

Differenzbasierte Repräsentation räumlicher Relationen zur probabilistischen Szenenerkennung mittels hierarchischen Constellation Models

Bachelorarbeit
von

Joshua Enrico Link

An der Fakultät für Informatik
Institut für Anthropomatik und Robotik
Lehrstuhl Prof. Dr.-Ing. R. Dillmann

Erstgutachter:	Prof. Dr.-Ing. R. Dillmann
Zweitgutachter:	???
Betreuender Mitarbeiter:	Dipl.-Inform. Pascal Meißner

Bearbeitungszeit: 11. Juni 2017 – 10. September 2017

Hiermit erkläre ich an Eides statt, dass ich die von mir vorgelegte Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen – die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Karlsruhe, den **(Datum)**

ToDo

Joshua Enrico Link

Inhaltsverzeichnis

Abbildungsverzeichnis	v
1 Einführung	1
2 Motivation und Problemstellung	2
2.1 Motivation	2
2.2 Fokus der Arbeit	3
3 Grundlagen	5
3.1 PSM (Probabilistic Scene Model)	5
3.1.1 Aufbau	6
3.1.2 Learner - Anlernen der Daten	6
3.1.3 Scene Model - Modellerzeugung	6
3.1.4 Inference - Szenenerkennung	6
3.2 Datengetriebene Entwicklung	6
3.2.1 Allgemein	6
3.2.2 Am Beispiel: PSM	6
4 Konzept	7
4.1 Ansatz	7
4.2 Erkennungsalgorithmus	7
5 Implementierung	8
5.1 Umbau PSM	8
5.1.1 Klassenaustausch	8
5.1.2 Ersatzfunktionen	8
5.1.3 Datenbankeinbindung	8
5.2 Differenzbasierter Erkennungsalgorithmus	8
6 Evaluation	10
6.1 Experiment 1: Frühstück	10
6.2 Experiment 2: Office	10
7 Zusammenfassung und Ausblick	11

Literaturverzeichnis

12

Abbildungsverzeichnis

2.1	Beispiel: Relative Position eines Objekts zu einem anderen	3
-----	--	---

1. Einführung

In der Robotik ist die Servicerobotik wohl der Forschungsbereich, welcher den größten Alltagsbezug für den Menschen hat, da er sich mit der Entwicklung und Weiterentwicklung von autonomen Robotern beschäftigt, welche dem Menschen im Alltag assistieren. Man findet mittlerweile Roboter im Privaten, die das Putzen, Staubsaugen oder Rasenmähen übernehmen, in der Industrie, bei Montage und Fertigung, sowie auch in der Medizin, als Pflegehilfe, Botengänger oder Assistent.

Allerdings müssen die Roboter ihre Umwelt für komplexere Aufgaben so präzise wie möglich wahrnehmen und verstehen. Sie könne Aufgaben übernehmen bei denen sie gezielt Objekte umfahren, suchen und auch aufnehmen und benutzen. Dieser Funktionsumfang kann mit dem Prinzip Programmieren durch Vormachen (PdV) ermöglicht werden, bei dem die Roboter Objekte und Tätigkeiten ihrer Umgebung kennen lernen, wieder erkennen und nachahmen können. So lässt sich die hohe Komplexität umgehen, die die manuelle Programmierung vieler Aufgaben mit sich bringen würde.

Um tatsächlich selbstständige Serviceroboter zu schaffen muss man aber noch zu einer Objekterkennung ein Kontextverständnis hinzufügen. Die Roboter müssen erkannte Objekte in einen Zusammenhang bringen, um die dadurch resultierenden Aufgaben zu verstehen. Zum Beispiel hat ein Teelöffel, welcher neben einer Tasse Tee liegt eine andere Aufgabe zu verrichten, als wenn er neben einem Becher Joghurt platziert ist. Nur am Kontext lässt sich dort entscheiden warum im einen Fall umgerührt und im anderen gelöffelt wird. Ebenso wäre ein Stück Butter verschieden zu verwenden, wenn es auf einem Frühstückstisch steht als wenn es mit anderen Zutaten neben einer Rührschüssel vorkommt.

Somit braucht man eine Szenenerkennung, welche zuverlässig die Objekte erkennen und ihren jeweiligen Kontext verstehen und einschätzen kann. Diese Erkennung ist nicht immer eindeutig, da der eben genannte Löffel ebenso zwischen einem Becher Joghurt und einer Tasse Tee liegen könnte, deshalb bietet es sich an mit Wahrscheinlichkeitsabschätzungen des vorliegenden Kontexts zu arbeiten.

2. Motivation und Problemstellung

2.1 Motivation

Wie schon in der Einführung erwähnt ist es elementar wichtig, dass man eine zuverlässige Szenenerkennung und ein gutes Kontextverständnis schafft um den Robotern die Möglichkeit zu bieten, sinnvoll mit ihrer Umwelt zu interagieren. Mit einer präzisen Wahrscheinlichkeitseinschätzung wie die momentane Umgebung beschaffen ist, lässt sich abschätzen welche Aufgaben es zu bewältigen und welche Probleme zu lösen gilt. Um dies zu gewährleisten muss man in das System möglichst viele Referenzdaten einspeisen, damit es jeden vorhandenen Kontext erkennen kann. Szenen werden zu diesem Zweck aufgebaut um der Erkennung als neue Szene vorgestellt. Jede Szene die die Szenenerkennung auf diese Weise lernt, hilft das Chaos der sie umgebenen Objekte mehr und mehr zu interpretieren und einzuordnen. Natürlich könnte man das System auch dynamisch mit jedem auftreten eines Objekts neue Kontexte beziehungsweise Szenen lernen lassen, das würde allerdings die Erkennung an Präzision einbüßen lassen.

Die Szenenerkennung nutzt also die Daten die sie zur Verfügung gestellt bekommt, bereits gelernte Szenen wiederzuerkennen. In dem bereits vorhandenen PSM(Probabilistic Scene Model)-System werden die Daten pro Szene zu einem Modell zusammengefasst, bei dem Auffällige Zusammenhänge berücksichtigt werden und scheinbar nicht miteinander in Verbindung stehende Objekte voneinander gelöst betrachtet werden. Zum Beispiel findet man eine Computermouse signifikant häufig vor dem Computerbildschirm und nie dahinter, allerdings kann dabei das räumliche Verhältnis der Maus zur Tastatur stark variieren. Die Maus wäre in diesem Beispiel manchmal direkt neben der Tastatur, manchmal dichter beim Bildschirm eben so oft weiter entfernt. In diesem Fall würde möglicherweise die Relation zwischen Maus und Tastatur wegfallen und nur jeweils das räumliche Verhältnis zum Bildschirm betrachtet werden. Dadurch gibt es Vorteile in der Laufzeit und möglicherweise auch eine signifikantere Erkennung, allerdings findet natürlich auch ein Informationsverlust statt, der zu Fehlern führen kann. In dieser Arbeit wird ein Ansatz getestet, der dichter an den erhaltenen Daten arbeitet.

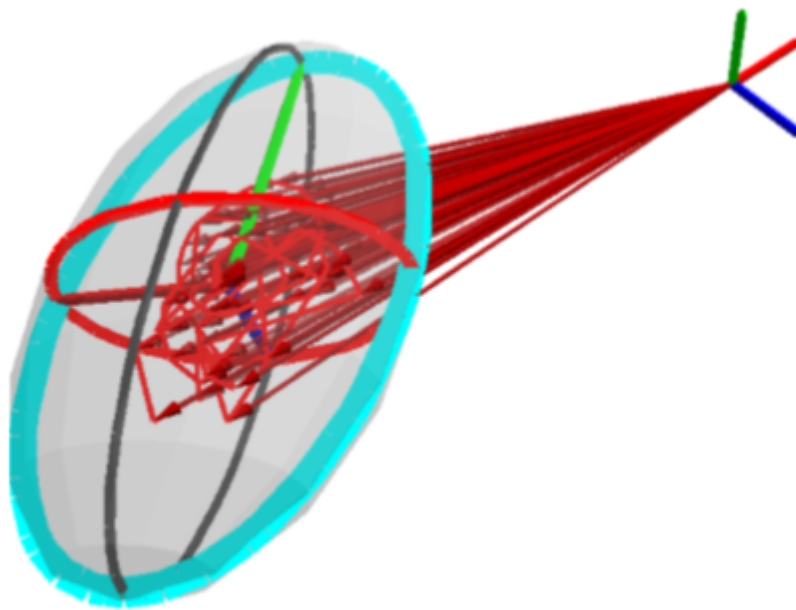


Abbildung 2.1: Beispiel: Relative Position eines Objekts zu einem anderen

In Abbildung 2.1 sieht man wie das Modell eine Relation zwischen zwei Objekten aufgrund der erhaltenen Daten erstellt. Die roten Pfeile beschreiben hier die jeweiligen Relationen, die aus den Daten berechnet wurden und die Ellipsen zeigen, welchen Bereich das Modell als Basis für die Wahrscheinlichkeitsabschätzung benutzt. [Geh14]

2.2 Fokus der Arbeit

Ziel der vorliegenden Arbeit ist es das PSM-System zu überarbeiten und einen neuen Modus der Szenenerkennung im PSM-System zu entwickeln. Dieser Modus soll alternativ zu den bereits vorhandenen Modi auswählbar sein und das System erweitern. Außerdem soll sowohl die Positionierung als auch die Rotation der erkannten Objekten mit den bekannten Szenen verglichen werden um eine Wahrscheinlichkeitsabschätzung auszugeben, welche die Wahrscheinlichkeit aller möglichen Szenen ausgibt sowie auch die Wahrscheinlichkeit, dass es sich um keine der Szenen handelt.

Um eine interaktive Szenenerkennung zu schaffen ist es sinnvoll das System datengetrieben zu programmieren. Datengetriebene Entwicklung zeichnet sich dadurch aus, dass sich der Programmfluss ändert aufgrund der Daten die das System während der Laufzeit als Eingabe erhält. Das PSM-System wurde bereits datengetrieben programmiert und in der zu dieser Arbeit gehörigen Implementierung wurde auch am datengetriebenen Entwicklungsmodell festgehalten.

Da die Erkennung innerhalb des PSM-Systems nutzbar sein soll sind Eingabe- und Aus-

gabeschnittstellen sowie die Visualisierung vorgeschrieben und sollen für gute Vergleichbarkeit denen des vorhandenen Systems entsprechen.

Die Arbeit ist wie folgend strukturiert. In 3 Grundlagen wird auf die Grundkenntnisse eingegangen die man braucht um den Rest der Arbeit gut zu verstehen. Vorallem wird in diesem Kapitel das bestehende PSM-System erklärt. In 4 Konzept wird das theoretische Konzept thematisiert, welches zum Algorithmus geführt hat, dass den neuen Modus vom alten System unterscheidet. Außerdem wird der Algorithmus selbst erläutert. In 5 Implementierung wird die Software beschrieben und dokumentiert die im Zuge dieser Arbeit entstanden ist sowie die Änderungen die am bestehenden PSM System vorgenommen wurden. 6 Evaluation beschreibt die durchgeführten Experimente und interpretiert sie. In 7 Zusammenfassung und Ausblick wird die Arbeit noch einmal zusammen gefasst und ein Ausblick darauf gegeben in welche Richtung man das bestehende System weiter entwickeln und auf welche Weise man möglicherweise die Szenenerkennung noch weiter verbessern kann. Am Schluss stehen die Quellen welche zur Recherche für diese Arbeit genutzt wurden.

3. Grundlagen

In diesem Kapitel wird das bestehende Probabilistic Scene Model - System vorgestellt und erklärt. Dabei wird auf das Grundprinzip, die Struktur und die einzelnen Komponenten eingegangen. Außerdem wird das Modell der datengetriebenen Entwicklung erklärt und an einem Beispiel verständlich gemacht.

3.1 PSM (Probabilistic Scene Model)

Bei der Entwicklung des Systems war das Ziel, ein System zur Erkennung von Szenen zu entwickeln. Dabei sollten viele Informationen umfasst werden. Dies umfasste die beteiligten Objekte bzw. deren Erscheinung, welche von den zur Erkennung eingesetzten Werkzeugen abhängig ist. Die Auftrittshäufigkeit der Objekte sollte ebenfalls mit einfließen. Der Hauptinformationsträger sind die Relationen der Objekte untereinander, welche in Form von relativen Objektlagen berücksichtigt wurden. Es wurde berücksichtigt, dass Relationen nicht statisch, sondern dynamisch sind. Da eine Szene auch unabhängig von ihrem Ort der Demonstration erkannt werden soll, musste die Lagebeschreibung invariant gegenüber Rotation und Translation sein.

Weiterhin sollte Robustheit gegenüber verschiedenen Störfaktoren bestehen. Fehlende Objekte sollen eine Erkennung der Szene erlauben, wenn auch mit entsprechend reduzierter Konfidenz. Da jedes Objekt fehlen kann darf es kein zentrales Referenzobjekt geben, von dem die Relationen zu den anderen Objekten der Szene ausgehen. Überzählige Objekte sollen sich nicht negativ auf das Erkennungsergebnis auswirken. Generell soll sich die Funktionsweise des entwickelten Systems im Rahmen dessen bewegen, was plausibel erscheint.

Es wird im folgenden erst auf die Struktur des Systems, dann auf die Eingabeverarbeitung, das innere Datenmodell und letztendlich auf die eigentliche Erkennung eingegangen. [Geh14]

3.1.1 Aufbau

Das System ist in mehrere Komponenten aufgeteilt die verschiedenen Aufgaben dienen. Für diese Arbeit wichtig sind vorallem die Komponenten Learner und Inference. Der Learner ist für das Anlernen der Daten und das Einspeichern von Szenen zuständig. Die Inference übernimmt die Erkennung der Szene indem sie in Echtzeit Daten empfängt und die Wahrscheinlichkeiten der Szenen ausgibt. Es gibt außerdem eine Visualizer-Komponente, die dafür zuständig ist, dass verschiedene programminterne Prozesse für den Nutzer sichtbar und verständlich gemacht werden.

3.1.2 Learner - Anlernen der Daten

3.1.3 Scene Model - Modellerzeugung

3.1.4 Inference - Szenenerkennung

3.2 Datengetriebene Entwicklung

3.2.1 Allgemein

3.2.2 Am Beispiel: PSM

Länge max. halb so lang wie Konzept + Implementierung

simpel beschreiben

konkret:

bestehendes System : PSM

Relevanz erklären?

Datengetriebene Entwicklung erklären

Viele Bilder benutzen, auch aus Joachims Arbeit

Auch aus Joachims Arbeit

[Geh14]

4. Konzept

4.1 Ansatz

4.2 Erkennungsalgorithmus

komplexer mathematischer formulieren
Vergleichsbasierte Erkennung erklären
Stochastische Richtigkeit beweisen

[DSS93]

5. Implementierung

Im Kapitel Implementierung wird alles beschrieben und erklärt, was am bestehenden PSM-Projekt verändert und hinzugefügt wurde. Außerdem werden ausgewählte hinzugefügte und veränderte Klassen sowie launch-Dateien dokumentiert, sodass das Kapitel das Verständnis und die Nutzung der Neuheiten im System vereinfacht. Nachdem der Umbauprozess beschrieben wird, bei dem eine Klasse komplett aus dem PSM-Projekt ausgetauscht wurde, widmet sich das Kapitel der Umsetzung des Algorithmuskonzepts und der Einbettung in das vorhandene System.

5.1 Umbau PSM

Da das Paket "pbd_msgs" nicht kostenlos zur Verfügung gestellt wird, mussten alle Vorkommen der Klassen aus diesem Paket zu alternativen Ersatzklassen geändert werden. Teilweise konnte man dies durch simple Ersetzung erreichen, allerdings gab es nicht für jede Klasse eine Ersatzklasse mit dem selben Funktionsumfang. In den Fällen, in denen Funktionen fehlten oder geringfügig anders funktionierten, konnte man den Umbau durch kleine Anpassungen erreichen oder musste eigene Funktionen schreiben, welche die nötigen Operationen verrichten konnten. Alle auf diese Weise programmierten Funktionen wurden hinreichend auf Gleichheit mit ihren Ursprungsfunktionen in ihrer Funktionsweise getestet, indem die Ergebnisse bei gleichen Eingangsparametern abgeglichen wurden. Außerdem habe ich eine Datenbankschnittstelle für das PSM-System hinzugefügt.

5.1.1 Klassenaustausch

5.1.2 Ersatzfunktionen

5.1.3 Datenbankeinbindung

5.2 Differenzbasierter Erkennungsalgorithmus

alle Klassen die umgebaut wurden
neuer differencebased modus

[DSS93]

6. Evaluation

6.1 Experiment 1: Frühstück

6.2 Experiment 2: Office

Viele Bilder, beschreiben Daten

Text interpretiert

Fazit am Ende

[DSS93]

7. Zusammenfassung und Ausblick

Zwei Sätze zu jedem größeren Kapitel

[DSS93]

Literaturverzeichnis

- [DSS93] Randall Davis, Howard Shrobe und Peter Szolovits: *What is a Knowledge Representation?* AI Magazine, 14(1):17–33, 1993. <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1029>.
- [Geh14] Joachim Gehring: *Probabilistische Szenenerkennung durch hierarchische Constellation Models über räumliche Relationen aus demonstrierten Objekttrajektorien*. Seiten 5–63, 2014. <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1029>.