgebung agiert und dadurch kontinuierlich die Karte erweitert.

Ein mögliches anderes Risiko, das den Einsatz auf den Carolo-Cup Fahrzeugen erschweren oder sogar verwehren könnte, ist die hohe Frequenz, mit der Bilder aufgenommen und verarbeitet werden. Die verwendete Kamera hat eine maximale Bildwiederholrate von 87 Stück. Allerdings arbeiten die Kameras zurzeit mit einer Rate von 44 Stück pro Sekunde. Dies liegt auch an den 25 Millisekunden, die jede Task Zeit hat zu rechnen. Damit sollen vor allem doppelte Bilder sowie unnötige Berechnungen vermieden werden. Der RatSLAM Algorithmus wurde allerdings bisher nur mit normalen Kameras bzw. Webcams getestet, die normalerweise eine Bildwiederholrate von 24 haben und damit deutlich unter der verwendeten Rate liegen. Hier ist es fraglich, ob die hohe Rate in Form von einer zu großen und gleichen Datenmenge dem Algorithmus Probleme bereitet oder ob RatSLAM mit dieser hohen Informationsflut umgehen kann.

6 Ausblick

Als weiterführende Arbeit könnte auf dem CampusBot die mittels RatSLAM erzeugte Karte mit Bedeutungen versehen werden. Dabei wäre es durchaus interessant, wenn der CampusBot selbst diese Bedeutungen definieren könnte, indem ein neuronales Netz bestimmte Merkmale im Bild klassifiziert. Die Alternative zur selbstständigen Erkennung wäre eine manuelle Einpflegung der Bedeutungen. Dabei ist mit Bedeutung gemeint, dass ein Raum nicht nur ein Ort auf der Karte ist, sondern als Beispiel auch die Raumnummer erfasst wird. Darüber hinaus könnte auch der Standort von Toiletten und Bibliotheken vermerkt sein sowie die Lage der Mensa.

Dann könnte der Benutzer über die Tablet Fernbedienung sagen, wo er gerne hin möchte. Dies könnte eine bestimmte Raumnummer sein oder nur die nächstgelegene Toilette. Ist die Eingabe erfolgt, kann der CampusBot dann durch seine eigene Position und der Position auf der Karte dem Benutzer entweder mitteilen, wo er den gesuchten Ort findet oder der CampusBot geleitet den Benutzer sofort zu dem gesuchten Ort.

Literatur

- [Aulinas u. a. 2008] AULINAS, J. ; PETILLOT, Y. ; SALVI, J. ; LLADO, X.: The SLAM problem: a survey / University of Girona, Heriot-Watt University. 2008. – Forschungsbericht
- [Campbell u. a. 2005] CAMPBELL, J. ; SUKTHANKAR, R. ; NOURBAKHSI, I. ; PAHWA, A.: A Robust Visual Odometry and Precipice Detection System Using Consumer-grade Monocular Vision / Carnegie Mellon University, Intel Research Pittsburgh, NASA Ames Research Center. 2005. – Forschungsbericht
- [Carolo-Cup a] CAROLO-CUP: *Regelwerk 2012*. – URL <http://www.carolo-cup.de/uploads/media/Regelwerk.pdf>
- [Carolo-Cup b] CAROLO-CUP: *Website*. – URL <http://www.carolo-cup.de/>
- [J.Milford und Wyeth 2008] J.MILFORD, M. ; WYETH, G. F.: Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System / University of Queensland. 2008. – Forschungsbericht
- [J.Milford u. a. 2004] J.MILFORD, M. ; WYETH, G. F. ; PRASSER, D.: RatSLAM: A Hippocampal Model for Simultaneous Localization / University of Queensland. 2004. – Forschungsbericht
- [Samsonovich und McNaughton 1997] SAMSONOVICH, A. ; MCNAUGHTON, B. L.: *Path integration and cognitive mapping in a continuous attractor neural network model*. Bd. 17. S. 5900 – 5920, Journal of Neuroscience, 1997
- [Siciliano und Khatib 2008] SICILIANO, Bruno ; KHATIB, Oussama: *Springer Handbook of Robotics*. Springer, 2008
- [Suenderhauf u. a. 2010] SUENDERHAUF, N. ; NEUBERT, P. ; PROTZEL, P.: The Causal Update Filter - A Novel Biologically Inspired Filter Paradigm for Appearance Based SLAM. / Technische Universität Chemnitz. 2010. – Forschungsbericht
- [VolksBot] VOLKSBOT: *entwickelt vom Fraunhofer Institut IAIS*. – URL <http://www.volksbot.de/>

Abbildungsverzeichnis

1	Erstellte Karte und Positionierung eines autonomen Roboters	3
2	Der CampusBot	6
3	1. Generation	8
4	2. Generation	8
5	Grid Cells	10
6	Head Direction Cells	10
7	Place Cells	10
8	Prinzipieller Aufbau von RatSLAM	11