

Computer Systems and Telematics — Distributed, Embedded Systems

Bachelorarbeit

Design und Implementierung eines mobilen Referenzsystems für die Indoorlokalisierung

Benjamin Aschenbrenner

Matr. 4292264

Simon Schmitt

Matr. 4287788

Betreuer: Prof. Dr. rer. nat. Mesut Güneş
Betreuender Assistent: Dipl.-Inf. Heiko Will

Institut für Informatik, Freie Universität Berlin, Deutschland

16. Juli 2011

Text so ok, oder lieber zwei Texte? Unterschreibt so auch jeder dass der andere nicht abgeschrieben hat? Wir versichern, dass wir die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, sind als solche gekennzeichnet. Die Zeichnungen oder Abbildungen sind von uns selbst erstellt worden oder mit entsprechenden Quellennachweisen versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner Prüfungsbehörde eingereicht worden.

Berlin, den 16. Juli 2011

(Benjamin Aschenbrenner)

(Simon Schmitt)

Zusammenfassung

Zusammenfassung

Das Forschungsprojekt Indoorlokalisierung der Arbeitsgruppe Computer Systems and Telematics beschäftigt sich mit der Positionsbestimmung in GPS-freien Umgebungen anhand von Signallaufzeiten zwischen mobilen Sensorknoten. Der Schwerpunkt dieser Bachelorarbeit liegt darin, ein möglichst genaues Testsystem für die Indoorlokalisierung anhand von visuellen Daten zu entwickeln. Dabei werden Microsoft Kinects eingesetzt, durch die es möglich wird, Räume dreidimensional zu erfassen.

Abstract

The indoor localization research project of the computer systems and telematics departement works on positioning determination in gps-less areas using signal propagation delays between mobile sensory nodes. The main focus of this bachelor thesis is to develop an accurate testing system for indoor localization based on visual data. In doing so, Microsoft Kinects are used, which enable three dimensional room capturing.

Inhaltsverzeichnis

Abbildungsverzeichnis	ix
Tabellenverzeichnis	xi
Quellcodeverzeichnis	xiii
Glossar	xv
Akronyme	xvii
1 Einleitung	1
1.1 Motivation	1
1.2 Lösungsansätze	1
1.3 Aufgabenstellung	2
2 Umsetzung	3
2.1 Konzept	3
2.2 Soft- und Hardware	3
2.2.1 Microsoft Kinect	3
2.2.2 Robot Operating System	3
2.2.3 OpenNI Stack	3
2.2.4 Turtle Bot	4
2.3 Pathfinder	4
2.4 NavStack	4
3 Analyse	5
3.1 Testlauf	5
3.2 Genauigkeit	5
3.3 Fazit	5
4 Ausblick	7
5 Anhang	9
Literaturverzeichnis	11

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Glossar

Robot Operating System Ein Framework für Roboter das Hardwareabstraktion, eine Paketverwaltung, ein Nachrichtensystem und die Möglichkeit zum Betreiben auf mehreren Computern liefert. xvii, 3

Akronyme

ROS Robot Operating System. 3

KAPITEL 1

Einleitung

1.1 Motivation

lorem ipsum!

1.2 Lösungsansätze

Es gibt zahlreiche Möglichkeiten der Lokalisierung. Damit die Entfernungsmessung der Sensorknoten evaluiert werden kann, werden unabhängige Daten benötigt. **Das heißt, dass nicht auf das bereits bestehende Testbed zurückgegriffen werden kann.** Im Folgenden werden einige andere Möglichkeiten beschrieben und ihre Vor- sowie Nachteile bezüglich der Anwendung in einer Testimplementierung erörtert.

Die einfachste Möglichkeit wäre, die Positionsbestimmung manuell durchzuführen. Dabei könnte eine Testmessung auf einer zuvor festgelegten Strecke durchgeführt werden. Eine Messung der Strecke per Hand würde zwar verlässliche tatsächliche Positionsdaten liefern, jedoch wäre der zu betreibende Aufwand für viele verschiedene Strecken enorm. Um eine robuste Evaluierung realisieren zu können, ist es jedoch gerade von Nöten, Messungen auf verschiedenen Strecken umzusetzen. Theoretisch denkbar wäre eine Verwendung des Global Positioning Systems, das eine dynamische Positionsbestimmung erlaubt. Jedoch wäre man dann auf das GPS Signal angewiesen, und könnte keine Ortung in Gebäuden durchführen. Nötig wären Versuchsaufbauten im Freien. Nur könnten hier nicht die gleichen Bedingungen erreicht werden, die tatsächlich innerhalb eines geschlossenen Gebäudes herrschen. Durchaus von Interesse wären dagegen Odometriedaten. Odometrie wird von einem fahrenden Objekt von dessen Antriebssystem geliefert. Dieses kann mit einer gewissen Wahrscheinlichkeit sagen, dass das Objekt aufgrund der Aktivierung gewisser Vortriebsmechanismen eine bestimmte Wegstrecke zurückgelegt hat. Ebenfalls bestimmbar muss die Richtung sein, in die die Bewegung ausgeführt wurde. Allerdings kann diese Methode nicht absolut fehlerfrei sein. Dadurch summieren sich kleinste Fehler mit der Zeit so weit auf, dass eine Positionsbestimmung unmöglich wird. Wird jedoch diese Art der Lokalisierung mit Anderen kombiniert, erhält man serwohl eine gute probabilistische Position. Eine weitere Möglichkeit wäre eine ständige Positionsbestimmung anhand von Kameras. Dabei können Bewegungen zwischen

zwei aufgenommenen Bildern erfasst werden. Handelt es sich um dreidimensionale Bildinformationen, können schon heute relativ robuste Aussagen über eine zurückgelegte Wegstrecke getroffen werden. Allerdings handelt es sich hierbei um ein aktuelles Forschungsgebiet, in dem es oft noch keine optimalen Lösungen für dabei auftretende Probleme gibt. Diese Art der Lokalisierung entspricht sinnhaft in etwa der der Odometriedaten. Es handelt sich hier quasi um visuelle Odometrie. Deshalb können sich auch hier theoretisch Fehler aufaddieren. Dennoch kann diese Methode ungleich genauer sein, da sie lediglich von der Güte der Daten und den verwendeten Algorithmen abhängt. So kann ein solches System beispielsweise auf eine zurückgelegte Wegstrecke zurückblicken, und anhand des neuen Blickwinkels auf bereits gesammelte Daten eben diese optimieren, um bessere Aussagen über den gerade zurückgelegten Weg treffen zu können.

Leider handelt es sich bei den genannten Möglichkeiten entweder um Daten die für sich betrachtet zu ungenau erscheinen, oder um komplexe Algorithmen, die viel Rechen- und Speicheraufwand erfordern. Deshalb sind unweigerlich bewährte Kombinationen obiger Verfahren am besten geeignet, um eine möglichst robuste Lokalisierung im Gebäude durchführen zu können.

1.3 Aufgabenstellung

Die Aufgabe des Testsystems besteht darin, möglichst genaue Positionsinformationen relativ zur Startposition und -laufrichtung zu liefern. Diese Daten müssen zuverlässiger, als die gegenwärtige Konfiguration der Sensorknoten sein, sodass eine Optimierung der Entfernungsmessung dieser stattfinden kann.

Die Daten werden so erhoben, dass eine visuelle Auswertung sowohl während der Ausführung des Tests als auch später stattfinden kann. Dabei muss erkennbar sein, zu welcher Zeit welche Positionsdaten von welchem System vorlagen.

Außerdem muss es möglich sein, eine mathematische Analyse der Abweichungen der Sensordaten im Vergleich zu den Daten unseres Systems zu erstellen. Diese Arbeit soll jedoch hierauf noch keinen Fokus legen, da zunächst die zu verwendenden Systeme evaluiert werden sollen.

Als Basis wird ein Roboter dienen, der durch das Gebäude fährt. Die Steuerung erfolgt zunächst per Tastatur, wobei alternative Steuerungsmöglichkeiten getestet werden sollen. In diesem Kontext sollte der Roboter möglichst autonom durch Flure navigieren und eine gefahrene Strecke bei Bedarf wiederholen können.

Damit die Daten zunächst visuell ausgewertet werden können, muss eine Karte der Teststrecke erstellt werden, die es theoretisch erlaubt, sowohl die Positionsdaten der Sensorknoten, als auch die unseres Testsystems anzuzeigen.

KAPITEL 2

Umsetzung

2.1 Konzept

lorem ipsum!

2.2 Soft- und Hardware

Im Folgenden werden die wichtigsten Konzepte näher erläutert.

2.2.1 Microsoft Kinect

lorem ipsum! Auflösung, Reichweite, Genauigkeit in Abhängigkeit zur Entfernung, generelle Funktionsweise

2.2.2 Robot Operating System

lorem ipsum!

Nodes, Topics, Msgs

Dies ist ein text [1, S. 12ff].

Dies ist ein text [1, S. 12ff].

Dies ist ein text [2].

Robot Operating System (ROS)

2.2.3 OpenNI Stack

lorem ipsum! Pointclouds

2.2.4 Turtle Bot

lorem ipsum!

2.3 Pathfinder

lorem ipsum!

2.4 NavStack

lorem ipsum!

KAPITEL 3

Analyse

3.1 Testlauf

lorem ipsum!

3.2 Genauigkeit

lorem ipsum!

3.3 Fazit

lorem ipsum!

KAPITEL 4

Ausblick

lorem ipsum!

KAPITEL 5

Anhang

lorem ipsum!

Literaturverzeichnis

- [1] Morgan Quigley, Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. Ros: an open-source robot operating system, 2009.
- [2] Heiko Will, Stefan Pfeiffer, Stephan Adler, Thomas Hillebrandt, and Jochen Schiller. Distance measurement in wireless sensor networks with low cost components, 2011.