Masterarbeit

Aufmerksamkeitsgetriebene Objektexploration für autonom lernende Roboter

im Studiengang **Informatik**
des Fachbereichs Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von **Oliver Erler** (4341785) am **25.08.2011**

Erstgutachter: Prof. Dr. Marc Toussaint
Zweitgutachter: Prof. Dr. Raúl Rojas

„*As it turned out, what is easy for people – recognizing a friend, reading, drinking from a coffee cup, folding a newspaper, preparing a meal – was extremely hard for machines, and what is often hard for people – things like logic, solving puzzles, and playing chess – is easy for computers.*“

Rolf Pfeifer und Christian Scheier in *Understanding Intelligence* [PS99]

Kurzfassung

Grundvoraussetzung, um zielgerichtete Handlungen im Alltag durchzuführen, ist es Objekte zu identifizieren und deren spezifische Eigenschaften und Funktionen zu erkennen. Während Menschen diese kognitive Fähigkeit und das Wissen im Kindesalter meist auf spielerische Art und Weise erlernen, werden bei heutigen Robotern die Objekteigenschaften oftmals über ein a priori Wissen fest vorgegeben. Deshalb wird in dieser Arbeit ein System vorgestellt, welches die Umwelt nach Objekten autonom exploriert und die Eigenschaft der Beweglichkeit analysiert. Als Vorbild dient hierfür die menschliche visuelle Wahrnehmung. Verschiedene elementare Merkmale der Szene werden zu einer Aufmerksamkeitskarte fusioniert. An den Orten mit hoher Aufmerksamkeit werden Objekthypothesen generiert. Die Verifizierung jener Hypothesen und die gegebenenfalls zugehörige Eigenschaftsanalyse findet anschließend über eine Interaktion mit Hilfe eines Roboterarms oder mit bereits verifizierten Objekten statt. Zur Evaluierung des Systems wurden mehrere Experimente mit unterschiedlichen Objekten durchgeführt. Die Ergebnisse zeigen, dass das entwickelte System oftmals in der Lage ist auch unter schwierigen Situationen alle Objekte zu finden und deren Beweglichkeit zu bestimmen.

Abstract

A basic prerequisite for goal-oriented actions in everyday life is to identify objects and to perceive their specific properties and functions. While humans learn this cognitive abilitiy and the knowledge mostly in a playful manner during childhood, the object properties by today's robots are often predetermined via a priori knowledge. Therefore in this thesis a system is presented which autonomously explores the environment for objects and analyses the property of movability. It is inspired by models of the human visual attention. Different elementary cues of the scene are combined into a saliency map. At each location that draws high attention an object hypothesis is generated. These hypotheses are verified either by interaction with a robot arm or by comparison with already verified objects. For the evaluation of the system various experiments with different objects have been carried out. The results show that the developed system often has the ability to find all objects and to determine their movability even in difficult situations.

Inhaltsverzeichnis

1 Einleitung

Roboter sind längst nicht mehr nur in Forschungseinrichtungen oder Industrieanlagen zu finden. Vielmehr haben sich Roboter auch im alltäglichen Lebensbereich stetig weiter verbreitet. Zumeist sind diese noch auf kleine spezielle Aufgaben beschränkt, wie das Staubsaugen oder das Rasenmähen. Längerfristiges Ziel ist es jedoch, dass die Roboter immer komplexere Abläufe selbständig erledigen können. Vor allem ist die Absicht das Service- und Pflegeroboter alltägliche und unliebsame Aufgaben im Alltag übernehmen und so den Menschen hilfreich zur Seite stehen. So sollen zukünftige Roboter Hol- und Bringdienste übernehmen, im Haushalt helfen oder kranke Patienten unterstützen. Dementsprechend steigen die notwendigen Anforderungen an die Flexibilität im Einsatzbereich, das autonome Verhalten und die Interaktion mit dem Menschen stetig an.

Dabei ist das autonome Lernen von Objekten und deren spezifischen Eigenschaften und Verwendung oftmals eine grundlegende Voraussetzung, um später die erwünschten praxisrelevanten Aufgaben bewältigen zu können. Beim Ein- und Ausräumen einer Geschirrspülmaschine erfordert es nicht nur eine sichere Handhabung des Geschirrs, sondern der Roboter muss zuerst einmal Wissen welche für ihn erstmal unbekannten Gegenstände auf dem Tisch zur Kategorie Geschirr gehören. Dann muss für jedes Einzelteil vom Geschirr deren spezifische Eigenschaften bekannt sein, wie Größe, Gewicht, Material um zum Beispiel den Gegenstand richtig greifen zu können. Es ist daher nicht verwunderlich das maschinelles Lernen ein sehr aktives Forschungsfeld in der Informatik ist. Klassische Ansätze, wie die Objekterkennung, basieren hauptsächlich allein auf der geschickten Auswertung von Bildern oder Laserscandaten, welche meist durch passive Beobachtung aufgenommen wurden. Aktuelle Methoden können bereits sehr hohe Erkennungsraten aufweisen, wenn die Objekte sich klar im Sichtfeld befinden und sich vom Hintergrund gut absetzen. Allerdings liegt hierin auch ein entscheidender Nachteil. Da sich Serviceroboter in einem oft unbekannten, unstrukturierten, menschlichen Umfeld bewegen, sind die Objekte nicht immer eindeutig sichtbar. Dazu treten auch Mehrdeutigkeiten auf, wenn z.B. zwei Objekte nebeneinander oder aufeinander liegen. Durch alleiniges Betrachten kann der Roboter nicht erfahren ob es sich um zwei einzelne oder um ein zusammenhängendes großes Objekt handelt. Auch weitere oftmals physikalische Eigenschaften des Objekts (z.B. Gewicht, Materialtyp, Manipulierbarkeit), lassen sich durch das bloße betrachten nicht feststellen. Um zu prüfen ob eine Flasche voll oder leer ist oder ob ein Tisch beweglich ist, muss mit dem zu untersuchenden Objekt interagiert werden.

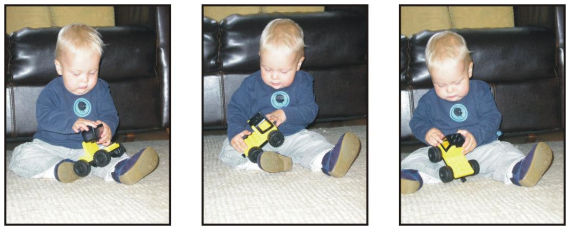


Abbildung 1.1: Ein Kind betrachtet ein Spielzeug aus verschiedenen Blickwinkeln. [KYDB07]

Während dies für die heutigen Roboter eine anspruchsvolle Aufgabe darstellt, ist es für uns Menschen ein selbstverständlicher (zumeist unbewusster) Prozess, Objekte zu kategorisieren und Eigenschaften zu identifizieren. Im Laufe unseres Lebens lernen wir Tag für Tag die Welt mit allen unseren fünf Sinnen immer mehr kennen. Wir *sehen* zum Beispiel wie sich die Bäume bei starkem Wind verbiegen, *hören* das Zwitschern der Vögel, *riechen* den Duft der Blumen, *schmecken* den Kuchen und *fühlen* die Hitze am Lagerfeuer. Mit Hilfe dieser Eindrücke und Erfahrungen lernen Menschen die Zusammenhänge unserer komplexen Welt zu verstehen. Besonders beim Spielen von Kleinkindern kann man gut beobachten, wie sie sich die Welt, auf meist spielerische Art und Weise, erschließen. Beim Aufstapeln von Bauklötzern zu hohen Türmen oder beim Erstellen von ersten Figuren aus einer Knetmasse lernen Kleinkinder nicht nur die Beschaffenheit der Materialien kennen, sondern auch mit welchen Formen von Bauklötzen sich am Besten ein hoher Turm bauen lässt. Dabei geht hauptsächlich die Neugierde von den Kindern selbst aus. Sie wollen die Umwelt für sich entdecken und unbekannte Dinge begreifen, siehe auch [Bei05]. Beim menschlichen Lernen entscheidet oft die visuelle Aufmerksamkeit welche Dinge das Kind als nächstes erforschen will. Dabei spielt sicherlich die Interaktion mit dem Objekt eine maßgebende Rolle. Sieht ein Kind ein unbekanntes aber auffälliges Spielzeug auf dem Teppich liegen, wird es mit hoher Wahrscheinlichkeit zum Spielzeug hin krabbeln und es anschupsen, aufheben oder vielleicht sogar werfen. Alle diese Aktionen sind aktive Handlungen und erlauben es dem Kind viele Merkmale und Eigenschaften vom Objekt selbständig herauszufinden. Unter anderem, ein Objekt aus mehreren Blickwinkeln zu betrachten (siehe Abbildung 1.1) oder Mehrdeutigkeiten aufzulösen, wie die Trennung des Objekts von einem strukturierten Hintergrund, siehe [KYDB07].

Die Motivation in dieser Arbeit besteht darin, das beschriebene menschliche Verhalten eines Kleinkindes für einen autonom lernenden Roboter nachzuahmen. Das Ziel ist es deshalb ein autonomes System zu entwickeln, welches die Umwelt auf Objekte hin aufmerksamkeitsgetrieben exploriert und deren Eigenschaften durch physische Interaktionen analysiert. Aufgrund der Komplexität werden allerdings zwei Einschränkungen vorgenommen. Erstens soll der Roboter stationär auf einem Tisch starre, physikalische Gegenstände explorieren und zweitens soll exemplarisch für alle interaktive Eigenschaftsanalysen von Objekten nur die Beweglichkeit betrachtet werden. Das ideale Zielszenario

des Systems sollte folgenden Ablauf aufweisen: Ein Roboter befindet sich vor einem Tisch, auf dem sich unterschiedliche, domänunabhängige Objekte befinden. Am Anfang hat der Roboter wenig Wissen über die Objekte, weder was, wo, wieviele und ob sich welche auf dem Tisch befinden. Mit Hilfe einer Kamera extrahiert er verschiedene visuelle Merkmale und erstellt daraus eine sogenannte Aufmerksamkeitskarte, die angibt wo sich potentielle Objekte befinden. An den Orten mit hohen Aufmerksamkeitswerten wird eine genauere Untersuchung[1] durchgeführt, indem eine erste Objekthypothese erstellt und mit dem Wissen vergangener Betrachtungen verglichen wird. Ist es bekannt, braucht die Stelle nicht weiter untersucht werden. Ist sie allerdings noch unbekannt, versucht der Roboter das potentielle Objekt zu verschieben, um zu erfahren ob es ein Objekt ist und ob es beweglich ist. Dies wiederholt der Roboter für alle Objekthypothesen bis er alle interessanten Bereiche der Aufmerksamkeitskarte einmal untersucht hat. Idealerweise hat er am Ende zu jedem Objekt auf dem Tisch eine Zuordnung ob das Objekt beweglich ist oder nicht.

Die Arbeit gliedert sich wie folgt: Im Kapitel „Grundlagen der visuellen Wahrnehmung“ wird zunächst eine Einführung in die menschliche visuelle Wahrnehmung gegeben. Dabei werden die neurobiologischen Verarbeitungen im Auge und Gehirn erläutert und einige darauf aufbauende kognitive psychophysikalische Effekte vorgestellt. Im dritten Kapitel „Verwandte Arbeiten“ wird ein Überblick über verschiedene aber verwandte kognitive Systeme vermittelt und die Vor- und Nachteile der verwendeten Verfahren betrachtet. Im vierten Kapitel „Systemkonfiguration und -umgebung“ wird die Systemumgebung und deren technische Eigenschaften aufgezeigt, sowie das grundlegende System konfiguriert. Im fünften Kapitel „Aufmerksamkeitsgetriebene Objektexploration“ werden die verwendeten Schritte des Gesamtsystems erklärt und im Einzelnen detailliert beschrieben. Dabei werden die Schritte unabhängig von der Implementierung erläutert und der Schwerpunkt auf die fachlichen und wissenschaftlichen Aspekte gelegt. Im sechsten Kapitel „Experimentelle Evaluierung“ werden einige ausgesuchte Experimente durchgeführt und die ermittelten Ergebnisse und Grenzen des Systems dargelegt. Im siebten und letzten Kapitel „Zusammenfassung und Ausblick“ werden die Resultate zusammengefasst und einige weiterführende Entwicklungsmöglichkeiten aufgezeigt.

[1] In der Analogie zum Kleinkind Beispiel, welche das auffällige Spielzeug genauer untersucht.

2 Grundlagen der visuellen Wahrnehmung

Wenn wir eine Analogie zwischen der Verarbeitung unserer visuellen menschlichen Wahrnehmung und einer digitalen Kamera schaffen müssten, könnte jemand auf folgende Zusammenhänge kommen: Beide nehmen die Reflexion von Lichtstrahlen an Oberflächen auf, die über eine Linse auf eine Retina beziehungsweise CCD-Sensor[1] projiziert wird, welche wiederum in elektrische Signale umgewandelt werden, um diese letztendlich in verschiedenen Gehirnarealen (oder im Rechner) zu verarbeiten.

Natürlich ist die menschliche Wahrnehmung wesentlich vielschichtiger als diese sehr vereinfachte Sichtweise. Zur Verdeutlichung dieses Sachverhaltes wird an dieser Stelle eine einfache Überlegung aus [SS11] dargelegt: Schaut man sich die menschliche Retina nämlich genauer an, so gibt es einen Bereich wo sich keine zum Sehen notwendigen Rezeptoren befinden (der blinde Fleck) und einen kleinen Bereich wo sich bestimmte Rezeptoren anhäufen (die Fovea). Dadurch erhalten wir am blinden Fleck keine Bildinformation und nur im Bereich der Fovea eine klare Abbildung der Umwelt. Daher müssten wir theoretisch immer einen schwarzen Fleck und nur einen kleinen scharfen Ausschnitt beim Sehen beobachten. Kommt jetzt noch mit in die Betrachtung, dass wir unsere Augen ständig hin und her bewegen, würde ein vollkommenes Chaos entstehen. Das dies nicht der Fall ist, *sehen* wir im wahrsten Sinne des Wortes.

Daher müssen sich in der menschlichen Wahrnehmung weitaus komplexere Prozesse abspielen, die in diesem Kapitel näher betrachtet werden. Es werden zuerst wichtige neurobiologische Grundlagen eingeführt und anschließend einige kognitive psychophysikalische Effekte beleuchtet. Diese dienen einerseits als Vorbild zum Entwurf des vorgestellten technischen Systems und andereseits bieten sie dem Leser ein besseres Verständnis in die Thematik.

[1] Ein CCD-Sensor (*Charge-coupled Device-Sensor*) ist ein häufig eingesetztes lichtempfindliches Bauelement bei digitalen Kameras, welche die Lichtstrahlen in elektronische Signale umwandelt.

2.1 Neurobiologische Verarbeitung

In den folgenden Unterkapiteln werden die wesentlichen Stationen, die für die visuelle Wahrnehmung zuständig sind, vorgestellt. Angefangen von den Lichtstrahlen die durch das Auge gehen, bis zu der neurobiologischen Verarbeitung im menschlichen Gehirn.

2.1.1 Anatomie des Auges

Auf den Weg zur Retina müssen die Lichtstrahlen zunächst einmal die Hornhaut und Linse eines jeden Auges passieren. Diese bündeln und brechen die ankommenden Lichtstrahlen so, dass es zu einer klaren Abbildung der Umwelt auf der Retina führt. Damit dies gelingt wird die kapselförmige Linse von Muskelsträngen bei nahen Objekten zusammen und bei entfernten Objekten auseinander gezogen, was eine Veränderung der Linsendicke bewirkt. Das dies eine wichtige Funktion ist, erleben wir wenn die Brechkraft einmal nachlässt. Wir sehen die Umwelt verschwommen und benötigen zur Korrektur eine Sehhilfe. Umschlossen vom Ganzen wird die Linse von der pigmentierten Iris. Die daraus entstehende Öffnung wird Pupille genannt und reguliert den Lichteinfall des Auges. Sie weitet sich bei Dunkelheit, um möglichst viel Restlicht aufzunehmen, und verengt sich wiederum bei hellem Licht. Die Blickrichtung wird durch sechs seitlich am Auge verlaufende Muskeln gesteuert, die sich stets auf das zu fixierende Objekt ausrichten [SS11].

2.1.2 Retina und Sehbahn

Die Retina ist eine dünne Schicht am Augenhintergrund. Hier befinden sich die lichtempfindlichen Rezeptoren, die dass auffallende Licht in elektrische Nervensignale umwandelt. Dabei teilen sich die Rezeptoren in zwei ergänzende Arten von Zapfen und Stäbchen (engl. cones and rods) auf. Ersteres übernimmt vorwiegend die Aufgabe des Farbsehens am Tage und letzteres das schwarz-weiß Sehen in der Nacht. Allerdings ist die Verteilung der Rezeptoren nicht gleichmäßig, siehe Abbildung 2.1. Vielmehr erkennt man, dass im Bereich des schärfsten Sehens, der Fovea, die Anzahl der Zapfen stark auf ca. 150 000 pro Quadratmillimeter anwächst aber gleichzeitig die Stäbchenanzahl sich bis auf Null absenkt. Im Bereich des blinden Fleckes existieren aus anatomischen Gründen weder Zapfen noch Stäbchen, weil sich dort die Austrittsstelle des Sehnverves befindet [Bac04] [HT00].

Nach der Umwandlung in elektrische Nervensignale werden die Informationen durch ein Netzwerk von bipolaren Zellen, an die innenliegenden Ganglienzellen weitergeleitet. Dabei beträgt das Verhältnis ca. 1:126 zwischen der Anzahl der Ganglienzellen gegenüber den Rezeptoren. Jede Ganglienzelle ist somit

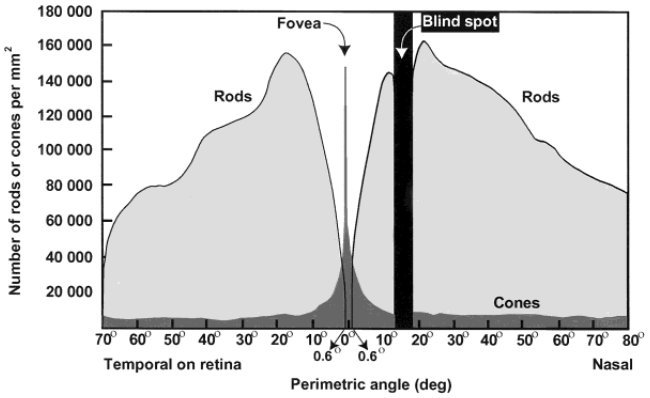


Abbildung 2.1: Die Verteilungsdichte der Rezeptoren auf der menschlichen Retina. [HT00]

mit einer oder mehreren Rezeptoren verbunden, was als rezeptives Feld bezeichnet wird. Allerdings stellt sich heraus, dass auch hier die Verteilung der Verbindungen nicht gleichmäßig ist. Die Ganglienzellen im Bereich der Fovea sind mit jedem einzelnen Zapfen direkt verbunden (was zu der guten Sehschärfe der Retina beiträgt), im Gegensatz zu den restlichen Ganglienzellen, welche ihre Informationen aus einer großen Fläche von Rezeptoren erhalten. Insgesamt führt das dazu, dass nur ein Teil der Informationen an das Gehirn weitergeleitet wird und es so zu einer deutlichen Signalreduktion kommt. Eine weitere wichtige Eigenschaft der Ganglienzellen ist es, den Kontrast der Umwelt zu verstärken, damit zum Beispiel Kanten deutlicher werden. Dazu ist jedes rezeptive Feld in ein kreisförmiges Innenzentrum und einen Außenring aufgeteilt. Durch die Art und Weise wie die beiden Bereiche miteinander verschaltet sind, spricht man von *On-Center* Neuronen oder von *Off-Center* Neuronen. *On-Center* Neuronen haben die Eigenschaft, dass sie ein erregendes Zentrum und einen hemmenden Außenring besitzen. Treffen zum Beispiel Lichtstrahlen ausschlichlich die Rezeptoren, die mit dem Zentrum eines *On-Center* Neurons verbunden sind, steigt die Feuerungsrate des Neurons erheblich an, siehe Abbildung 2.2 (mittlerer Kreis). Allerdings wird zusätzlich der Außenring erregt, hemmt er die Feuerungsrate und das Neuron sendet entsprechend weniger Signale an das Gehirn, siehe Abbildung 2.2 (zweiter Kreis von rechts). Das *Off-Center* Neuron hat genau die konträren Eigenschaften, das heißt das Zentrum is hemmend und der Außenring ist erregend [Die08].

Weiterhin existieren spezielle Ganglienzellen für die Farbwahrnehmung. Diese sind vernetzt mit dem zum Farbsehen zuständigen Zapfen. Entsprechend der Sensitivität der verschiedenen Wellenlängen des Lichtes werden die Zapfen in S-Zapfen (engl. Short-wave receptor), M-Zapfen (engl. Medium-wave receptor) und L-Zapfen (engl. Long-wave receptor) eingeteilt. Kongruent zu dem Farbspektrum werden diese auch manchmal als Blau-, Grün- und Rot-Zapfen bezeichnet. Die Ganglienzelle summiert und differenziert die Zapfentypen in drei verschiedene Kanäle auf, dass für eine Dekorrelation und Verstärkung der

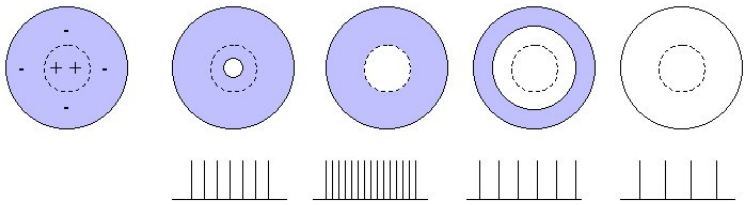


Abbildung 2.2: Unterschiedliche Beleuchtung eines rezeptiven Feldes und die dazugehörigen Feuerungsraten eines *On-Center* Neurons. [Die08]

Komplementärfarben sorgt. Der informationsreichste Kanal ist der Luminanz-Kanal, der die drei Farben summiert. Ein weiterer Kanal ist der Rot-Grün Kanal der eine Differenz zwischen rot und grün bildet. Der letzte und mit dem geringsten Informationsgehalt ist der Blau-Gelb Kanal, der zuerst die Summe zwischen rot und grün bildet und das Ergebnis mit blau differenziert [Bac04].

In jedem Auge entsteht durch die Bündelung der Nervenfasern aller Ganglienzellen die sogenannte Sehbahn. Die Nervenfasern verlassen über den blinden Fleck das jeweilige Auge und kreuzen sich einmal auf dem Weg zum visuellen Kortex. Das Ergebnis ist, dass das rechte Sehfeld (*nicht Auge!*) von der linken Gehirnhälfte und das linke Sehfeld von der rechten Gehirnhälfte weiter verarbeitet wird.

2.1.3 Visueller Kortex

Die von den beiden Augen ankommende Sehbahn gelangt zunächst zur primären Kortexschicht. Der visuelle Kortex ist grob in vier Neuronenschichten (V1-V4) eingeteilt. Die V1-Neuronen verarbeiten die Reize in gewisser Weise, wie die *On/Off-Center* Neuronen in der Retina. Allerdings sind die rezeptiven Felder nicht mehr kreisförmig, sondern haben eine längliche Form. Durch die veränderte Form sind sie in der Lage auf Linien oder Kanten zu reagieren. Für jede mögliche Orientierung und Ausdehnung einer Linie, existieren im visuelle Kortex eine Vielzahl passender V1-Neuronen mit unterschiedlichen Längen, Größen und Orientierungen. Da die Fovea vom Gehirn am detailiertesten ausgewertet wird, beansprucht sie auch am meisten Platz in der primären Sehrinde. Weiterhin wurde festgestellt, dass sie einer topographischen Anordnung unterliegt, in der benachbarte Regionen der Retina auch in der primären Sehrinde benachbart bleiben. Zum Beispiel liegen die Neuronenpaare für das linke bzw. rechte Auge immer abwechselnd nebeneinander. Dabei sind die geordneten Neuronenpaare für das Stereosehen verantwortlich. Erst wenn beide Neuronen gleichzeitig einen Reiz bekommen, senden sie spezielle Nervensignale aus. In den höheren Schichten sind die V2-Neuronen für die Lokalisierung der Linien im Raum zuständig, die V3-Neuronen analysieren hauptsächlich die Bewegung und die V4-Neuronen

werten die Farben aus. Zusätzlich gibt es noch sehr speziallisierte gesonderte Neuronen, die z.B. für die Gesichtserkennung zuständig sind [HH09].

Die Auswertung passiert völlig getrennt und parallelisiert in Bruchteilen einer Sekunden voneinander. Verfolgt man ausgehend von der V1-Schicht den Weg durch die vier beschriebenden Neuronenschichten erkennt man zwei Pfade, den sogenanten dorsalen Pfad und ventralen Pfad (auch häufig in der Literatur als „Wo“ und „Was“-Pfad bezeichnet). Die Informationen die im dorsalen Pfad entlang laufen, sind assoziiert mit der räumlichen Lokalisierung und Bewegung, vereinfacht ausgedrückt *wo* sich das Objekt befindet. Der ventrale Pfad hingegen ist für das Erkennen der Form und die Art des Objektes zuständig, dementsprechend sagt es aus um *was* für ein Objekt es sich handelt. Dazu bezieht es auch das Langzeitgedächtnis mit ein, um zum Beispiel teilverdeckte Formen wiederzuerkennen [And07].

2.2 Kognitive psychophysikalische Effekte

In den nachfolgenden Unterkapiteln werden einige kognitive psychophysikalische Effekte, die auf den neurobiologischen Erkenntnissen im letzten Kapitel beruhen, vorgestellt.

2.2.1 Visuelle Aufmerksamkeit

Mit seinen fünf Sinnen ist der Mensch in jedem Augenblick mit einer Flut von Information konfrontiert. Damit wir dieser enormen Informationsflut nicht hilflos ausgesetzt sind, hat die Evolution Mechanismen entwickelt, Wichtiges von Unwichtigem zu trennen. Fahren wir zum Beispiel mit einem Fahrzeug im dichten Stadtverkehr, nehmen wir nicht jede einzelne Information bewusst wahr, weil unser Gehirn nur eine begrenzte Kapazität hat. Würden wir im Beispiel jede Haarfarbe der vorbeilaufenden Menschen beachten oder den Windzug der Klimaanlage auf unserer Haut bewusst wahrnehmen, hätte das Gehirn keine Ressourcen mehr frei für die eigentliche Aufgabe, nämlich die Steuerung des Fahrzeugs. Dieses Problem umgeht der Mensch, indem er die Umweltreize nur zu einem Teil verarbeitet, siehe [Bac04].

Der Mechanismus beim Menschen der die visuelle Selektion durchführt, ist die visuelle Aufmerksamkeit. Eine weitverbreitete und anerkannte Theorie in diesem Bereich ist die der gesteuerten Suche (engl. guided search theory) von Jeremy M. Wolfe [Wol94]. Dessen Grundannahme ist es, dass jeder visuelle Reiz aus Kombinationen von grundlegenden, seperaten Merkmalen (z.B. blau, lang, vertikal) zusammengesetzt werden kann. Die dann in einzelne Dimensionen (z.B. Farbe, Orientierung, Bewegung) kategorisiert werden. Repräsentiert werden die Dimensionen durch sogenannte Merkmalskarten, worin die Erregungsorte und -stärke der Reize im Sehfeld verzeichnet werden. In der Theorie von Wolfe setzt sich die Erregungsstärke von einem *bottom-up*- und *top-down*-Prozess zusammen. Die Werte

des *bottom-up*-Prozesses hängen allein von der Salienz[2] innerhalb einer Dimension ab. Ein schwarzes Schaf in einer Herde voller weißer Schafe hätte zum Beispiel im Merkmal „schwarz“ der Dimension Farbe eine hohe Salienz aber unter einer Herde voller schwarzer Schafe eine niedrige Salienz. Im *top-down*-Prozess hingegen fließt ein a priori Wissen über das Zielreizmerkmal in die Bestimmung der Salienzhöhe gewichtet mit ein. Zum Beispiel will man einen blauen Legostein in einer Schachtel voller bunter Legosteine finden, wäre ein Zielreizmerkmal „blau“ der Dimension Farbe und ein weiteres Zielreizmerkmal „eckig“ der Dimension Form. Dadurch bekommen alle blauen und eckigen Legosteine in der Schachtel eine höhere gewichtete Salienz, als zum Beispiel grüne, eckige Legosteine oder gelbe, runde Legosteine. Beide Prozesse werden dann im nächsten Schritt zu einer gemeinsamen sogenannten Aufmerksamkeitskarte aufsummiert. Gemäß der Theorie richtet sich die menschliche Aufmerksamkeit dann zu dem Ort, der die höchste Salienz besitzt.

Grundlage dieser Theorie sind die sogenannten visuellen Wahrnehmungstests. Hierbei müssen Versuchspersonen einen gegebenen Zielreiz unter mehereren Distraktoren (Ablenkreizen) suchen. Zwei Messmethoden werden üblicherweise eingesetzt. Bei der ersten Methode werden der Versuchsperson bestimmte Muster kurzzeitig vorgelegt, in denen manchmal der gesuchte Zielreiz enthalten war oder komplett fehlte. Anschließend wird die Versuchsperson befragt, ob ein Zielreiz im Muster vorhanden war oder nicht. Anhand der korrekt gegebenen Antworten wird dann ein Prozentsatz ermittelt. Bei der zweiten Methode hingegen wird die Reaktionszeit gemessen, welche die Versuchsperson benötigt, um den gegebenen Zielreiz zu lokalisieren. Zudem werden bei beiden Methoden die Anzahl der Distraktoren und deren Dimensionen variiert [HKMS11].

In den Versuchen konnte festgestellt werden, dass sich bei steigender Distraktoranzahl die Reaktionszeit kaum verändert, wenn sich der Zielreiz nur aus einem einfachen Merkmal zu den umgebenden Distraktoren unterschied, siehe Abbildung 2.3 (linke Seite). Daher liegt die Vermutung nahe, dass bei einer einfachen Merkmalssuche (engl. simple feature search), der Suchprozess parallel verläuft. Auch ist es irrelevant, ob ein Zielreiz vorgegeben wird oder nicht, der Reiz „springt“ geradezu ins Auge und wird deshalb auch als *pop-out* Effekt bezeichnet. Dagegen steigt die Reaktionszeit linear mit der Anzahl der Distraktoren an, wenn die Variabilität der Distraktordimension erhöht wird, siehe Abbildung 2.3 (rechte Seite). Hieraus folgt, dass bei komplexeren Merkmalskombinationen eine sogenannte Merkmalskonjunktionssuche (engl. conjunction feature search) durchgeführt wird, d.h. das einzelne Distraktoren sukzessiv mitbetrachtet werden. Somit kommt man zur These, dass die Suche sequentiell verläuft. In den Experimenten konnte weiterhin herausgefunden werden, dass die Reaktionszeit sich durchschnittlich verdoppelte, wenn kein Zielreiz im Muster vorhanden war.

[2] Salienz ist ein Maß für die Andersartigkeit bezüglich seiner Umgebung.

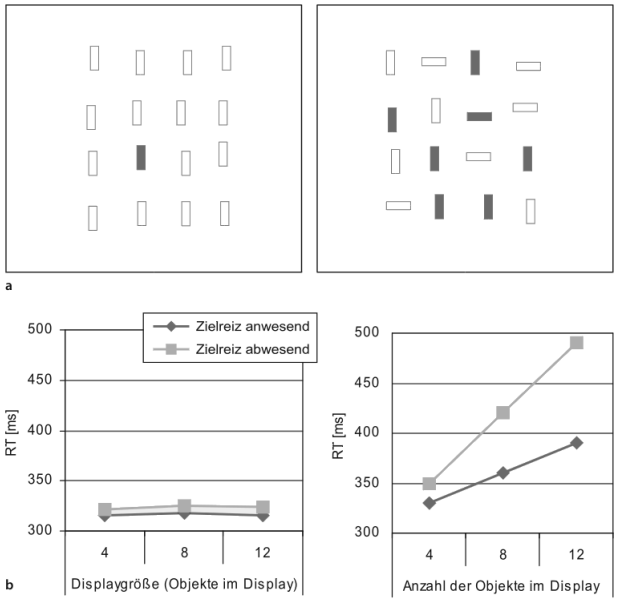


Abbildung 2.3: Visueller Wahrnehmungstest mit (a) unterschiedlicher Variabilität der Distraktordimension und (b) den gemessenen Reaktionszeiten bei unterschiedlicher Distraktoranzahl. [HKMS11]

2.2.2 Gestaltprinzipien

Die menschliche Wahrnehmung beruht nicht allein auf dem was wir mit unserem Auge sehen. Vielmehr bilden wir ständig Hypothesen über unsere Umwelt. Deutlich wird dies wenn man sich die Abbildung 2.4 anschaut. Es erschließt sich für uns sofort, dass eine gewisse Struktur in den Bildern erkennbar ist. Bereits um 1912 erforschten Psychologen dieses Phänomen und stellten daraufhin gewisse Gesetzmäßigkeiten auf, die heutzutage unter dem Namen Gestaltprinzipien bekannt sind. Das wichtigste Prinzip ist das Prinzip der Prägnanz. Es besagt, das Menschen jedes Muster auf die Art und Weise wahrnehmen, dass die Struktur eines Musters möglichst einfach interpretierbar ist. Zum Beispiel sehen wir augenscheinlich in Abbildung 2.4d zwei in sich überlappende Kreise, obwohl der verdeckte Kreis nicht unbedingt die Form eines Kreises haben muss. Doch die Hypothese von zwei Kreise ist für uns am leichtesten nachvollziehbar. Auch kann die Zeichnung mit dem Prinzip der Geschlossenheit erklärt werden, da wir dazu neigen unvollständige Formen zu komplettieren.

In Abbildung 2.4a wird das Prinzip der Nähe dargestellt. Instinktiv gruppieren wir die acht einzelnen Linien in vier Paare, weil diese etwas näher beieinander liegen. Hingegen ist in Abbildung 2.4b das Prinzip der Ähnlichkeit veranschaulicht. Weisen Elemente ähnliche Merkmale auf werden diese

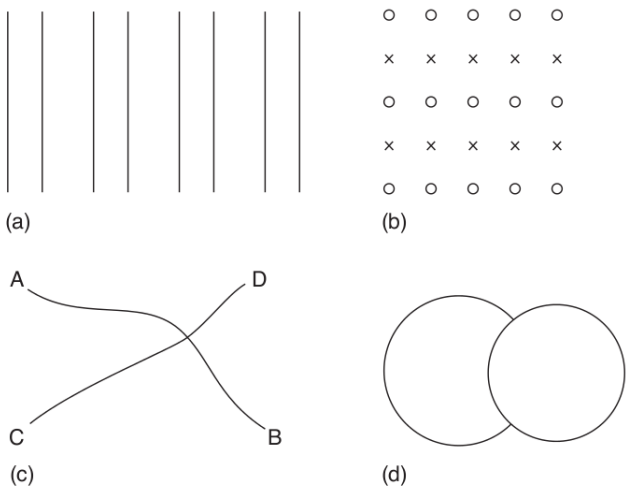


Abbildung 2.4: Vier exemplarische Gestaltprinzipien: (a) das Prinzip der Nähe, (b) das Prinzip der Ähnlichkeit, (c) das Prinzip der Kontinuität und (d) das Prinzip der Geschlossenheit oder Prägnanz. [And07]

zusammengehörig erlebt, als Elemente die untereinander verschiedene Merkmale besitzen. Darum erscheinen uns die Kreise und Kreuze als alternierende Zeilen. Ein nächstes Prinzip ist aus Abbildung 2.4c ersichtlich. Wenn wir wieder die Abbildung betrachten, entdecken wir eine Linie die von A nach B und eine zweite Linie die von C nach D verläuft. Das Prinzip, das dahintersteckt ist die Kontinuität, d.h. wir setzen Linien immer auf die Weise fort, dass sie einen glatten Verlauf aufweisen. Eine Linie die zum Beispiel von A nach C geht hat ein abruptes Abknicken zur Folge, was uns unnatürlich erscheint, obwohl nichts dagegen spricht [And07].

Das eine Hypothese uns auch in die Irre führen kann, bemerken wir immer wieder anhand von optischen Täuschungen. Zum Beispiel erscheint uns der abendliche Mond in Horizontnähe viel größer, als wenn dieser hoch am Himmel steht. Ein möglicher Grund (aber nicht die alleinige Erklärung) für diese optische Täuschung ist das Prinzip der Vergleichsobjekte. Dieses besagt, dass die Wahrnehmung der Größe eines Objektes bezüglich zu der Objektgröße der Umgebung immer genau entgegengesetzt erscheint. Im Falle des horizontnahen Mondes sind es z.B. die kleineren Bäume, Häuser oder Berge mit dem verglichen wird und wodurch ein falscher Eindruck der Mondgröße entsteht [SS11].

Die Gestaltprinzipien scheinen für die menschliche Wahrnehmung eine wichtige Rolle einzunehmen, da diese anscheinend häufig unvollständige Informationen mittels Hypothesen ergänzen. Woher und wie diese Fähigkeit genau funktioniert ist allerdings noch nicht eindeutig erschlossen.

2.3 Zusammenfassung und Diskussion

Wie in diesem Kapitel ersichtlich wurde, ist die menschliche visuelle Wahrnehmung ein komplexer, vielschichtiger Prozess. Erste wichtige Vorverarbeitungsschritte laufen bereits im Auge ab. Neben der adäquaten Umwandlung der Lichtstrahlen in elektrische Nervensignale, hat zum Beispiel die Retina weitere wichtige Funktionen zu erfüllen, wie die Kontrastverstärkung und die Signalreduzierung. Die von der Retina ans Gehirn weitergeleiteten Signale werden im nächsten Schritt zunächst in viele kleinere, unabhängige Merkmale zerlegt und in topographischer Art und Weise angeordnet. Die höheren Neuronenschichten hingegen, so scheint es, setzen die zerlegten Merkmale wieder zu komplexeren Formen zusammen, welche dann zu einem kohärenten Gesamteindruck der Umwelt führt. Wie allerdings diese Zusammensetzung genau funktioniert ist noch weitgehend ungeklärt und wird in der Literatur als Bindungsproblem (engl. binding problem) bezeichnet.

Das menschliche Gehirn kann die ankommenden visuellen Reize jedoch nicht gleichzeitig verarbeiten, da die Kapazität des Gehirns begrenzt ist. Es wertet stattdessen immer nur einen kleinen Teilauschnitt unserer Umwelt aus. Daraus folgt, dass wir nur die Dinge bewusst wahrnehmen, der wir auch unsere Aufmerksamkeit zuwenden. Betrachtet man zusätzlich, dass wir ständig unsere Umwelt zerlegen und wieder zusammenbauen, wird die Wirklichkeit im Prinzip erst als ein Konstrukt in unserem Gehirn erschaffen oder wie Heinz von Foerster in [vF07] zugespitzt formulierte:

> „*Die Umwelt, so wie wir sie wahrnehmen, ist unsere Erfindung.*“

Das in der visuellen Wahrnehmung noch mehr stattfindet als die reine optische Auswertung, belegen die Gestaltprinzipien. Sie stellen zahlreiche Gesetzmäßigkeiten auf, mit welchen Sichtweisen der Mensch die Umwelt wahrnimmt. Obwohl die Gestaltprinzipien noch nicht eindeutig erschlossen sind, geben sie uns doch einen guten Hinweis darauf, dass ständig Hypothesen der Umwelt mit in den Wahrnehmungsprozess einbezogen werden.

3 Verwandte Arbeiten

In diesem Kapitel wird ein Überblick über verwandte Arbeiten im Bereich der kognitiven Systeme gegeben. Der Schwerpunkt der Auswahl liegt bei Systemen, die dem menschlichen Lernen zu Grunde liegt. Die folgende Auswahl besitzt allerdings keinen Anspruch auf Vollständigkeit. Für tiefergehende Details der einzelnen Arbeiten wird daher auf die jeweiligen Literaturvermerke verwiesen.

3.1 Kognitive Systeme

Das Bestreben ein intelligentes System zu erschaffen, dass die kognitiven Fähigkeiten des menschlichen Vorbilds nachahmt, ist eines der anspruchsvollsten Herausforderungen in der künstlichen Intelligenz. Die Forschung in diesem Bereich ist höchst interdisziplinär. Sie weist nicht nur große Schnittmengen zu der Robotik auf, sondern ist eng mit den Kognitionswissenschaften, Neurowissenschaften, Wahrnehmungs- und Entwicklungpsychologie verwoben, siehe [AHK+09].

In den Anfängen der künstlichen Intelligenz bestand die vorwiegende Sichtweise, dass Intelligenz allein auf geschickte Anwendung von Algorithmen beruht, siehe [PI04]. Es entstanden Systeme wie der Schachcomputer *Deep Blue* von der Firma IBM oder das Expertensystem *ELIZA* von Joseph Weizenbaum. Obwohl diese Systeme wichtige Meilensteine darstellen, brachten die reinen algorithmischen Ansätze keine „Intelligenz“ im eigentlichen Sinne hervor. Eine neu entstandene Theorie ist daher, dass künstliche Intelligenz einen physikalischen Körper benötigt (engl. embodied artificial intelligence). Die Idee dahinter ist nach Yasuo Kuniyoshi et al. [KYS+07] zitiert nach [AHK+09] folgende:

> „*The agent's physical body specifies the constraints on the interaction between the agent and its environment that generate the rich contents of its process or consequences. It also gives the meaningful structure to the interaction with environment, and is the physical infrastructure to form the cognition and action.*“

Diese Sichtweise kommt somit dem Grundkonzept des menschlichen Lernens näher. Zugleich entspricht sie der in der Einleitung erwähnten Zielsetzung, dass die Umgebung wie ein Kleinkind exploriert werden soll. Um das Thema zu gliedern, werden die verschiedenen kognitiven Systeme in Anlehnung

an [LMPS03] in sozial interagierend, autonom interagierend und in sensormotorisch selbst beziehende Systeme unterteilt.

3.1.1 Sozial interagierend

Sozial interagierende Systeme lernen ihr Verhalten oder akquirieren ihr Wissen hauptsächlich aus den Interaktionen mit dem Menschen. Dies kann durch Imitation, sprachliche Kommunikation oder Beobachtung passieren [LMPS03].

In der Dissertation von Frank Lömker [Lö04] wurde ein komplettes System entwickelt, welches die Benennungen von physikalischen Objekten mit Hilfe von verbaler Kommunikation und Gestenhandlungen erlernt. Das Hauptszenario ist, dass ein Benutzer dem System innerhalb einer Szene dialoggestützt auf Objekte sprachlich und eventuell zusätzlich durch Zeigegesten referenziert. Anschließend versucht das System die Objekte zu lokalisieren oder auch zu identifizieren. Zur Bestimmung wird ein kombiniertes, multimodales Verfahren eingesetzt. Dazu wird anfangs die sprachliche Anfrage nach bestimmten Schlüsselwörtern untersucht und mit einer Objektdatenbank verglichen. Ist das Objekt dem System bekannt wird anhand, der in der Datenbank abgespeicherten Farb- und Strukturmerkmale, das referenzierte Objekt im Bild gesucht. Dabei bevorzugt die Suche das Umfeld auf das eine eventuelle ausgeführte Zeigegeste deutet. Konnte die Position des Objektes nicht ermittelt werden oder ist es dem System unbekannt, wird das Objekt mit Hilfe des Benutzers erlernt. Dazu fordert es dem Benutzer auf das referenzierte Objekt aus der Szene zu entfernen. Anhand der Differenzbildung der Szene vor und nach der Entnahme wird die Position des unbekannten Objektes bestimmt. Anschließend werden visuelle Merkmale des Objektes extrahiert und mit den sprachlichen Schlüsselwörtern des Benutzers verknüpft. Die so ermittelten Informationen werden in die Objektdatenbank eingepflegt, die in der nächsten Anfrage wiederum zur Verfügung stehen.

In [CDLI09] werden die möglichen Greifkonfigurationen eines Objektes mit Hilfe eines Menschens erlernt. Dazu trägt der Benutzer für jede Hand ein Datenhandschuh, die jeweils mit visuellen Markern und taktilen Sensoren ausgestattet sind. Anschließend nimmt er mit einer oder beiden Händen ein Objekt auf und betrachtet das Objekt aus verschiedenen Seiten. Dabei werden die Marker von fünf im Raum angebrachten Kameras verfolgt und deren 3D-Positionen kontinuierlich bestimmt. Gleichzeitig werden die taktilen Drucksensoren, die sich an den Fingerspitzen im Datenhandschuh befinden, aufgezeichnet, siehe Abbildung 3.1. Aufgrund der Tatsache, dass dem Benutzer erlaubt ist das Objekt frei in der Hand zu betrachten, werden die Markerpositionen über eine lineare Transformation zwischen den einzelnen Greifkonfigurationen ausgeglichen. Die entstehende Punktwolke wird im nächsten Schritt einer Clusteranalyse unterzogen. Diese unterteilt die Punktewolke in Regionen, in denen eine

hohe Anzahl an Greifkonfiguration zustande kam. Anschließend werden für jede Region die besten Greifkonfigurationen (Annäherungsvektor, Greifmittelpunkt, Orientierungvektor etc.) ermittelt und in mehreren sogenannten „*grasp oriented bounding box*“ zusammengetragen.

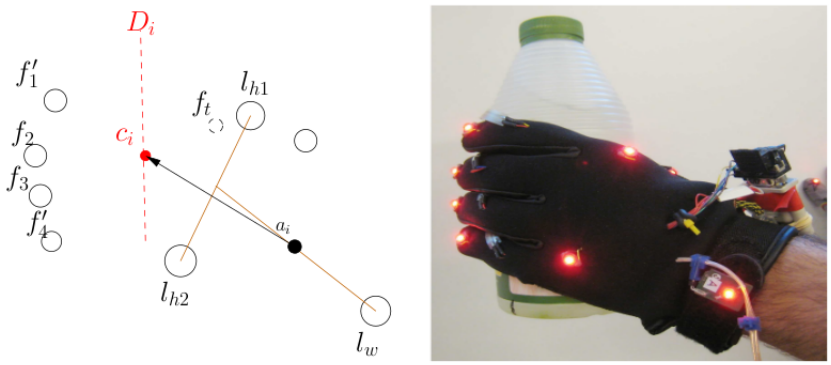


Abbildung 3.1: Schematische und die dazugehörige reale Darstellung einer Greifkonfiguration mit Handrichtung D_i, Handmittelpunkt c_i, Kontaktpunkte der Finger f_i und Anfangspunkt a_i des Annäherungsvektors. [CDLI09]

3.1.2 Autonom interagierend

Nicht-sozial interagierende Systeme arbeiten autonom und beziehen ihr Wissen aus der Umwelt, durch unterschiedliche Sensoren und/oder Aktoren, ohne die Interaktion mit Menschen oder anderen Robotern. Typische Aufgaben sind die Manipulation, Navigation oder Exploration der Umwelt [LMPS03].

In der Arbeit von Gratal et al. [GBBK10] wird zum Beispiel die Umwelt autonom exploriert. Dazu wird eine Szenen Repräsentation aufgebaut, die für das Greifen von bekannten Objekten geeignet ist. Das System besteht aus einen Roboterkopf mit zwei Kamerapaaren und einem Roboterarm mit angebauter Roboterhand. Dabei hat das eine Kamerapaar die Aufgabe, einen groben Überblick über die Szene zu erhalten und das zweite Kamerapaar übernimmt die Funktion der menschlichen Fovea. Der erste Schritt um eine Szene aufzubauen, besteht darin eine Aufmerksamkeitskarte zu erstellen. Dabei richtet sich die Fovea Kamera auf den Ort aus, welche die höchste Salienz besitzt. Anschließend wird aus den aufgenommenen Bildern der Stereokamera eine Tiefenkarte über Stereomatching errechnet. Die Segmentierung der ersten Objekthypothese erfolgt mittels der Farb- und Tiefeninformationen in einen zweistufigen iterativen Prozess. Der Schwerpunkt der Objekthypothese wird ermittelt und die Fovea Kamera wird dahin neu ausgerichtet. Anschließend wird in einer Datenbank die Objekthypothese

verifiziert. Anhand der gespeicherten Informationen über die Objektdimension wird die Orientierung des Objektes auf dem Tisch errechnet. Der erforderliche Greifpunkt für ein *top-grasp* erschließt sich aus den gespeicherten Informationen und der ermittelten Position. Der nachfolgende Greifprozess erfolgt über *visual servoing* mit einer am Handgelenk angebrachten Leuchtdiode.

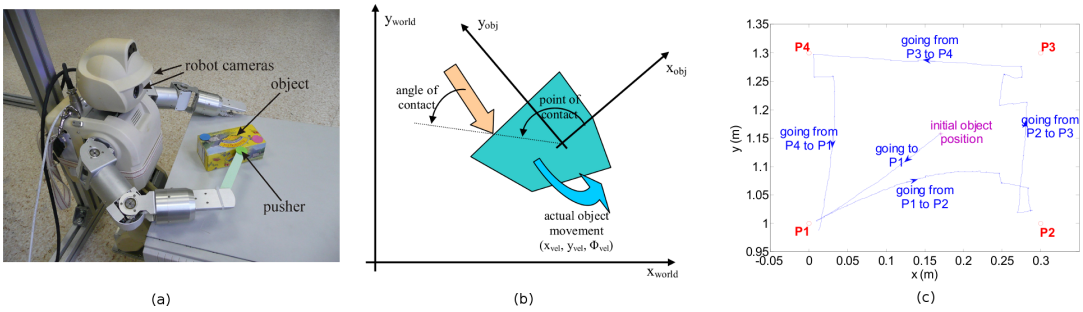


Abbildung 3.2: (a) Reale und (b) schematische Darstellung einer *poke*-Aktivität. (c) Objekttrajektorie (blau) während einer gezielten Punkt-zu-Punkt Bewegung zu den vier Eckpunkten (rot), geändert aus [DOK08].

In [DOK08] wird erlernt, wie ein unbekanntes aber planares Objekt zielgerichtet auf eine bestimmte Position verschoben werden kann. Als Experimentierplattform dient ein humanoider Roboter, siehe Abbildung 3.2a. Wie in der Abbildung zu erkennen ist, hat er in der rechten Hand ein *poke*-Werkzeug, mit dem das Schubsen erleichtert wird. Zusätzlich sind am Ende des *poke*-Werkzeugs und am Objekt Markierungen angebracht, die die visuelle Erfassung vereinfachen. Im Lernmodus schubst der Roboter vollkommen willkürlich unter verschiedenen Winkeln und Seiten das Objekt mehrmals an und beobachtet die Reaktion des Objektes, siehe Abbildung 3.2b. Aus den gesammelten Daten wird ein Teil genutzt, um ein zweischichtiges Neuronales Netzes zu trainieren, wohingegen der Rest zur Verifikation des Neuronalen-Netzes Verwendung findet. Als Eingabeschicht dient der Aufprallwinkel und der Kontaktpunkt vom *poke*-Werkzeug und in der Ausgabeschicht die Beschleunigung der Orientierung und Position des Objektes. Bevor der Roboter zielgerichtet das Objekt in eine Richtung schubsen kann, wird der gewichtet quadratische Fehler zwischen der vorhergesagten Bewegung und der gewünschten Bewegung minimiert. Diese Optimierung führt zu einer verbesserten Gesamtbewegung. Ein Resultat eines Experiments ist in Abbildung 3.2c dargestellt, in dem ein Objekt in vier Ecken gezielt verschoben wurde.

In den beiden Arbeiten [KB08] und [KOB10] von Dov Katz et al., sowie in der Arbeit von Jürgen Sturm et al. [SPS+09] werden gleichfalls Objekte, bzw. deren starre Körper, mit einem Roboterarm bewegt. Allerdings ist diesmal das Ziel das kinematische Modell von einem unbekannten Objekt zu bestimmen. In [KB08], auf was ich mich nachfolgend beziehe, werden speziell die Dreh- und

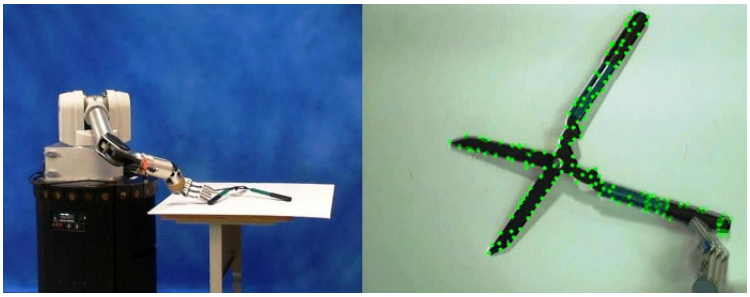


Abbildung 3.3: Links der Roboterarm zum Finden des kinematischen Modells vom Objekt. Rechts die Szenenaufnahme aus dem Blickwinkel der Kamera mit den detektierten Merkmalen (grüne Punkte). [KB08]

Schubgelenke eines planar liegendenden Objektes analysiert. Dazu führt der Roboterarm verschiedene vorher aufgezeichnete Interaktionen mit dem Objekt aus, die von einer Kamera beobachtet werden, siehe Abbildung 3.3. Einzelne Bewegungen in der Szene werden mit Hilfe des optischen Flusses im Bild wahrgenommen, welche fortlaufend markante Merkmale im Bild verfolgt und somit die Trajektorien von jedem einzelnen Merkmal liefert. Ausgehend von diesen Informationen wird ein ungerichteter Graph erstellt. Jeder Knoten im Graph repräsentiert ein Merkmal und eine Kante existiert nur dann, wenn die relative Entfernung zwischen zwei Knoten während der ganzen Beobachtung unter einem bestimmten Schwellwert lag. Aufgrund der Tatsache, dass sich die Merkmale auf einem starren Körper relativ zueinander nicht stark bewegen, spiegelt jeweils ein stark vernetzter Teilgraph den Hintergrund oder einen starren Körper vom Objekt wieder. Die Ermittlung der Teilgraphen erfolgt über die iterative Anwendung des *min-cut* Algorithmus auf dem erzeugten Graphen, wobei kleine Teilgraphen mit weniger als vier Knoten verworfen werden. Die Identifiziernug der Gelenke erfolgt sukzessiv, indem jeweils die relative Bewegung zweier Teilgraphen aus der Menge aller Teilgraphen analysiert wird. Zwei starre Körper die mit einem Drehgelenk verbunden sind weisen z.B. eine gemeinsame Rotationsachse auf. Dies entspricht im Graphen dem Knoten der konstant eine Entfernung zwischen zwei Teilgraphen hat. Daher werden alle Knoten im Graphen, die mit zwei Teilgraphen verbunden sind, zur Menge der gefundenen Drehgelenke zugeordnet. Schubgelenke haben indessen die Eigenschaft, dass die starren Körper eine Translation zueinander aufweisen. Zur Erkennung dieser werden von den beiden Körpern die Positionen vor und nach der Interaktion abgespeichert. Anschließend wird die Transformationsmatrix der Bewegung vom ersten Körper bestimmt und auf den zweiten Körper angewendet. Weicht die beobachtete Position des zweiten Körpers mit der berechneten Position stark ab, wird von einem Schubgelenk ausgegangen. Stimmen die beiden Position jedoch überein, gehören die beiden Körper anscheinend zu einem Körper. Der Rest der Teilgraphen, bei denen keine Gelenke identifiziert wurden, werden dem Hintergrund zugeordnet.

3.1.3 Sensormotorisch selbst beziehend

Sensormotorisch selbst beziehende Systeme explorieren ihre eigenen Sensoren und Aktoren, in der Art und Weise, dass sie die Funktionsweise oder neue Fähigkeiten autonom erlernen. Darunter fallen zum Beispiel die Exploration von Bewegungsmustern, Hand-Auge Koordination oder motorische Feinfühligkeiten [LMPS03].

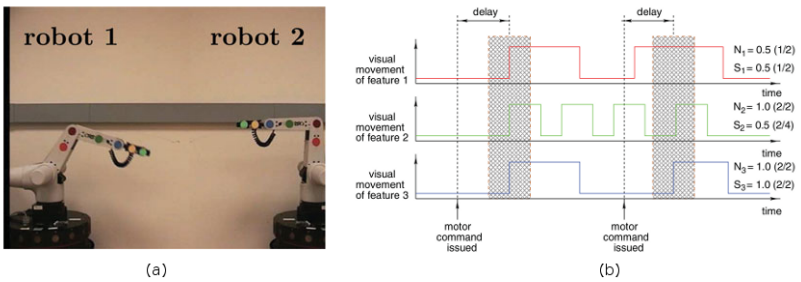


Abbildung 3.4: (a) Darstellung zweier unkorreliert, bewegender Roboterarme. (b) Diagramme von drei Markerbewegungen nach zwei gesendeten Motorkomanndos, geändert aus [Sto11].

A. Stoytchev befasst sich in [Sto11] mit der autonomen Selbstdetektion von Robotern. Die Absicht ist es die eigenen Bewegungen von anderen Bewegungen zu differenzieren, um später Aussagen zu treffen, ob Körperteile oder Objekte zum Roboterkörper gehören oder nicht. Dafür wird in der Arbeit der Roboterarm, wie in Abbildung 3.4a zu sehen, an verschiedenen Körperstellen mit farbigen Markern ausgestattet. Als erstes wird untersucht, wie die zeitliche Verzögerung zwischen einem gesendeten Motorkommando und der wahrgenommenen Eigenbewegung ist. Dafür beobachtet eine stationäre Kamera die ganze Szene, worin am Anfang nur ein Roboterarm zu sehen ist. Damit der Roboter seine Bewegungen observieren kann, führt dieser willkürliche Bewegungen aus. Dabei versucht der Roboter immer eine zufällige Zielkoordinate des Endeffektors zu erreichen. Gelingt es nicht die Zielposition in einer festgelegten Zeit zu erreichen, wird eine neue zufällige Zielkoordinate ausgewählt. Diese Technik ist unter dem Begriff *motor babbling* bekannt. Die sich bewegenden Markern werden mittels einer HSV-Farbsegmentierung lokalisiert und verfolgt. Verändert sich ein Marker über einen bestimmten Schwellwert wird es als Bewegung aufgefasst andernfalls nicht. Das führt dazu, dass im jedem Frame ein binäre Auswertung der Marker erfolgt. Aus den binären Signalen werden die Zeitverzögerungen bestimmt, indem für jeden Marker die Zeiten zwischen dem gesendeten Motorkommando und der ersten Bewegung gemessen werden. Die Zeit mit der höchsten Häufigkeit, ist dann die erwartete Zeitverzögerung nach einem Motorkommando. Mit dem gelernten Wissen über die erwartete Zeitverzögerung, kann das Problem der Differenzierung der Eigenbewegung zu anderen Bewegungen angegeangen werden. Dazu wurde ein zweites Experiment durchgeführt, worin ein zweiter

Roboter in dieselbe Szene gesetzt wurde. Beide Roboter führen wieder den *motor babbling* Algorithmus aus. In Abbildung 3.4b sind drei exemplarische Diagramme dargestellt mit Markerbewegungen aus der Sicht eines Roboters. Zur Bestimmung der Eigenbewegung werden zwei Werte, *necessity index* (N_i) und *sufficiency index* (S_i) berechnet. Der erste Wert ist der Quotient von der Anzahl an Markerbewegungen im erwarteten Zeitfenster zu der Anzahl an gesendeten Motorkommandos. Beim *sufficiency index* hingegen ist der Divisor die Anzahl der erkannten Markerbewegungen. Beide Werte müssen über einen bestimmten Schwellwert liegen, damit die Bewegung eines Markers als Eigenbewegung wahrgenommen wird. In den ersten beiden Diagrammen in Abbildung 3.4b ist das nicht der Fall. Im oberen Diagramm fehlt beim zweiten Motorsignal die erwartete positive Flanke und im mittleren Diagramm wurden zu viele Bewegungen detektiert. Allein das untere Diagramm erfüllt beide Bedingungen. Daher betrachtet der Roboter den letzteren Marker als einen Teil seiner Eigenbewegung.

3.2 Zusammenfassung und Diskussion

In diesem Kapitel wurden exemplarisch einige kognitive Systeme, die dem natürlichen Lernen aus der Sicht eines Kleinkindes nahe kommen, vorgestellt. Sozial interagierende Systeme versuchen Wissen und Fähigkeiten mit Hilfe von Menschen zu erlenen und haben häufig einen kommunikativen Ansatz, wohingegen bei autonomen Systemen das selbstständige Entdecken der Umwelt im Vordergrund steht. Sensormotorisch selbst bezogende Systeme versuchen wiederum eine Selbstwahrnehmung zu entwickeln, die für andere Systeme eine notwendige Voraussetzung bilden.

Vorteil der sozial interagierenden Systeme ist es, dass Menschen als Experten beim Lernen zur Seite stehen. Diese können im Falle eines Lernfehlers direkt eingreifen, Fragen beantworten oder Fähigkeiten unterrichten. Jedoch liegt hier auch ein gravierender Nachteil, weil die sozial interagierenden Systeme maßgeblich vom Menschen abhängig sind. Ein Großteil der kognitiven Fähigkeiten wird dem Menschen auferlegt und daher fällt es den Systemen schwerer oder es ist ihnen unmöglich eigene neue Fähigkeiten hervorzubringen. Der Vorteil bei autonomen Systemen ist es gerade, dass sie ohne Hilfe von Menschen ihre Umwelt selbständig explorieren und versuchen aus ihr zu lernen. Ein Nachteil ist allerdings, festzulegen was in der Umwelt für das System interessant sein kann und wie dieses aus den gewonnen Informationen eine geeignete Struktur bzw. Repräsentation aufbaut. Bei sensormotorisch selbst bezogenden Systemen ist der Vorteil, dass viel aus der eigenen Beobachtung erlernt werden kann und sich diese auf wechselnde Umweltgegebenheiten einstellen können. Die häufig lange Lernzeit kann als Nachteil gewertet werden. Oft verändern sich die sensormotorischen Fähigkeiten nicht und können daher dem System fest vorgegeben werden. Gemein haben alle Systeme, dass die Interaktion (mit der Umwelt, dem Menschen oder mit sich selbst) einen hohen Stellenwert einnimmt.

4 Systemkonfiguration und -umgebung

In diesem Kapitel wird die für die Masterarbeit zur Verfügung stehende Systemumgebung und deren technische Eigenschaften aufgezeigt. Des Weiteren wird die wesentliche Konfiguration des Systems beschrieben, die für das im nächsten Kapitel vorgestellte Verfahren die Grundlage ist.

4.1 Verwendete Hardware

In dieser Arbeit wird als Manipulator ein Light Weight Arm der Firma Schunk eingesetzt, siehe Abbildung 4.1a. Dieser besitzt 7 Freiheitsgerade (engl. degree of freedom, kurz DoF) mit dem ein maximaler Arbeitsbereich von 1,87 x 1,65 x 1,87 m^3 angefahren werden kann. Als Endeffektor kommt eine Schunk Dextrous Hand zum Einsatz, welche der menschlichen Hand nachempfunden ist. Allerdings besitzt diese nur drei Finger mit insgesamt 7 Freiheitsgeraden. Das Verhältnis zwischen der Größe einer menschlichen Hand und der Roboterhand beträgt $1 : 1,4$. Dadurch sind die Finger entsprechend größer und das Greifen und Verschieben von kleineren Gegenständen wird erschwert. Zusätzlich sind an den Fingerspitzen und an den mittleren Fingersegmenten jeweils Drucksensoren angebracht. Angesteuert wird die Hand und der Arm über einen CAN-Bus, welcher über einen CAN-USB-Konverter an ein 8-Kern Rechner angeschlossen ist [Sch11].

Zur 3D-Bilderfassung wird eine Kinect Kamera der Firma Microsoft verwendet. Diese liefert von einer Szene über einen RGB-Sensor ein 640x480px großes Farbbild mit einer maximalen Framerate von 30Hz. Das für die 3D-Rekonstruktion benötigte korrespondierende Tiefenbild wird durch eine Kombination aus einem monochromen Bildsensor und einem Infrarot Projektor erzeugt. Dazu strahlt der Projektor ein strukturiertes, für den Menschen unsichtbares, Infrarotmuster aus. Das Muster wird an den Oberflächen deformiert und auf einem monochromen Bildsensor zurück reflektiert. Aus diesen Informationen berechnet ein eingebauter Prozessor mit einem patentierten Verfahren von PrimeSense (namens *Light Coding*) das benötigte Tiefenbild, siehe [Pri11]. Der Arbeitsbereich, indem Tiefeninformationen sicher bestimmt werden, liegt durch experimentelle Versuche zwischen $0,7$m bis 4m. Der Abstand der optischen Zentren zwischen der RGB-Kamera und der monochromen Kamera beträgt ca. 2,5cm. Daneben verfügt die Kinect Kamera über vier Mikrofone, die unter anderem für eine

Trennung von Sprachsignalen und Störsignalen sorgt. Zusätzlich hat die Kamera einen Freiheitsgrad, da sich im Gehäusefuß ein kleiner Motor befindet mit dem ein Neigungswinkel von $\pm 28°$ eingestellt werden kann, siehe [Mic11]. Anfangs waren die Sensoren und Aktoren der Kinect Kamera ausschlichlich für die kontaktlose Bewegungssteuerung von Konsolenspielen aus dem Hause Microsoft vorgesehen. Allerdings führte *Reverse Engineering* und später auch die Veröffentlichung der Referenztreiber dazu, dass die Kinect Kamera auch für andere Anwendungsbereiche offen steht und somit in der Arbeit verwendet werden kann.

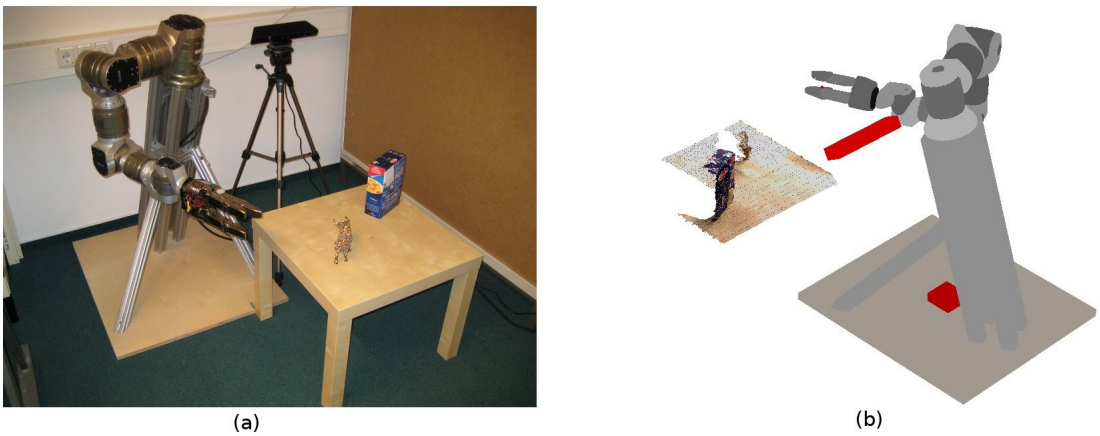


Abbildung 4.1: (a) Reale Darstellung des Versuchsaufbaues und (b) die dazugehörige Abbildung im Simulator.

4.2 Anpassung des Simulators

Für die Steuerung und die Trajektorienplanung des Roboterarms wurde auf eine bereits bestehende Bibliothek[1] im Fachbereich Informatik von der Arbeitsgruppe maschinelles Lernen zurückgegriffen. Diese benutzt eine approximative Inferenz Methode basierend auf einer stochastisch optimalen Steuerung, siehe [Tou09]. Zur Berechnung der Trajektorie und der Kollisionen nutzt der Algorithmus einen Simulator, in dem die Welt abgebildet ist. Bisher werden zum Beispiel die realen Objekte durch vorgefertigte Formen (wie z.B. Zylinder, Quader, Kugeln) per a priori Wissen modelliert. Daraus folgt, dass alles was im Simulator falsch oder nicht abgebildet ist, auch nicht von der Kollisionserkennung richtig erkannt wird. In dieser Arbeit kann und sollen allerdings keine Annahme über die Objektformen gemacht werden. Um nicht einzelne Objekte zu modellieren musste der Simulator angepasst werden. Dazu wurde die gesamte 3D-Szene mit Hilfe der Kinect Kamera im Simulator abgebildet, indem aus dem Tiefenbild

[1] *Robot Simulation Toolkit*, siehe http://userpage.fu-berlin.de/~mtoussai/source-code/

und dem Farbbild eine Punktwolke erstellt wurde. Zur besseren Darstellung wurde die Punktmenge zusätzlich durch eine einfache Triangulation in ein Dreiecksnetz überführt und überlagert dargestellt, siehe Abbildung 4.1b. Allerdings funktionierte danach der Algorithmus zur Kollisionserkennung nicht mehr, da dieser nur mit geschlossenen konvexen Formen umgehen kann. Als Ersatz wurde daher ein alternativer Algorithmus eingesetzt. Dieser betrachtet alle Segmente des Roboterarms und die 3D-Szene als Punktwolken. Anschließend wird in jedem Zeitschritt eine approximierte Nächste-Nachbar Suche, siehe [ML09], zwischen jedem Segment des Roboterarmes zur 3D-Szene durchgeführt. Liegt die Entfernung zwischen einem Segment und der Szene unter einem festgelegten Schwellwert wird dies als potentielle Kollision bewertet und die Trajektorie entsprechend „bestraft“. Der Nachteil der Methode ist der hohe Rechenaufwand, da eine hohe Anzahl von Punkten untersucht werden müssen. Zur Steigerung der Performance wurde deshalb die Punktwolke der 3D-Szene vorher ausgedünnt und alle Punkte die einen größeren Abstand als 100cm haben entfernt. Damit der Roboterarm nicht durch eine zu sehr ausgedünnte Punktwolke hindurchgreifen kann ist der maximale Zwischenraum der Punktwolke auf 1cm festgelegt worden. Dieser Wert begründet sich darauf, dass ein Roboterfinger eine Breite von ca. 2cm besitzt und er somit nirgendwo durch die Punktwolke durchkommen kann, ohne eine Kollision zu verursachen.

4.3 Kalibrierung der Kamera

Um das geometrische Verhältnis zwischen Bildkoordinaten $\vec{x_i}$ der Kamera und Weltkoordinaten $\vec{X}_i$ des Roboters zu beschreiben und um eine perspektivisch korrekte Abbildung der aufgenommenen Szene für den Simulator zu bestimmen, ist es notwendig die Kamera zu kalibrieren. Der erste Ansatz zur Kamerakalibrierung besteht häufig darin, die Kamera als ideale Lochkamera anzusehen. Geometrische Abbildungsfehler werden, aufgrund der fehlenden Linse, in dieser Betrachtung vorerst nicht berücksichtigt. Dadurch reduziert sich das Problem auf das Auffinden einer Projektionsmatrix P, so dass im optimalen Fall gilt:

$$\mathcal{P}(\vec{x_i}) = \mathcal{P}(P\vec{X}_i) \tag{4.1}$$

Wobei $\mathcal{P}$ einer nicht-linearen perspektivische Projektion von homogenen Bildpunkten beziehungsweise von homogenen Weltkoordinaten entspricht. Die Projektionsmatrix setzt sich aus intrinsischen und extrinsischen Parametern zusammen.

$$P = K[R|\vec{t}] \tag{4.2}$$

Die intrinsischen Parameter K umfassen hierbei die spezifischen Kamerawerte, wie die Brennweite und die Pixelskalierung. Wohingegen die extrinsischen Parameter, mit dem Translationsvektor $\vec{t}$ und der Rotationsmatrix R, die Pose der Kamera in der Welt beschreiben. Es wird ersichtlich, dass die

intrinsischen Parameter für eine Kamera konstant bleiben und daher nur einmal berechnet werden müssen. Dagegen müssen auf jede Änderung der Pose die extrinsischen Parameter neu bestimmt werden. Im vorliegenden Fall erhalten wir die intrinsischen Parameter bereits über die API[2] der Kinect Kamera. Gleichzeitig korrigiert sie den geometrischen Abbildungsfehler der realen Kamera und transformiert die Koordinatensysteme der RGB-Kamera und der monochromen Kamera aufeinander, um den bestehenden Abstand der beiden optischen Zentren auszugleichen.

Eine explizite Berechnung fällt daher nur für die extrinsischen Parameter an. Eine häufig eingesetzte Methode ist es ein Schachbrettmuster als Kalibrierungsobjekt zu benutzen. Darauf lassen sich die Eckpunkte der Felder häufig leicht im Bild detektieren. Mit dem Wissen über die Feldgröße und die Feldanzahl lässt sich dann die Pose der Kamera errechnen. Allerdings hat das Verfahren auch einige Nachteile. Das Schachbrettmuster muss ausreichend groß und sauber ausgedruckt und knitterfrei manuell vor die Kamera gelegt werden. Anschließend muss der Offset, vom Weltkoordinatensystem des Roboters zum Kalibrierungsobjekt, dem System mitgeteilt werden. Dazu muss der Abstand jedes Mal von Hand ausgemessen werden, wenn sich die Position des Kalibrierungsobjektes verändert hat, was alles sehr fehleranfällig und zeitaufwändig ist. Daher wird hier ein anderes Verfahren eingesetzt, welches ohne eine Person auskommt und daher dem autonomen Lernen der Hand-Augen Koordination nahe kommt, siehe Kapitel 3.1.3 für die Definition.

Die Idee besteht darin, dass der Roboterarm autonom in der Szene ein Schachbrettmuster zeichnet und die benötigten Bild- und Weltkoordinaten selber errechnet, vgl. [GBBK10]. Dazu wird dem Roboterarm am Handgelenk eine Leuchtdiode (LED) angebracht, um die Bestimmung der Bildkoordinaten zu erleichtern. Anschließend fährt der Roboterarm in einem leicht verdunkelten Raum N Punkte im Form eines Schachbrettmusters ab und berechnet sich durch Vorwärtskinematik direkt die Weltkoordinate $\vec{X}_i$ der LED. Die korrespondierenden Bildkoordinaten $\vec{x}_i$ entstehen aus der Detektion des LED-Mittelpunktes im aufgenommen RGB-Bild der Kamera. Dafür wird das RGB-Bild zunächst einmal in den HSV-Farbraum transformiert und der V-Kanal (Helligkeits-Kanal) mit einem einfachen Schwellwert binarisiert. Als Ergebnis sind nur noch sehr helle Stellen im Bild vorhanden. Damit die Lokalisierung robuster gegen Rauschen und Störungen wird, werden im nächsten Schritt Kreise im Bild gesucht. Hierfür werden zunächst einmal Kanten erzeugt, indem das binarisierte Bild von einer mittels Erosion verkleinerten Bildkopie subtrahiert wird. Anschließend wird eine vereinfachte Version der Hough-Transformation für Kreise durchgeführt, der um jeden Kantenpunkt im Parameterraum einen diskretisierten Kreis mit unterschiedlichen Radien akkumuliert. Die Vereinfachung ist möglich, da wir nur an den Kreismittelpunkten und nicht an den tatsächlichen Radien der Kreise interessiert sind. Der Ort im Parameterraum an dem sich die meisten Kreise schneiden, ist dann der gefundene

[2] Hier der OpenNI API: http://www.openni.org

Kreismittelpunkt der LED. Da dieser einen diskreten Wert hat, wird um dem Kreismittelpunkt im binarisierten V-Kanalbild eine kleine Region definiert. Die dann benutzt wird, um den Mittelpunkt der LED noch genauer zu präzisieren, indem der Bildschwerpunkt der Region berechnet wird.

Nachdem alle Welt- und Bildkoordinaten ermittelt wurden, erfolgt die Berechnung der Projektionsmatrix P auf Basis des sogenannten Reprojektionsfehlers e_{rep}. Dieser stellt mit folgender Formel den Fehler des euklidischen Abstandes zwischen den projektierten Weltkoordinaten und den detektierten Bildkoordinaten dar:

$$e_{rep} = \frac{1}{N} \sum_{i=1}^{N} ||(\mathcal{P}(\vec{x_i}) - \mathcal{P}(P\vec{X}_i)||^2 \tag{4.3}$$

Kleine Fehler weisen auf eine gute Projektionsmatrix hin, deswegen gilt es die Kostenfunktion der Formel 4.3 zu minimieren. Aufgrund der nicht-linearen perspektivische Projektion $\mathcal{P}$ wird die Funktion über ein Gradientenabstiegsverfahren minimiert. Damit ist das Problem der Kalibrierung gelöst. Es können nun zu jeder Weltkoordinate, ausgehend von der Formel 4.1, die zugehörige Bildkoordinate und umgekehrt über die Projektionsmatrix ermittelt werden.

4.4 Detektierung der planaren Hauptfläche

Im Versuchsaufbau sind die unbekannten Objekte auf einem kleinen Tisch platziert, weil in typischen Szenarien die Objekte meistens auf einer planaren Oberfläche (wie Fußböden, Kommoden oder Regale) liegen. Diese Annahme über die Form der Oberfläche wird für die Optimierung der späteren Algorithmen ausgenutzt, siehe Kapitel 5.2.1 und Kapitel 5.3. Daher wird im Folgenden auf das Auffinden der planaren Hauptfläche in einer Punktwolke mit Hilfe des RANSAC-Algorithmus eingegangen.

RANSAC (engl. random sample consensus) ist ein iterativer Algorithmus, der erstmalig 1981 von Fischler und Bolles in der Arbeit [FB81] vorgestellt wurde. Im Gegensatz zu der einfacheren Methode der kleinsten Quadrate hat dieser den Vorteil, dass eine robuste Schätzung der Parameter eines vorgegebenen Modells auch bei relativ stark verrauschten Daten möglich ist. Das Modell ist in diesem Fall die folgende Ebenengleichung:

$$Ax + By + Cz + D = 0 \tag{4.4}$$

mit den vier unbekannten Paramatern $\theta \in \{A, B, C, D\}$. Die Daten sind die Koordinaten der Punktwolke $\vec{p_i} = (x_i, y_i, z_i)^T$. In dieser wählt der Algorithmus im ersten Schritt jeder Iteration drei zufällige Punkte aus. Dabei ist wichtig, dass die gewählten Punkte nicht kollinear sind. Erfüllen sie diese notwendige Eigenschaft werden die Parameter θ für eine Modellhypothese wie folgt bestimmt:

$$(A, B, C)^T = (\vec{p_2} - \vec{p_1}) \times (\vec{p_3} - \vec{p_1}) \tag{4.5}$$

$$D = -(A, B, C)^T \cdot \vec{p_1} \tag{4.6}$$

Anschließend wird anhand der aufgespannten Ebene eine sogenante Konsensmenge (engl. consensus set) ermittelt. Diese bestehen aus jenen Punkten, die einen geringen geometrischen Abstand von $d < T$ zur Ebene aufweisen, wobei T ein definierter Schwellwert ist. Unter der Voraussetzung, dass die Parameter θ auf 1 normalisiert sind, kann für die Distanzberechnung zwischen einem Punkt und einer Ebene die folgende Gleichung herangezogen werden:

$$d = |Ax + By + Cz + D| \tag{4.7}$$

Nach N Iterationen mit jeweils zufälligen Punkten wird die beste Modellhypothese genommen, welche die größte Konsensmenge besitzt.

Allerdings beschreibt das Modell eine unendlich ausgedehnte Ebene im Raum. Reale Tische oder Stuhlflächen besitzen aber nur eine endliche Ausdehnung. Daher wird anhand der Konsensmenge zusätzlich die konvexe Hülle berechnet. Dazu werden als Erstes alle Punkte auf die Modellebene projiziert, um das Finden der konvexen Hülle von einem 3D-Problem auf ein 2D-Problem zu reduzieren. Als Algorithmus kommt der rekursive *Quickhull*-Algorithmus zum Einsatz, siehe [PS85]. Dieser bestimmt aus der projizierten Punktmenge die zwei entferntesten Randpunkte, die logischerweise auf jedenfall zur konvexen Hülle gehören. Zwischen diesen beiden Punkten wird eine Gerade gezogen, welche die Punktmenge in zwei Hälften unterteilt. Innerhalb einer Hälfte wird dann der Punkt gesucht, von dem der Abstand zur Geraden maximal ist. Der maximale Punkt ist Teil der konvexen Hülle und bildet zudem mit den anderen zwei Punkten ein Dreieck. Punkte die sich im Inneren des Dreiecks befinden, brauchen im weiteren Verlauf nicht weiter betrachtet zu werden, da diese definitiv nicht zu der Menge der konvexen Hülle gehören. Die Seiten des Dreiecks teilen die Punktwege weiter in kleinere Teilmengen, die nach dem gleichen Prinzip rekursiv weiter verarbeitet werden. Der Algorithmus terminiert, wenn keine Punktmenge weiter unterteilt werden kann.

Ausschlaggebend für den Erfolg des RANSAC-Algorithmus und dem nachfolgenden *Quickhull*-Algorithmus ist es, dass genügend Punkte für die Generierung der Modellhypothesen zur Verfügung stehen. Dafür darf beim RANSAC-Algorithmus die maximale Iterationsanzahl und der Schwellwert für den geometrischen Abstand nicht zu gering gewählt werden. Beides führt sonst zu Ungenauigkeiten im Modell. In dieser Arbeit erbrachten die experimentell gefundenen Werte $T = 0,01$ und $N = 1000$ gute Ergebnisse.

5 Aufmerksamkeitsgetriebene Objektexploration

Im diesem Kapitel wird das eingesetzte Verfahren vorgestellt. Dafür wird am Anfang eine Übersicht über das ganze System gegeben und nachfolgend die einzelnen Schritte im Detail erklärt.

5.1 Übersicht des Gesamtsystems

Das entwickelte System verfügt über zwei alternierende Phasen. Eine Hypothesengenerierungsphase und eine Hypothesenverifizierungsphase, welche der Aufgabe des menschlichen dorsalen („Wo“) und des ventralen („Was“) Pfades im Gehirn nahekommt, siehe [Sch00] und Kapitel 2.1.3.

Bei der Hypothesengenerierungsphase ist somit das Ziel eine möglichst genaue Hypothese über den Aufenthaltsorts eines realen Objektes zu erstellen. Dafür wird am Anfang eine Aufmerksamkeitskarte erzeugt, in der topologisch verschiedene fusionierte visuelle Merkmale verzeichnet sind. Orte in der Karte mit hohen Aufmerksamkeitswerten gelten für das System als besonders interessant, weil die Annahme besteht, dass sich dort besonders häufig Objekte befinden. Dementsprechend wird immer der Explorationsort ausgewählt, der einen maximalen Aufmerksamkeitswert aufweist. Damit das System nicht immer den gleichen Ort untersucht muss die Aufmerksamkeit bei erfolgreicher Besichtigung auf einen anderen Ort gelenkt werden. Es erfolgt durch eine Hemmung aller vergangenen Explorationsorte in der Aufmerksamkeitskarte. Dies hat zur Folge, dass immer uninteressantere Orte (mit niedrigen Aufmerksamkeitswerten) exploriert werden. Sinkt der Wert unter einem vorher festgelegten Schwellwert ab, wird dem System signalisiert, dass es nichts Interessantes mehr in der Szene zu entdecken gibt. Daraufhin ist die Objektexploration beendet oder das System wendet sich einer neuen Szene zu. Liegt der Wert allerdings über dem Schwellwert, wird an dem Ort eine Objekthypothese erstellt, indem die Ausdehnung des Objektes ermittelt wird.

Die anschließende Hypothesenverifizierungsphase versucht die Hypothese über den Aufenthaltsort eines Objektes zu verifizieren. Als Erstes vergleicht das System anhand von visuellen Merkmalen, ob die Objekthypothese mit einem memorierten Objekt übereinstimmt. Konnte ein memoriertes Objekt gefunden werden, gilt die Hypothese als bekannt und daher als bereits verifiziert. Es kann daher direkt eine neue Objekthypothese generiert werden. Konnte kein memoriertes Objekt gefunden werden,

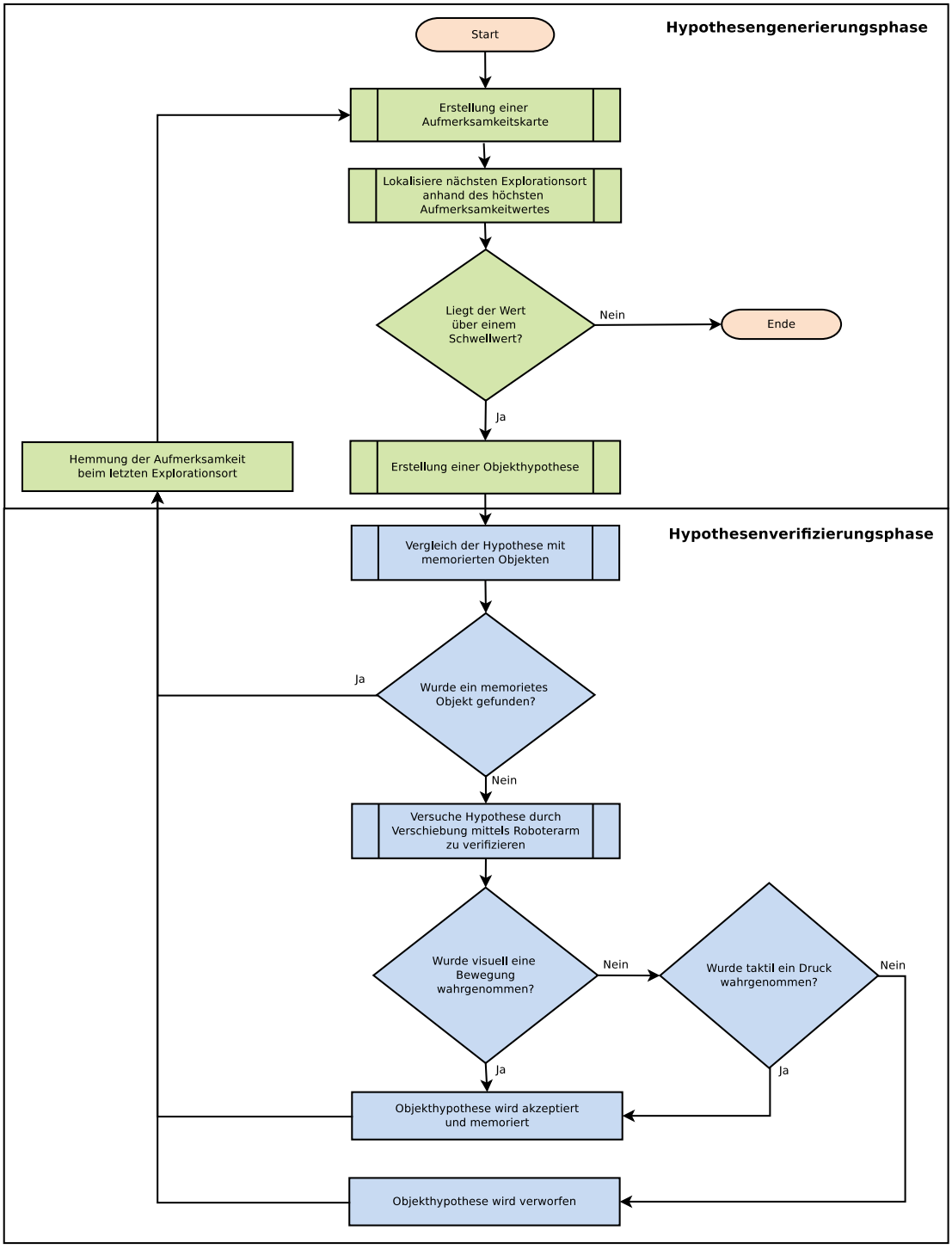


Abbildung 5.1: Programmablaufplan der aufmerksamkeitsgetriebene Objektexploration

geschieht die Verifizierung anhand visueller und taktiler Informationen. Bei einer verifizierten Hypothese, erfolgt zugleich eine Eigenschaftenzuweisung über deren Beweglichkeit. Die einzelnen Beziehungen der Phasen und deren Komponenten zueinander sind in Abbildung 5.1 als Programmablaufplan graphisch ersichtlich.

5.2 Aufmerksamkeitsbasierte Suche nach Explorationsorten

Das Ausgangsmodell zur aufmerksamkeitsbasierenden Suche nach Explorationsorten zur Generierung von Objekthypothesen ist das Aufmerksamkeitsmodell von Itti et al. [IKN98]. In dieser Version von 1998 ist das Auffinden ein reiner *bottom-up* gesteuerter Prozess[1], was insbesondere nach den Erkenntnissen der menschlichen visuellen Wahrnehmung modelliert ist, siehe Kapitel 2. Das Fehlen des *top-down* Prozesses kann in dieser Arbeit vernachlässigt werden, da keine aufgabenspezifischen Objektannahmen getätigt werden, wie zum Beispiel das bevorzugte Finden von Verkehrsschildern oder roten Bällen in einer Szene.

Das Ausgangsmodell von Itti et al. extrahiert aus einem RGB-Bild zuerst verschiedene elementare visuelle Merkmale, wie Winkel, Intensität und Differenzen von Farbkanälen. Nach dem Prinzip der *On/Off-Center* Neuronen werden aus den extrahierten Merkmalen mehrere unterschiedlich große topografische Merkmalskarten (engl. feature maps) berechnet und normalisiert. Zum Wiedererwerb der ursprünglichen Bildgröße werden diese zu drei gleichgroßen Auffälligkeitskarten (engl. conspicuity maps), entsprechend der Dimensionen Intensität, Farbe und Orientierung, interpoliert und normalisiert. Anschließend werden die Auffälligkeitskarten nochmals zu einer Aufmerksamkeitskarte (engl. saliency map) linear kombiniert, um die separaten Dimensionen in einer einzigen Karte zu repräsentieren. Das Lokalisieren des Ortes mit der höchsten Aufmerksamkeit funktioniert über ein *winner-take-all* Neuronales-Netz. Dieses wertet die Aufmerksamkeitskarte und eine sogenannte Hemmungskarte (engl. inhibition map) aus, welche die Aufgabe hat einen Wechsel zu verschiedenen Orten zu ermöglichen.

Vom Ausgangsmodell wurde ein angepasstes und erweitertes Aufmerksamkeitsmodell erstellt, welches in Abbildung 5.2 dargestellt ist. Das Modell und die Erweiterungen werden im Einzelnen in den nachfolgenden Unterkapiteln besprochen.

[1] In der neueren Version von 2005 wurde das Modell zusätzlich um ein *top-down* Prozess erweitert, siehe [NI05].

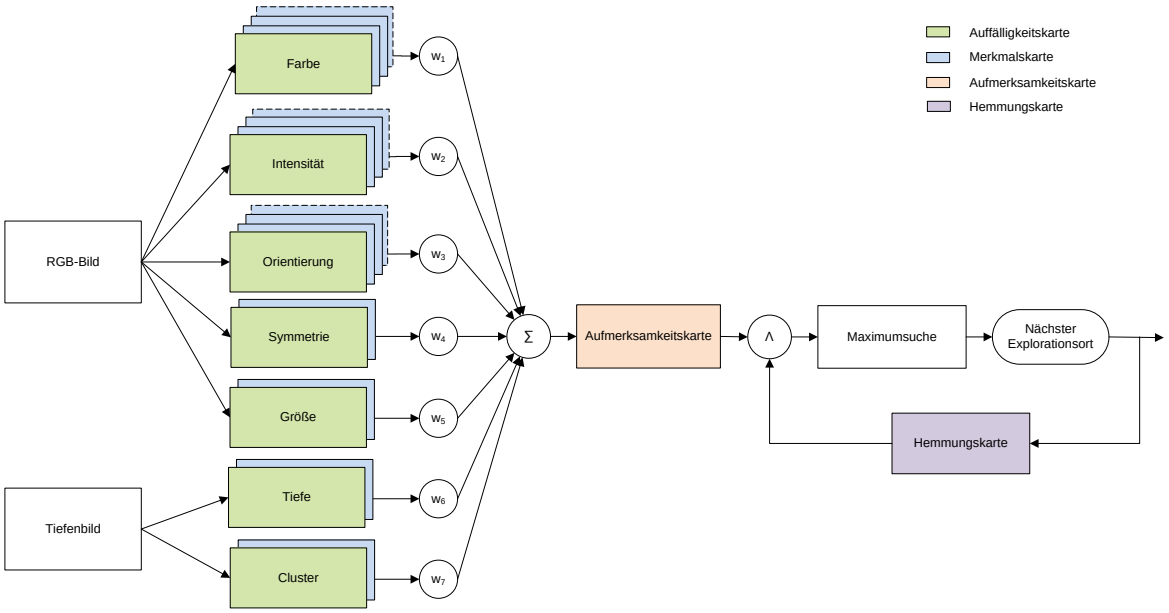


Abbildung 5.2: Das erweiterte Aufmerksamkeitsmodell zum Auffinden des nächsten Explorationsortes, basierend auf [IKN98].

5.2.1 Generierung von Merkmalskarten

Die Generierung von Merkmalskarten bildet die Grundlage für das Aufmerksamkeitsmodell. Die Berechnung erfolgt typischerweise für die ganze Szene, da sich überall im Bild Merkmale befinden können. Eine Dimension kann aus einer einzelnen oder aus mehreren Merkmalskarten bestehen. Wichtig ist vor allem, dass die topologischen Informationen der Merkmale nicht verloren gehen. Da keine aufgabenspezifischen Objektannahmen im Modell erstellt werden, muss sich die Auswahl der Merkmale nach allgemeingültige Richtlinien orientieren. Als Orientierungshilfe dienen die in [Bac04] aufgestellten Merkmalseigenschaften:

- *Informativität*: Möglichst informative Merkmalseigenschaften, die für eine Szene relevant sind, sollten extrahiert werden.
- *Objektzusammenhang*: Merkmale sollten bevorzugt einen Zusammenhang zu einem Objekt haben.
- *Stabilität*: Störungen durch Rauschen oder leichte Szenenveränderungen sollten das Merkmal nicht beeinflussen.
- *Ähnlichkeit zum menschlichen Vorbild*: Merkmale sollten der menschlichen unterbewussten Wahrnehmung ähnlich sein.

- *Komplementarität*: Es sollten viele unterschiedliche Merkmale extrahiert werden, die sich möglichst gut ergänzen.
- *Einfachheit*: Merkmale sollten schnell berechenbar sein.

Im erweiterten Aufmerksamkeitsmodell wurde, im Sinne der Komplementarität, die Anzahl der verwendeten Dimensionen auf sieben erhöht, so dass jetzt für die Auswertung insgesamt folgende Dimensionen zur Verfügung stehen: Intensität, Farbe, Orientierung, Symmetrie, Größe, Tiefe und Cluster.

Intensität

Das Merkmal der Intensität I wird in [IKN98] aus den drei Farbkanälen des RGB-Bildes durch eine einfache arithmetischen Mittelwertsbildung berechnet:

$$I = \frac{r + g + b}{3} \tag{5.1}$$

Für die Nachbildung der *On/Off-Center* Neuronen der menschlichen Retina wird anschließend eine Gauß-Pyramide $I(\sigma)$ mit insgesamt neun Ebenen $\sigma \in [0..8]$ erstellt. Die Konstruktion der Pyramide erfolgt durch eine sukzessive Anwendung eines Gauß-Filters auf die erstellten Ebenen, was jeweils eine Halbierung der Ebenenauflösung bewirkt und die Ebenengröße um die Hälfte verkleinert. Pixel in den höher aufgelösten Ebenen $c \in \{2, 3, 4\}$ werden als Zentrum der *On/Off-Center* Neuronen definiert, wohingegen die Pixel der Umgebung den gröberen Ebenen $s = c + \delta$ mit $\delta \in \{3, 4\}$ zugeordnet sind. Anschließend wird die gröbere Ebene auf die Größe der höher aufgelösten Ebene interpoliert, um eine pixelweise Differenzbildung der Zentren und des Umfeldes durchführen zu können. Diese Berechnung wird auch als Zentrum-Umfeld Operation (engl. center-surround operation) bezeichnet, welches in den nachfolgenden Formeln mit dem Operator $\ominus$ dargestellt wird. Durch die Kombinationsmöglichkeiten der groben und höher aufgelösten Karten ergeben sich insgesamt sechs Merkmalskarten mit:

$$\mathcal{I}(c, s) = |I(c) \ominus I(s)| \tag{5.2}$$

Farbe

Für die Modellierung der Ganglienzellen in der Retina, die für die Farbwahrnehmung verantwortlich ist (siehe Kapitel 2.1.2), werden in [IKN98] die Farbkanäle des RGB-Bildes zunächst in Rot (R), Grün (G), Blau (B) und Gelb (Y)-Kanäle neu berechnet:

$$R = r - \frac{(g + b)}{2} \tag{5.3}$$

$$G = g - \frac{(r+b)}{2} \tag{5.4}$$

$$B = b - \frac{(r+g)}{2} \tag{5.5}$$

$$Y = \frac{(r+g)}{2} - \frac{|r-g|}{2} - b \tag{5.6}$$

Wobei negative Ergebnisse auf 0 gesetzt werden. Die Umrechnung hat zur Folge, dass die Kanäle nur bei den entsprechenden Farben maximal reagieren und die Farben weiß, schwarz und alle Grauwerte „ignoriert“ werden.

Aus den vier neuen Kanälen wird, gemäß der menschlichen visuellen Wahrnehmung, ein Rot-Grün Kanal und ein Blau-Gelb Kanal gebildet, die gleichzeitig als Merkmalskarten fungieren. Die Berechnung erfolgt wieder auf der Grundlage der Zentrum-Umfeld Operation mit den beiden Formeln:

$$\mathcal{RG}(c,s) = |(R(c) - G(c)) \ominus (G(s) - R(s))| \tag{5.7}$$

$$\mathcal{BY}(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))| \tag{5.8}$$

Orientierung

Das letzte verwendete Merkmal im Ausgangsmodell ist die Orientierung. Die annähernd die Verarbeitung der V1-Neuronen im visuellen Kortex nachbildet, siehe Kapitel 2.1.3. Dazu wird ein 2D-Gabor Filter verwendet, der auf Linien und Kanten mit einer vorgebenden Orientierung besonders stark reagiert. Nach [VR06] ist ein 2D-Gabor Filter wie folgt definiert:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{\bar{x}^2}{\sigma_x^2} + \frac{\bar{y}^2}{\sigma_y^2}\right)\right] \exp\left(2\pi j W \bar{x}\right) \tag{5.9}$$

mit

$$\bar{x} = x\cos\theta + y\sin\theta \tag{5.10}$$

$$\bar{y} = -x\sin\theta + y\sin\theta \tag{5.11}$$

wobei σ die Skalierung und θ die Orientierung des Filters bestimmen.

Dieser wird in [IKN98] auf das Intensitätsbild angewendet, woraufhin mehrere Gabor-Pyramiden $O(\sigma,\theta)$ mit den verwendeten Filterparametern $\sigma \in [0..8]$ und $\theta \in \{0°, 45°, 90°, 135°\}$ entstehen. Durch eine abschließende Zentrum-Umfeld Operation ergeben sich insgesamt 24 Merkmalskarten (6 pro Winkelauflösung):

$$\mathcal{O}(c,s,\theta) = |O(c,\theta) \ominus O(s,\theta)| \tag{5.12}$$

Symmetrie

Künstlich geschaffene Objekte und auch viele natürliche Objekte, wie Pflanzen oder Tiere weisen eine hohe Symmetrieeigenschaft auf. Daher liegt es Nahe zusätzlich die Symmetrie als Merkmal mit einzubeziehen. Die Berechnung erfolgt auf Grundlage einer Frequenzanalyse in [Kov97]. Dazu werden mit Hilfe einer *Log-Gabor-Wavelet* Transformation die lokalen Frequenzen und insbesondere die Phasen im gesamten Bild bestimmt. Durch Anwendung von n skalierten geraden und ungeraden Wavelets ergeben sich für einen Pixel mehrere Filterantwortvektoren (engl. filter response vector). Dabei wird die Länge des Vektors über die Amplitude A_n und die Ausrichtung über die Phase ϕ_n bestimmt. Zeigen alle Vektoren bzw. deren Phasen in eine Richtung ist die Symmetrie hoch, wohingegen bei unterschiedlichen Phasenrichtungen keine Symmetrie herrscht. Anschließend werden alle Filterantwortvektoren zu einem Wert kombiniert und in einer Merkmalskarte $\mathcal{V}$ gespeichert, indem für jede Bilddimension die geraden und ungeraden Filterantwortvektoren einer Pixel Differenz unterzogen werden und zusätzlich anhand ihrer Amplitude gewichtet werden:

$$\mathcal{V}(x) = \frac{\sum_n \lfloor A_n(x)[|\cos(\phi_n(x))| - |\sin(\phi_n(x))|] - T \rfloor}{\sum_n A_n(x) + \epsilon} \tag{5.13}$$

Wobei der Parameter T für eine Rauschunterdrückung sorgt und ϵ eine kleine Konstante ist, die dafür sorgt, dass nicht durch Null dividiert wird.

Größe

Die Merkmalskarte in [AHES09] hat kein neurobiologisches Modell als Vorbild. Als Ergebnis des Verfahrens hebt es das größte auffälligste Objekt, durch Analyse des globalen Kontrastes, vollständig hervor. Diese Eigenschaft ist in dieser Arbeit von Vorteil, weil die Roboterhand um das 1,4-fache größer ist als eine menschliche Hand und es daher leichter ist, größere Objekte zu verschieben, siehe Kapitel 4.1.

Dazu wird als Erstes das RGB-Bild in den Lab-Farbraum konvertiert und mit einem Gauß-Filter der Größe $\omega = \frac{\pi}{2{,}75}$ geglättet. Das sorgt dafür, dass kleine Details und Störungen reduziert werden. Anschließend werden die Merkmale für jeden Pixel im geglätteten Bild $\tilde{I_\omega}$ über den euklidischen Abstand zum arithmetischen Mittelwert $\tilde{I}_\mu$ des gesamten Bildes bestimmt:

$$\mathcal{S}(x, y) = ||\tilde{I}_\mu - \tilde{I}_\omega(x, y)|| \tag{5.14}$$

Wie zu erkennen ist erfolgt die Rechnung mit der vollen Auflösung des Ausgangsbildes, was dazu führt das die hervorgehobene Region klar abgegrenzt ist.

Tiefe

Als Merkmal soll die Tiefe mit in die Berechnung der Aufmerksamkeitskarte einfließen. Näher liegende Objekte sollen bevorzugt werden, als weiter entfernte Objekte. Dies erfolgt einerseits auf Grund der Tatsache, dass der Arbeitsraum des Roboterarms begrenzt ist (siehe Kapitel 4.1) und andererseits richtet sich die Aufmerksamkeit der Menschen bevorzugt auf Dinge der unmittelbaren Umgebung.

Die Merkmalskarte kann direkt aus dem Kehrwert des Tiefenbildes Z erstellt werden:

$$\mathcal{D}(x, y) = \frac{1}{Z(x, y)} \tag{5.15}$$

wobei Entfernungen die $Z > 100cm$, $Z = 0$ oder in der Konsensmenge der planaren Hauptfläche liegen auf 0 gesetzt werden.

Cluster

Für das letzte Merkmal im erweiterten Aufmerksamkeitsmodell wird die Annahme ausgenutzt, dass sich häufig Objekte auf einer planaren Oberfläche befinden. Das verwendete Merkmal sind hier die Cluster (konzentrierte Punktwolken), die über der planaren Hauptfläche gesucht werden. Das Auffinden dieser Ebene und die dazugehörige konvexe Hülle wurde bereits in Kapitel 4.4 beschrieben. Daher gilt es im Folgenden die Cluster zu bestimmen.

Dazu wird nach der Formel 4.7 zu jedem Punkt der Abstand zur Ebene berechnet und kontrolliert. Liegt die Distanz zwischen 0,5cm und 80cm und befindet sich der Punkt $\vec{X}_i$ innerhalb der konvexen Hülle wird dieser zu der Menge P aufgenommen. Der gewählte Mindestabstand ist erforderlich, da die Tiefeninformationen von der Kamera nicht stabil sind. Der Maximalabstand gewährleistet indessen, dass die Größe eines Objektes begrenzt ist. Anschließend wird die Punktmenge P nach dem Algorithmus 1 in einzelne Cluster unterteilt. Die Arbeitsweise des Algorithmus funktioniert im Prinzip wie ein *flood-fill* Algorithmus auf Punktwolken, wobei die Zusammengehörigkeit eines Clusters über den euklidischen Abstand d abhängt. Befinden sich rekursiv die Nächste-Nachbarpunkte p_i im Radius des Abstandes d, werden sie zu einer Clustermenge C_i vereint. Alle entfernteren Punkte gehören entsprechend zu anderen Clustermengen. Allerdings werden Clustergrößen die kleiner als $s_{min} = 50$ verworfen, da diese wahrscheinlich aus Rauschen entstanden sind. Die gefundenen Clusterkoordinaten werden anschließend auf eine Merkmalskarte $\mathcal{C}$ in binärer Weise projiziert, d.h. an den Orten wo sich Cluster befinden ist die Merkmalskarte ungleich 0. Dadurch wird das ganze Cluster gleichstark hervorgehoben. Damit die Clusterzentren und große Cluster ein höheres Gewicht erhalten, wird die Merkmalskarte abschließend mit einem großen Gaußkernel von $\sigma = 15$ geglättet.

Algorithmus 1 : Pseudocode zum Euklidischen-Clustern von Punktwolken, basierend auf [Rus09].

Eingabe : Punktwolke P, Clustertoleranz d, minimale Clustergröße s_{min}

Ausgabe : Die Menge an gefundenen Clustern C

$C \leftarrow \emptyset$

$V \leftarrow \emptyset$ // Menge der Punkte, die bereits besichtigt wurden

für alle $p_i \in P$ **tue**

  wenn $p_i \notin V$ **dann**

    $Q \leftarrow \emptyset$ // Initialisiere potentielle Clustermenge mit der leeren Menge

    $Q \leftarrow Q \cup \{p_i\}$

    wiederhole

      $q_i \leftarrow$ Hole ein nicht besichtigtes Element aus $(Q \setminus V)$

      $V \leftarrow V \cup \{q_i\}$

      $P^{NN} \leftarrow$ Hole alle Nächste-Nachbarn von q_i, welche im euklidischen Radius $r < d$ liegen

      für alle $p_i^{NN} \in P^{NN}$ **tue**

        wenn $p_i^{NN} \notin V$ **dann** // Wurde Nachbarpunkt noch nicht besichtigt?

          $Q \leftarrow Q \cup \{p_i^{NN}\}$

    bis $(Q \setminus V) = \emptyset$ // Schleifenabbruch wenn alle Elemente in Q besichtigt wurden

    wenn $|Q| > s_{min}$ **dann**

      $C \leftarrow C \cup \{Q\}$

5.2.2 Fusionierung zur Aufmerksamkeitskarte

Bevor die generierten Merkmalskarten zu einer Aufmerksamkeitskarte fusioniert werden, müssen diese auf ein Einheitsmaß zu sogenannten Auffälligkeitskarten normalisiert werden. Dies ist zwingend erforderlich, weil jede Dimension einen unterschiedlichen Wertebereich aufweist. Hat die Dimension Farbe zum Beispiel einen Wertebereich von $[0..1]$ und die Tiefe einen Wertebereich von $[200..1000]$, dann wird die Farbe fast keinen Einfluss auf die Aufmerksamkeit ausüben. Im Zusammenhang besteht auch bei einigen Dimensionen das Problem, dass einzelne herausragende Spitzen in einer Dimension von Rauschen einer anderen Dimension abgeschwächt oder sogar überlagert werden können. Zum Beispiel ist in Abbildung 5.3 der einzelne vertikale Balken in der Dimension Orientierung klar erkennbar. Allerdings würde dieser durch die einfache Summation mit der Intensität untergehen, weil diese Dimension durch das *salt-and-pepper* Rauschen hohe Spitzen auf der ganzen Merkmalskarte aufweist.

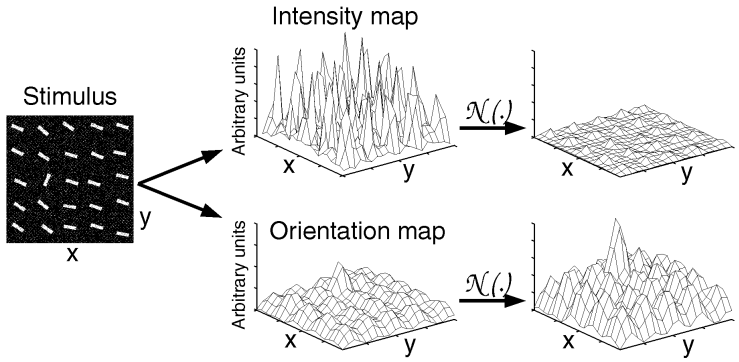


Abbildung 5.3: Veranschaulichung der Normalisierungsfunktion $\mathcal{N}(.)$, anhand der Dimension Intensität und Orientierung. [IKN98]

Daher wurde in [IKN98] die Normalisierungsfunktion $\mathcal{N}(.)$ eingeführt, welche einzelne Spitzen in einer Merkmalskarte aufwertet aber Merkmalskarten mit mehreren gleich hohen Spitzen abschwächt, siehe Abbildung 5.3. Dabei arbeitet die Funktion in den folgenden drei Schritten:

1. Normalisiere die Merkmalskarte auf einen Wertebereich von $[0..M]$.
2. Finde das globale Maximum M und berechne den Durchschnitt aller lokalen Maxima $\bar{m}$.
3. Multipliziere die Merkmalskarte mit $(M - \bar{m})^2$.

Die Auffälligkeitskarten für die Dimensionen Intensität $\bar{\mathcal{I}}$, Farbe $\bar{\mathcal{L}}$ und Orientierung $\bar{\mathcal{O}}$ kann nun mit folgenden Formeln berechnet werden:

$$\bar{\mathcal{I}} = \tilde{\mathcal{N}}\left(\bigoplus_{c=2}^{4}\bigoplus_{s=c+3}^{c+4}\mathcal{N}(\mathcal{I}(c,s))\right) \tag{5.16}$$

$$\bar{\mathcal{L}} = \tilde{\mathcal{N}}\left(\bigoplus_{c=2}^{4}\bigoplus_{s=c+3}^{c+4}[\mathcal{N}(\mathcal{RG}(c,s)) + \mathcal{N}(\mathcal{BY}(c,s))]\right) \tag{5.17}$$

$$\bar{\mathcal{O}} = \tilde{\mathcal{N}}\left(\sum_{\theta\in\{0°,45°,90°,135°\}}\left[\bigoplus_{c=2}^{4}\bigoplus_{s=c+3}^{c+4}\mathcal{N}(\mathcal{O}(c,s,\theta))\right]\right) \tag{5.18}$$

Wobei der Operator $\oplus$, die durch den Zentrum-Umfeld Operator unterteilten Merkmalskarten, wieder zu einer Karte vereinigt. Dieser erfolgt, indem der $\oplus$-Operator alle Ebenen auf die Größe der Ebene $\sigma = 4$ reduziert und dann punktweise addiert. Das resultierende Ergebnis wird anschließend nochmals normalisiert und gleichzeitig auf die Ausgangsgröße des Bildes zurück interpoliert.

Für die vier restlichen Auffälligkeitskarten der Dimension Symmetrie $\bar{\mathcal{V}}$, Größe $\bar{\mathcal{S}}$, Tiefe $\bar{\mathcal{D}}$ und Cluster $\bar{\mathcal{C}}$ werden die Merkmalskarten lediglich auf einen Wertebereich von [0..1] normalisiert. Dies erfolgt aufgrund der Tatsache, dass die zugrundeliegende Merkmalskarte allein ein Maß für die Auffälligkeit bestimmt und das häufige Auftreten eines Merkmales nicht unterdrückt werden soll.

Letztendlich entsteht die Aufmerksamkeitskarte $\mathcal{S}^\star$ im erweiterten Aufmerksamkeitsmodell durch eine gewichtete linear Kombination aller erzeugten Auffälligkeitskarten.

$$\mathcal{S}^\star = \frac{1}{7}[w_1\bar{\mathcal{I}} + w_2\bar{\mathcal{L}} + w_3\bar{\mathcal{O}} + w_4\bar{\mathcal{V}} + w_5\bar{\mathcal{S}} + w_6\bar{\mathcal{D}} + w_7\bar{\mathcal{C}}] \tag{5.19}$$

Die Gewichte w_i bestimmen inwiefern jede Dimension zur Aufmerksamkeit beiträgt. Im einfachsten Fall sind alle Gewichte gleich 1, was bedeutet, dass alle gleichstark zur Aufmerksamkeit beitragen. In Abbildung 5.4 ist beispielhaft eine Aufmerksmakeitskarte zerlegt und in ihre einzelnen Auffälligskeitskarten dargestellt. Wie zu erkennen ist geben die Cluster- und Tiefen-Auffälligkeitskarte am besten ein reales Objekte wieder. Von daher werden diese beiden im Gesamtsystem stärker gewichtet. Allerdings sollten, wenn eine aufgabenspezifische Objektannahme getätigt werden kann, die Gewichte idealerweise durch ein *top-down* Prozess eingestellt werden.

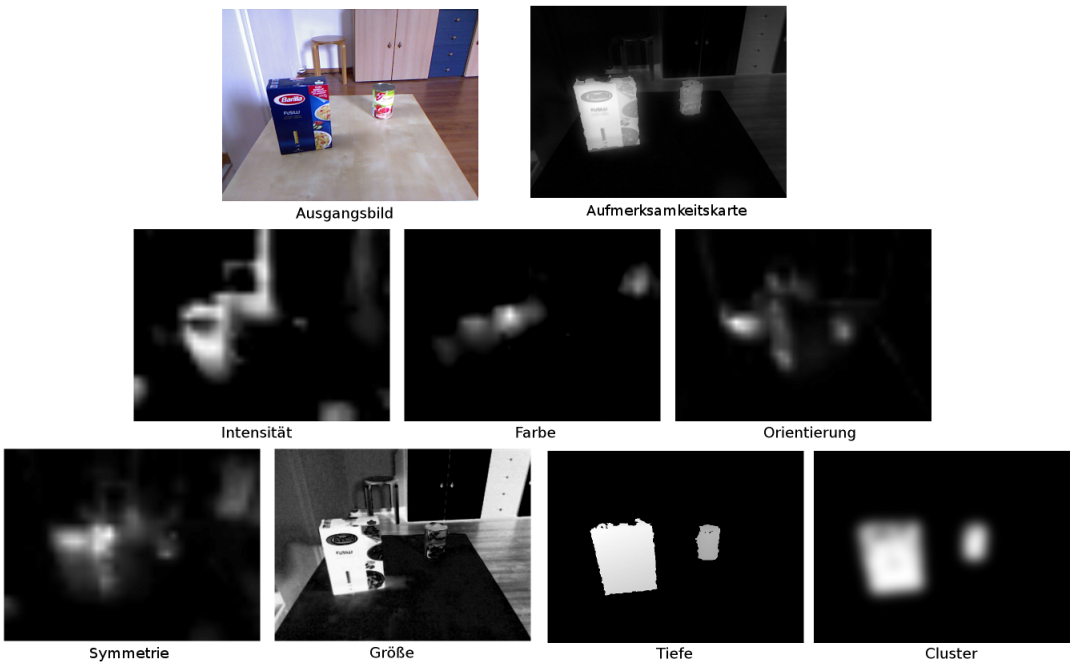


Abbildung 5.4: Aufmerksamkeitskarte zerlegt in ihre einzelnen Auffälligskeitskarten.

5.2.3 Lokalisierung des nächsten Explorationsortes

Das Ziel ist anhand der erstellten Aufmerksamkeitskarte den nächsten Explorationsort für die Erstellung einer Objekthypothese zu finden. Die Auswahl des nächsten Ortes richtet nach dem höchsten Wert in der Aufmerksamkeitskarte. Zum Finden wird in [IKN98] ein *winner-take-all* Neuronales Netz verwendet. Dieses ordnet jedem Bildpixel ein *integrate-and-fire* Neuron zu, das gedanklich als elektrische Teilchen aufgefasst werden kann. Der Anstieg der jeweiligen Potentiale richtet sich an den zugeordneten Werten in der Aufmerksamkeitskarte. Übersteigt das Potential einen gewissen Schwellwert feuert bzw. entlädt sich das Teilchen/Neuron auf der ganzen Karte. Die Position des zuerst entladenen Neurons ist der Gewinner und entspricht dem Ort mit der höchsten Aufmerksamkeit. Die Hemmung der Neuronen erfolgt durch die Gewichtung der Potentiale. Diese werden in einem festen Umkreis vom Gewinner-Neuron für eine voreingestellte Zeit schwächer gewichtet.

Der Grund für die Verwendung eines *winner-take-all* Neuronales-Netzes in der Arbeit von [IKN98], liegt an der nahen Verwandtschaft zu der Arbeitsweise der menschlichen Neuronen im Gehirn. Im vorliegenden Fall muss allerdings kein exakt neurobiologisches Modell nachgebaut werden, sondern nur dessen Verhalten imitiert werden. Zudem ist die zeitlich begrenzte und feste Größe der Hemmung für das System eher nachteilig, weil ein einmal besuchtes Objekt nicht nochmal untersucht werden soll. Daher

muss die Hemmung für eine Szene immer fortbestehen und am besten für die ganze Objektregion gelten. Um dieses zu gewährleisten wird eine binäre Hemmungskarte $\mathcal{H} \in [1, 0]$ erstellt, worin alle ehemaligen Explorationsorte verzeichnet werden und die gehemmte Region dynamisch anhand der Objekthypothese erfolgt, siehe dazu die späteren Kapitel 5.4.3 und 5.5.2. Die Aufmerksamkeitskarte wird dann mit der Hemmungskarte im jedem Explorationszyklus maskiert mit:

$$\bar{\mathcal{S}}^{\star}(x, y) = \begin{cases} \mathcal{S}^{\star}(x, y), & \text{wenn } \mathcal{H}(x, y) = 1, \\ 0, & \text{sonst.} \end{cases} \tag{5.20}$$

Das Auffinden des nächsten Explorationsortes $\mathcal{E}$ funktioniert anschließend über eine normale Maximumsuche innerhalb der gehemmten Aufmerksamkeitskarte $\bar{\mathcal{S}}^{\star}$:

$$\mathcal{E} = \underset{x \in [1..W], y \in [1..H]}{\arg\max} (\bar{\mathcal{S}}^{\star}(x, y)) \tag{5.21}$$

Wobei W die Breite und H die Höhe der Aufmerksamkeitskarte ist. Da immer der höchste Wert lokalisiert und nach jedem Zyklus maskiert wird, sinkt die Aufmerksamkeit mit der Zeit ab. Unterschreitet der Wert von $\mathcal{E}$ einen festgelegten Schwellwert (hier $T = 0.6$), geht das System davon aus, dass sich keine interessanten Objekte mehr in der Szene befinden und folglich die Exploration beendet ist. Alternativ könnte ein mobiler Roboter einen entfernteren Ort mit relativ hohen Aufmerksamkeitswerten anfahren und am Zielort den Explorationszyklus neu starten, indem die Hemmungskarte zurückgesetzt wird.

5.3 Erzeugung einer Objekthypothese

Im letzten Kapitel wurde das Auffinden eines nächsten Explorationsortes erläutert. Dabei korrespondiert die Position des Explorationsortes mit einem Punkt im RGB-Bild bzw. dem Tiefenbild der Kamera. Ein Punkt hat bekanntlich keine Ausdehnung und kann somit nicht ein zusammenhängendes physikalisches Objekt charakterisieren. Im Folgenden wird daher eine Objekthypothese erzeugt, die möglichst die komplette Ausdehnung eines Objektes im Bild beschreibt.

5.3.1 Pre-Segmentierung durch Expansion

Die Beschreibung der Ausdehnung erfolgt durch eine binäre Segmentierung, dass heißt alle Pixel die Teil der Objekthypothese H sind werden wie folgt markiert:

$$H(x,y) = \begin{cases} 1 & \text{Potentieller Pixel zu } \textit{einer} \text{ Objekthypothese} \\ 0 & \text{Hintergrundpixel} \end{cases} \tag{5.22}$$

Dabei ist zu beachten, dass hierbei nicht alle realen Objekte in der Szene segmentiert werden sollen, sondern nur die Pixel die dem realen Objekt am Explorationsort zugehörig sind.

Folglich wird unter der Annahme, dass der höchste Wert in der Aufmerksamkeitskarte zu einem realen Objekt gehört, der Explorationsort erstmal expandiert. Dazu wird auf einem normierten Tiefenbild $Z \in [0..2048]$ ein *flood-fill* Algorithmus ausgeführt. Ausgehend vom Explorationsort testet und markiert dieser jeweils iterativ alle 8 Nachbarpixel $Z(x,y)$ so lange bis die folgende Bedingung nicht mehr erfüllt ist:

$$Z(x',y') - \Delta \leq Z(x,y) \leq Z(x',y') + \Delta \tag{5.23}$$

Wobei $Z(x',y')$ der zuletzt markierte Pixel ist und der Parameter Δ die maximale zulässige Differenz zwischen zwei Nachbarpixeln angibt. Hierbei wurde der Parameter auf einen relativ kleinen Wert von 5 gesetzt. Allerdings kann es durch die sukzessive Ausbreitung der Pixel passieren, dass sich der Algorithmus auf die Untergrundfläche des potentielles Objektes ausbreitet. Um dieses zu verhindern, werden vorher alle Pixel im Tiefenbild die zur Konsensmenge der planaren Hauptfläche gehören (siehe Kapitel 4.4) auf Null gesetzt. Dadurch wird erreicht, dass die Differenz zur Untergrundfläche auf jedenfall größer ist als 5 und somit die Ausbreitung an diesen Stellen stoppt.

Der Vorteil der Segmentierung anhand der Tiefe ist das texturierte Objekte den Algorithmus nicht beeinflussen. Allerdings werden Objekte die große Löcher aufweisen oder wo Tiefeninformationen aufgrund von Reflexionen fehlen nicht ganz erfasst, siehe Abbildung 5.5c.

5.3.2 Verfeinerung der Segmentierung

Für die Verbesserung der Segmentierung und um die erwähnten Nachteile des *flood-fill* Algorithmus auszugleichen wird anschließend eine *GrabCut*-Segmentierung eingesetzt, siehe [RKB04]. Im wesentlichen basiert dieser auf einer iterativen Version des *Graph-Cut* Algorithmus von [BJ01] der zusätzlich noch für die Verwendung von Farbbildern angepasst wurde.

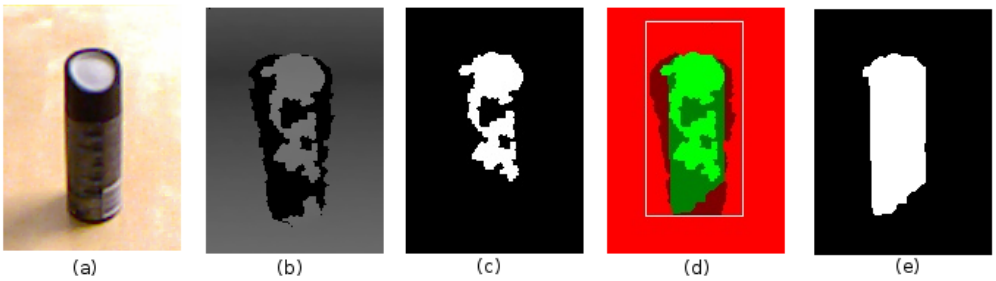


Abbildung 5.5: (a) RGB-Bild, (b) Tiefenbild, (c) *flood-fill* Segmentierung (d) *Grab-Cut* Segmentierung und (e) die resultierende Objekthypothese.

Für die optimale Segmentierung wird zunächst eine Kostenfunktion E aufgestellt, bestehend aus den gesamten Regioninformationen U und den Kanteninformationen V des RGB-Bildes I:

$$E = \sum_{i \in I} [U_{\text{fgd}}(x_i, S) + U_{\text{bgd}}(x_i, T)] + \gamma \sum_{i,j \in C} [h_i \neq h_j] V(x_i, x_j) \tag{5.24}$$

wobei C die Menge an Nachbarpixelpaaren ist. Allerdings werden nur Pixelpaare an den Segmentgrenzen aufsummiert. Das sind Pixel die zwischen zwei Regionen liegen und demnach in der Indikatorfunktion H unterschiedlich sind, vgl. Formel 5.22. Der Parameter γ ist dagegen ein konstanter Gewichtsfaktor, der bei kleinen Werten im Endergebnis zu geglätteten Kanten führt. Kodiert wird das Ganze über einen gewichteten Graphen, wobei jeder Pixel x_i im RGB-Bild einen Knoten n_i repräsentiert. Zusätzlich existieren im Graphen zwei spezielle Knoten: Ein *source*-Knoten S für die Repräsentation der gesamten potentiellen Objektpixel im Bild und ein *sink*-Knoten T für die restlichen Hintergrundpixel, siehe Abbildung 5.6.

Die Kantengewichte zwischen zwei Nachbarknoten mit den Pixel x_i und x_j werden einmalig durch die Kanteninformation bestimmt, indem der euklidische Abstand $||.||$ im RGB-Farbraum berechnet wird:

$$V(x_i, x_j) = \text{dist}(i, j)^{-1} \exp\left(-\beta ||x_i - x_j||^2\right) \tag{5.25}$$

wobei $\text{dist}(i, j)$ die örtliche euklidische Distanz der beiden Nachbarpixel ist und β entweder eine Konstante oder eine Erwartung über das ganze Bild darstellt. Das hat zur Folge, dass eine Kante zwischen zwei Pixeln (ein hoher Abstand im Farbraum) zu einem kleinen Kantengewicht im Graphen führt. Hingegen bestimmen die Kantengewichte zwischen den Pixelknoten n_i bzw. x_i und den beiden speziellen Knoten S und T die Regioninformationen mit:

$$U_{\text{fgd}}(x_i, S) = -\log(\text{GMM}_{\text{fgd}}(x_i)) \tag{5.26}$$

$$U_{\text{bgd}}(x_i, T) = -\log(\text{GMM}_{\text{bgd}}(x_i)) \tag{5.27}$$

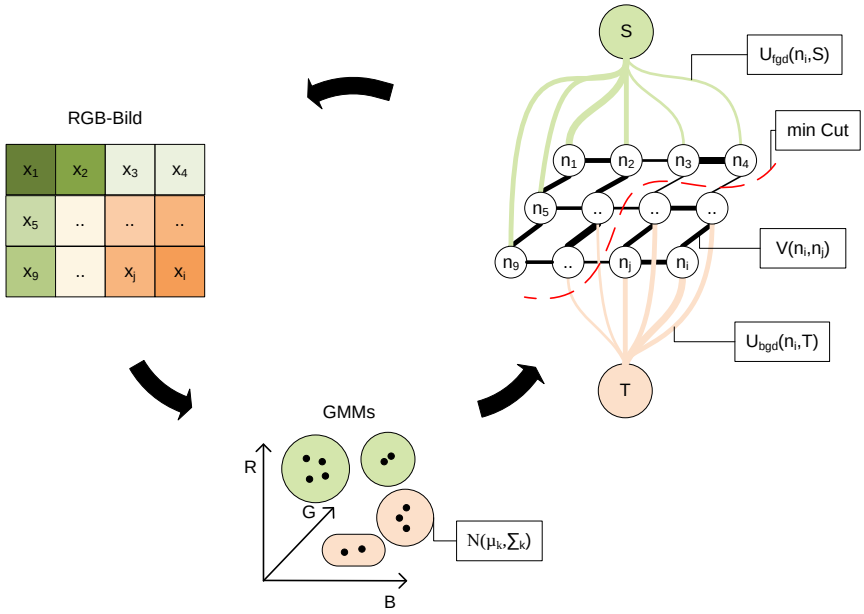


Abbildung 5.6: Schematische Darstellung einer *GrabCut*-Segmentierung, wobei die Linienstärken proportional zu den Kantengewichten eingezeichnet sind.

Im *GrabCut*-Verfahren werden die Regionen durch zwei *Gaussian Mixture Models* (GMMs) modelliert: Ein Vordergrundmodell GMM_{fgd} für die potentiellen Objektpixel und ein Hintergrundmodell GMM_{bgd} für die Hintergrundpixel. Dabei besteht das Modell GMM_{fgd} (GMM_{bgd}) aus einer gewichteten Summe von K Gauß-Verteilungen, die jeweils durch ein Gewicht π_k ($\hat{\pi}_k$), den Erwartungswert μ_k ($\hat{\mu}_k$) und der Kovarianzmatrix Σ_k ($\hat{\Sigma}_k$) beschrieben wird:

$$\text{GMM}_{\text{fgd}}(x_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \quad \forall i : x_i \in \mathcal{X}_{\text{fgd}} \tag{5.28}$$

$$\text{GMM}_{\text{bgd}}(x_i) = \sum_{k=1}^{K} \hat{\pi}_k \mathcal{N}(x_i|\hat{\mu}_k, \hat{\Sigma}_k) \quad \forall i : x_i \in \mathcal{X}_{\text{bgd}} \tag{5.29}$$

Für die Ermittlung der unbekannten Parameter der GMMs wird ein *Expectation-Maximization*-Algorithmus angewendet. Damit dieser jedoch arbeiten kann, müssen einige Pixel im RGB-Bild in dem Initialisierungsschritt manuell zum Vordergrund- bzw. dem Hintergrundmodell zugeordnet werden. Für das Vordergrundmodell werden hier die markierten Pixel des *flood-fill* Algorithmus als $\mathcal{X}_{\text{fgd}}$ herangezogen, in Abbildung 5.5d als hellgrüne Pixel dargestellt. Von dieser Menge wird zudem ein umschließendes Rechteck (engl. bounding box) berechnet und auf das 1,5 fache vergrößert. Alle Pixel die jetzt außerhalb des Rechteckes liegen werden der Menge $\mathcal{X}_{\text{bgd}}$ des Hintergrundmodells zugeordnet,

sowie alle Pixel die zur Konsensmenge der planaren Hauptfläche gehören und Pixel die eine größere Tiefeninformation aufweisen als $1m$. In Abbildung 5.5d sind alle manuell zugewiesenen Pixel, die zum Hintergrundmodell gehören, hellrot dargestellt.

Darauf folgend wird der gewichtete Graph durch einen Schnitt, bezüglich der Minimierung der Kostenfunktion E, in zwei separate Teile getrennt. Alle Pixel bei denen die Knoten nach dem Schnitt noch mit dem S-Knoten (T-Knoten) verbunden sind, definieren die neue $\mathcal{X}_{\text{fgd}}$ ($\mathcal{X}_{\text{bgd}}$) Menge. Aus den erhaltenen neuen Mengen werden wieder die beiden GMMs modelliert, die erneut die Gewichte des Graphen verändern. Ausgehend von dem neu gewichteten Graphen wird iterativ wieder der minimale Schnitt gesucht. Dies geschieht so lange bis entweder die Kostenfunktion konvergiert oder wie in dieser Arbeit der Vorgang durch eine feste Anzahl $N = 3$ an Iterationsschritten abgebrochen wird. Die resultierende Menge $\mathcal{X}_{\text{fgd}}$ ist letztendlich die Beschreibung der finalen Objekthypothese, siehe Abbildung 5.5e.

5.4 Vergleich der Hypothese mit memorierten Objekten

Das Vergleichen der Objekthypothese mit allen memorierten Objekten ist optional, hat aber den Zweck die Hypothese gegebenenfalls frühzeitig zu verifizieren und somit eine erneute Untersuchung von bereits explorierten Objekten zu verhindern. Zum Wiedererkennen der memorierten Objekte wird in dieser Arbeit eine skalierungsinvariante Merkmalstransformation (engl. scale-invariant feature transform, kurz SIFT) von [Low04] eingesetzt. Die grundlegende Idee des verwendeten SIFT-Algorithmus besteht in der robusten und eindeutigem Auffindung von Merkmalen in Bildern, die eine relativ zuverlässige Identifizierung von Objekten erlauben. Der Vorteil dieser Merkmale ist es, dass sie invariant gegenüber der teilweisen Verdeckung der Objekte, affinen Transformationen, Rauschen und den Lichtverhältnissen sind. Nachfolgend wird der Ablauf in drei Schritten erläutert, wie die Hypothese mit den memorierten Objekten mit Hilfe des SIFT-Algorithmus verglichen wird.

5.4.1 Detektierung markanter Merkmale

Der erste Schritt ist es markante Merkmale in der Objekthypothese zu detektieren. Dazu wird als Erstes im RGB-Bild die Objekthypothese herausgeschnitten, indem das Bild auf das kleinste umschließende Rechteck[2] der potentiellen Objektpixel verkleinert wird. Anschließend wird eine Gauß- und eine *Difference of Gaussian*-Pyramide (DoG-Pyramide) aufgebaut, um Merkmale der Objekthypothese zu erhalten die unabhängig von der Skalierung des Bildes sind, siehe Abbildung 5.7. Die Gauß-Pyramide

[2] Das umschließende Rechteck wird in diesem Fall meist auch als „*region of interest*“ oder kurz ROI bezeichnet.

wird ähnlich wie im Kapitel 5.2.1 durch sukzessive Anwendung eines Gaußfilters aufgebaut. Allerdings stellt diesmal eine Ebene in der Gauß-Pyramide eine ganze Oktave von verschiedenen gauß-geglätteten aber nicht in der Ebenengröße runterskalierten Ebenen dar. Erst nach mindestens vier Gauß-Ebenen wird die Ebenengröße um die Hälfte runterskaliert. Nachdem die Gauß-Pyramide erzeugt wurde, werden jeweils die benachbarte Gauß-Ebenen $G(x, y, k\sigma)$ und $G(x, y, \sigma)$ aller Oktaven zu einzelnen DoG-Ebenen $L(x, y, \sigma)$ subtrahiert:

$$L(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma) \tag{5.30}$$

Als Ergebnis erhält man eine DoG-Pyramide, was auch als eine approximierte Version einer Laplace-Pyramide aufgefasst werden kann.

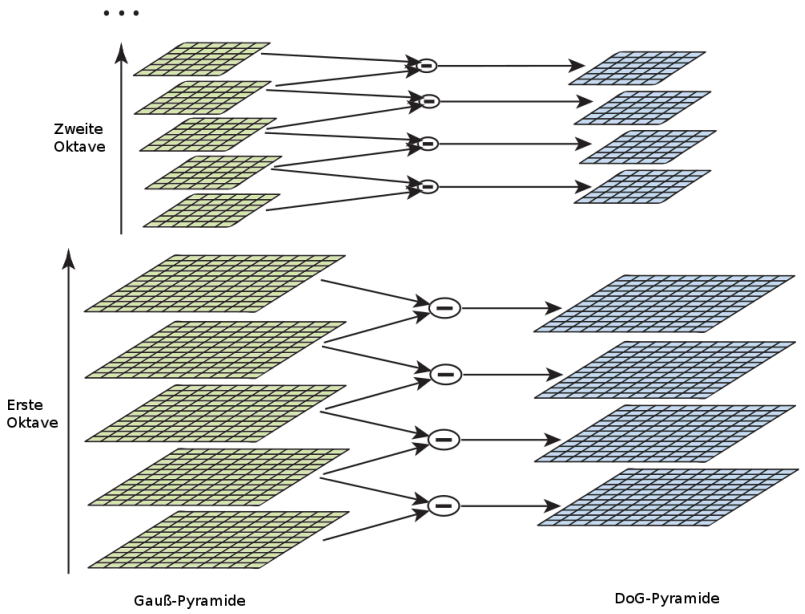


Abbildung 5.7: Links die Gauß-Pyramide unterteilt in einzelne Oktaven und rechts die durch Differenzbildung der Gauß-Ebenen entstandene DoG-Pyramide, geändert aus [Low04].

Anschließend wird jeder Pixel sukzessiv in der DoG-Pyramide auf lokale Extrema hin untersucht. Ein Maximum oder ein Minimum ist gefunden, wenn der Pixel im Vergleich zu allen seinen Nachbarpixeln größer oder kleiner ist. Zum Vergleich werden nicht nur die 8 Nachbarpixel der aktuelle Ebene mit einbezogen, sondern auch die jeweiligen 9 Nachbarpixel der niedrigeren und der höheren Ebene der DoG-Pyramide. Die gewonnen lokalen Extrema müssen aber nicht die wahren Extrema sein. Das Problem liegt an der Diskretisierung des Bildes, da zwischen zwei diskreten Pixelwerten noch höhere bzw. niedrige Extrema liegen könnten. In [Low04] werden daher die Extrema weiter verbessert, indem

diese mit einer Taylorreihe zweiten Gerades am Entwicklungspunkt $\vec{x}_E = (x_E, y_E, \sigma_E)^T$ mit folgender Formel approximiert werden:

$$L(\vec{x}) = L(\vec{x}_E) + \frac{\partial L(\vec{x}_E)^T}{\partial \vec{x}}(\vec{x} - \vec{x}_E) + \frac{1}{2}(\vec{x} - \vec{x}_E)^T \frac{\partial^2 L(\vec{x}_E)^T}{\partial \vec{x}^2}(\vec{x} - \vec{x}_E) \tag{5.31}$$

wobei $\vec{x} = (x, y, \sigma)^T$ ist. Durch Ableiten der Formel 5.31 und dem Null setzen, erfolgt die Lokalisation des neuen Extremas über den Offset $\bar{x}$ mit:

$$\bar{x} = -\frac{\partial^2 L(\vec{x}_E)}{\partial \vec{x}^2}^{-1} \frac{\partial L(\vec{x}_E)}{\partial \vec{x}} \tag{5.32}$$

Die daraus verbesserten Menge an Extrema bildet eine potentielle Menge an Merkmalen. Allerdings ist die Menge noch viel zu groß und wird deshalb anhand von zwei Kriterien weiter gefiltert. Das erste Kriterium ist der Kontrast. Merkmale die ein geringen Kontrast aufweisen werden herausgefiltert, weil diese besonders empfindlich gegen Rauschen und Störungen sind. Zur Bestimmung des Kontrastes wird die Formel 5.32 in 5.31 eingesetzt. Als Ergebnis erhält man folgende Formel:

$$L(\bar{x}) = L(\vec{x}_E) + \frac{1}{2}\frac{\partial L(\vec{x}_E)}{\partial \vec{x}}\bar{x} \tag{5.33}$$

Unter der Annahme das der Wertebereich der Pixel zwischen $[0..1]$ liegen, werden alle Merkmale die kleiner sind als $|L(\bar{x})| < 0.03$ verworfen.

Als zweites Kriterium dient die Krümmung am Merkmalsort. Dabei werden alle Merkmale rausgefiltert die auf Linien liegen. Bei geringen Änderungen eines potentiellen Objektes im Bild, verschiebt sich das Merkmal häufig entlang der Linie und kann daher nicht eindeutig wiedererkannt werden, weil diese oftmals die gleichen Merkmalseigenschaften besitzen. Dagegen sind Eckpunkte gut geeignet und sollten möglichst beibehalten werden. Für die Unterscheidung ob das Merkmal ein Eckpunkt oder auf einer Linie liegt wird eine 2x2 große Hessenmatrix $\bar{H}$ erstellt, da die Eigenwerte sich proportional zu der Krümmung verhalten. Allerdings müssen die Eigenwerte nicht explizit berechnet werden, da nur das Verhältnis der senkrechten und waagerechten Hauptkrümmung gebraucht wird. Zur Beurteilung reicht daher die Spur und die Determinante der Hessematrix aus, was zur folgenden Kriteriumsfunktion führt:

$$\frac{\text{spur}(\bar{H})^2}{\det(\bar{H})} < \frac{(r+1)^2}{r} \tag{5.34}$$

wobei r ein Schwellwert ist, der in [Low04] durch eine experimentelle Bestimmung auf 10 gesetzt wurde.

5.4.2 Erstellung von Merkmalsdeskriptoren

Aus der Menge an detektierten und gefilterten Merkmalen wird jetzt für jedes Element ein Merkmaldeskriptor erstellt. Das Ziel der Merkmalsdeskriptoren ist es für jedes Merkmal eine möglichst robuste und eindeutige Beschreibung zu haben, so dass eventuell zugehörige Merkmaldeskriptoren in den memorierten Objekten schnell wiedergefunden werden.

Die Erstellung erfolgt dabei immer relativ zu einer Orientierung, um eine Invarianz bezüglich der Rotation zu gewährleisten. Für die Bestimmung der Orientierung werden alle Gradienten der lokalen Pixelumgebung des Merkmales benötigt. Da die Gradientenberechnung nicht auf der DoG-Ebene ausgeführt werden kann, findet die Berechnung auf der Gauß-Ebene $G(x, y, \sigma)$ statt. Die Auswahl der Gauß-Ebene erfolgt anhand der DoG-Ebene mit dem entsprechenden σ in dem das Merkmal gefundenen wurde. Jeder Gradient i wird anschließend mittels der beiden Formeln beschrieben:

$$m_i(x, y) = \sqrt{[G(x+1, y) - G(x-1, y)]^2 + [G(x, y+1) - G(x, y-1)]^2} \tag{5.35}$$

$$\theta_i(x, y) = \tan^{-1}\left(\frac{G(x, y+1) - G(x, y-1)}{G(x+1, y) - G(x-1, y)}\right) \tag{5.36}$$

wobei θ die Richtung und m die Länge bzw. die Stärke des Anstieges eines Merkmals ist. Aus den beiden Informationen jedes Gradienten wird ein Orientierungshistogramm gebildet. Die Klassenbreite des Histogramms beträgt konstant $10°$. Folglich existieren insgesamt 36 Klasseneinteilungen für die Gradientenrichtungen, um den vollen Bereich von $360°$ abzudecken. Die Klassenhäufigkeit entsteht durch das Aufaddieren aller Gradientenstärken, wobei diese zusätzlich durch den Abstand zum Merkmal gewichtet werden. Die Gewichtung erfolgt mit einem Gauß-Filter, so dass entferntere Gradienten geringer zum Orientierungshistogramm beitragen. Im erzeugten Histogramm wird dann die maximale Gradientenstärke m_{max} bestimmt. Schließlich werden alle Orientierungen zu einem Merkmal ermittelt, die einen Gradientenstärke von über 80% bezüglich m_{max} haben. Dadurch erhalten Merkmalsorte die verschiedene lokale Orientierungsmaxima besitzen, auch mehrere Merkmalsdeskriptoren und das führt wiederum zu einer Stabilitätssteigerung.

Mit Hilfe der gefundenen Orientierungen lassen sich jetzt die eigentlichen Merkmalsdeskriptoren erzeugen. Dazu werden wieder die Gradienten einer lokalen 16x16 großen Pixelumgebung berechnet, wobei diesmal die Umgebung jeweils relativ zu den ermittelten Orientierungen ausgerichtet ist. Positionen, die aufgrund der Rotation, nicht exakt mit einem Pixel des Bildes übereinstimmen, werden interpoliert. Anschließend wird die Umgebung in 4x4 gleichgroße Unterbereiche eingeteilt, woraus wiederum einzelne Orientierungshistogramme erstellt werden. Die Berechnung der Histogramme erfolgt nach dem gleichen Prinzip wie zur Bestimmung der Hauptorientierung. Allerdings beträgt diesmal

die Klassenbreite $45°$ (insgesamt 8 Klassen). Demnach hat ein Merkmalsdeskriptor eine Dimension von 4x4x8=128.

Das Vorgehen zur Erstellung der Merkmalsdeskriptors zeigt nochmal die Abbildung 5.8 anhand einer 8x8 Umgebung. Im linken Bild sind die einzelnen Orientierungen dargestellt, die aus den Gradientenrichtungen und -stäken berechnet werden. Der blaue Kreis steht symbolisch für die Gewichtung der Gradientenstärken. Durch das Addieren der Gradientenstärken in den 2x2 gleichgroßen Regionen, entstehen die dazugehörigen Orientierungshistogramme, die wiederum im rechten Bild der Abbildung 5.8 dargestellt sind.

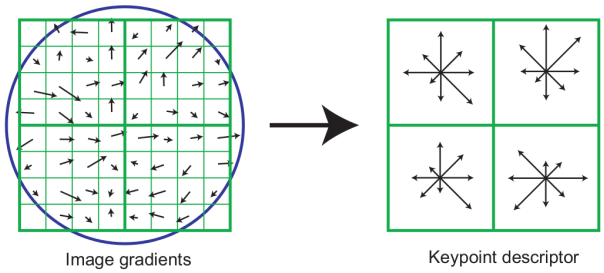


Abbildung 5.8: Erstellung eines Merkmalsdeskriptors am Beispiel einer 8x8 Umgebung. Links die einzelnen Orientierungen und rechts die dazugehörigen 2x2 Orientierungshistogramme. [Low04]

Eine Invarianz gegenüber Veränderungen zu linearen Lichtverhältnisse lässt sich relativ leicht erreichen, da sie nur auf die Gradientenstärke Einfluss nimmt. Die Normalisierung des Merkmalsdeskriptors auf einen Wertebreich von $[0..1]$ kann diese Beeinflussung neutralisieren. Schwieriger ist es jedoch bei nicht linearen Beleuchtungsschwankungen, die durch Schattenwürfe oder inhomogener Beleuchtung entstehen. Diese verursachen lokal sehr große Veränderungen in der Gradientenstärke aber nur kleine in der Orientierung. Zur Reduzierung dieses Effektes werden alle Werte im normalisierten Merkmalsdeskriptor auf 0,2 begrenzt und anschließend wieder neu normalisiert. Dies hat zur Folge, dass die Orientierungen ein stärkeres Gewicht im Merkmalsdeskriptor einnehmen, als die einzelnen Gradientenstärken.

5.4.3 Findung von Korrespondenzpaaren

Der berechnete Merkmalsdeskriptor der Objekthypothese wird jetzt mit allen Merkmalsdeskriptoren der memorierten Objekte verglichen, um die Hypothese zu verifizieren. Dafür sind mindestens vier korrekt korrespondierende Merkmalsdeskriptoren aus der Objekthypothese, mit den Merkmalsdeskriptoren eines memorierten Objektes erforderlich.

Der erste Ansatz zum Auffinden der Korrespondenzpaare ist, dass der nächste Nachbar mit Hilfe des euklidischen Abstandes naiv gesucht wird und anhand der Entfernung und einem Schwellwert die Korrespondenzzugehörigkeit ermittelt wird. Das Problem ist aber das bei dieser Methode oftmals zu viele falsche Korrespondenzpaare gefunden werden. Ein besseres Ergebnis konnte in [Low04] erzielt werden, wenn das Verhältnis zwischen den zwei besten Nachbarn betrachtet wird. Weicht das Verhältnis zu stark ab, wird das Korrespondenzpaar herausgefiltert. Das zweite meist größere Problem ist das schnelle Auffinden der beiden nächsten Nachbarn, da einerseits die Dimension des Merkmalsraums sehr hoch ist (hier $\mathbb{R}^{128}$) und andererseits eventuell mit vielen memorierten Objekten verglichen werden muss. Das Problem kann nicht vollständig gelöst werden und ist zurzeit noch ein sehr aktives Forschungsfeld. Allerdings wird in dieser Arbeit das Auffinden der besten zwei Nachbarn mittels einer approximierten k-Nächste-Nachbar Suche durchgeführt, siehe [ML09]. Der Algorithmus kann zwar nicht garantieren, dass die zwei besten nächsten Nachbarn gefunden werden aber dafür wird die Suche wesentlich beschleunigt.

Nachfolgend werden alle gefundenen Korrespondenzpaare nochmal über eine projektive Transformation kontrolliert. Dies erfolgt Aufgrund der Erkenntnis, dass einige Korrespondenzpaare exakt die gleichen Merkmalsdeskriptoren haben aber insgesamt im Bild so verteilt liegen können, dass eine tatsächliche Korrespondenz ausgeschlossen werden kann. Die projektive Transformation erfolgt durch eine Homographie-Matrix. Für eine robuste Schätzung der Matrix wird hier wieder der RANSAC-Algorithmus verwendet, siehe Kapitel 4.4. Zur Bestimmung der Matrix ist es jedoch erforderlich, dass mindestens vier nicht kollineare Korrespondenzpaare vorhanden sind. Liegt das transformierte Merkmal zu weit vom anderen Merkmal entfernt, wird das Korrespondenzpaar herausgefiltert.

Bleiben nach allen Filterungen mindestens vier Korrespondenzpaare übrig, wird die Objekthypothese verifiziert. Zur Unterdrückung des Explorationsortes wird die binäre Objekthypothese mittels Dilatation um 30px vergrößert und mit der binären Hemmungskarte disjunktiv verknüpft. Die Vergrößerung hat den Hintergrund, dass die Ränder eines echten Objektes manchmal nicht komplett von der Objekthypothese unterdrückt werden. Kamen weniger als vier oder keine Korrespondenzpaare zustande, wird versucht die Objekthypothese per Interaktion zu verifizieren.

5.5 Interaktion zur Verifizierung der Hypothese

Die Verifizierung der Hypothese per Interaktion, geschieht anhand einer versuchsweisen Verschiebung der Objekthypothese. Die bei erfolgreicher Verifizierung zugleich die Beweglichkeit eines Objektes

ermittelt. Dazu werden in den beiden nachfolgenden Unterkapiteln zuerst eine Verschiebungstrajektorie bestimmt und anschließend die visuellen und taktilen Informationen der Verschiebung analysiert.

5.5.1 Bestimmung der Verschiebungstrajektorie

Der Anfahrtspunkt, die Orientierung und der Bewegungspfad der Roboterhand (Endeffektor) beschreiben die Verschiebungstrajektorie, die es für jede Objekthypothese zu bestimmen gilt. Für ein stabiles Verschieben von echten Objekten, ist es erforderlich, dass insbesondere der Anfahrtspunkt mit Bedacht ausgesucht wird. Befindet sich der Anfahrtspunkt zum Beispiel oberhalb des Objektschwerpunktes, kann das Objekt umfallen. Ist der Anfahrtspunkt zu weit am Rand, streift die Roboterhand nur das Objekt und es kommt keine eindeutige Bewegung zustande. Optimaler Weise befindet sich daher der Anfahrtspunkt unterhalb des Objektschwerpunktes. In der Objekthypothese $H(x, y)$ wird dazu der Bildschwerpunkt $\bar{p} = (\bar{x}, \bar{y})$ bestimmt. Die Berechnung erfolgt hier über die geometrischen Momente $m_{p,q}$:

$$m_{p,q} = \sum_{x=1}^{M} \sum_{y=1}^{N} x^p y^q H(x, y) \tag{5.37}$$

mit

$$\bar{x} = \frac{m_{1,0}}{m_{0,0}} \qquad \bar{y} = \frac{m_{0,1}}{m_{0,0}} \tag{5.38}$$

Durch Subtrahieren der $\bar{y}$-Koordinate mit $\frac{1}{8}$ der Höhe des umschließenden Rechtecks der Objekthypothese, wird der Bildschwerpunkt herabgesetzt. Der Anfahrtspunkt ist dann der herabgesetzte Schwerpunkt, wobei dieser vorher noch mit Hilfe der gefundenen Projektionsmatrix in die korrespondierende Weltkoordinate umgewandelt wird.

Zum Verschieben wird allein der Mittelfinger verwendet. Damit die anderen beiden Finger nicht im Weg stehen, werden diese im gesamten Verlauf eingeklappt. Die Orientierung des Mittelfingers richtet sich an dessen Fingerkuppe, da sich dort die Drucksensoren befinden. Damit die Drucksensoren stets Kontakt zum potentiellen Objekt haben, wird die Fläche der Fingerkuppe orthogonal zur z-Weltkoordinate ausgerichtet. In Abbildung 5.9 ist der Anfahrtspunkt schematisch im Simulator dargestellt und annotiert. Das blaue Koordinatensystem zeigt das Weltkoordinatensystem, wohingegen das grüne Koordinantensystem die Flächenkoordinatensystem der Fingerkuppe repräsentiert.

Der Bewegungspfad der Verschiebungstrajektorie fängt von der Grundstellung des Roboterarms an. Die Grundstellung ist hierbei die Stellung worin alle Gelenkwinkel des Roboterarms gleich Null sind. Anschließend fährt der Roboterarm mit eingeschalteter Kollisionserkennung auf bis zu 4cm vor

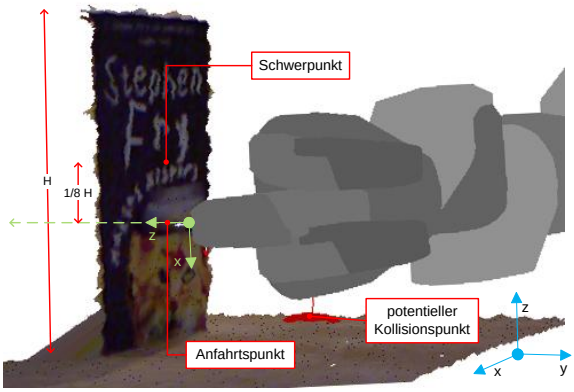


Abbildung 5.9: Annotierte Darstellung eines Anfahrtspunktes im Simulator. Blaues Koordinatensystem entspricht dem Weltkoordiantensystem und Grün dem Flächenkoordinatensystem der Fingerkuppe.

dem ermittelten Anfahrtspunkt, wobei die Fingerkuppe wie beschrieben zum Schluss der Trajektorie nach vorn ausgerichtet ist. Danach wird der Roboterarm langsam 10cm nach vorne und zurück bewegt. In Abbildung 5.9 als grüne, gestrichelte Linie dargestellt. Hierfür ist es notwendig, dass die Kollisionserkennung ausgeschaltet ist, um einen möglichen Kontakt zu erlauben. Damit der Roboterarm keinen Schaden nimmt und für die spätere Analyse, wird während der ganzen Vorwärtsbewegung fortlaufend der Drucksensor ausgewertet. Übersteigt der Druck einen festgelegten Schwellwert wird die Vorwärtsbewegung sofort abgebrochen und zurückgefahren. Der gesamte Bewegungspfad endet, indem der Roboterarm (wieder mit eingeschalteter Kollisionserkennung) zurück in die Grundstellung gefahren ist.

5.5.2 Visuelle und taktile Analyse

Zur Verifikation der Objekthypothese und der gleichzeitigen Bestimmung der Beweglichkeit eines verifizierten Objektes werden die visuellen und taktilen Informationen in drei verschiedenen Fällen ausgewertet.

Der erste Fall ist, dass eine visuelle Bewegung am Explorationsort wahrnehmbar ist. Dies stellt ein deutliches Anzeichen eines bewegbaren Objektes dar, womit die Objekthypothese verifiziert werden kann. Hierbei muss keine taktile Auswertung erfolgen, da die Bewegung selbst eine taktile Voraussetzung impliziert. Für die visuelle Bewegungsanalyse werden die Cluster oberhalb der planaren Hauptfläche einmal vor und nach der Interaktion als Punktwolke aufgenommen und gegeneinander subtrahiert,

um die Differenz der beiden Punktwolken zu erhalten. Da die Punkte bei mehreren Aufnahmen nicht stabil am Ort bleiben, wird die Subtraktion mittels des nächsten Nachbarn durchgeführt. Diesbezüglich wird zu jedem Punkt der nächste Nachbar, der den minimalen euklidischen Abstand in der anderen Punktwolke aufweist gesucht. Übersteigt der Abstand einen Schwellwert wird der Punkt als Teil einer Bewegung aufgefasst. Nachdem eine Differenzpunktwolke entstanden ist, wird diese, mit dem Algorithmus 1 (S. 34), in einzelne Cluster unterteilt. Dieser sorgt zum Einen dafür, dass nur Clustergrößen mit $s_{min} = 50$ als Bewegung wahrgenommen werden, worin kleine meist als Störungen verursachten Bewegungen der zwei Punktwolken ignoriert werden. Zum Anderen wird das größte Cluster s_{max} für die spätere Hemmung gebraucht.

Der zweite Fall liegt vor, wenn keine visuelle Bewegung wahrgenommen wurde. Jedoch heißt dies nicht zwangsläufig das kein Objekt vorhanden ist. Vielmehr könnte ein Objekt zu schwer oder einfach nicht beweglich sein. Daher wird hier zusätzlich noch die taktile Information ausgewertet. Wird ein Druck am Mittelfinger der Roboterhand wahrgenommen, spricht dies für die Existenz eines Objektes am Explorationsort und die Objekthypothese wird verifiziert. Gleichzeitig wird hierbei dem Objekt die Eigenschaft zugewiesen das es nicht bewegbar ist, im Gegensatz zur visuellen Bewegungsanalyse.

Im letzten möglichen Fall konnten keinerlei taktilen oder visuellen Informationen ermittelt wurden. Hier wird die Objekthypothese verworfen, da der Roboterarm offenbar in den leeren Raum gegriffen hat.

In allen drei Fällen muss die Aufmerksamkeitskarte gehemmt werden, um weitere Explorationen durchführen zu können. In den letzten beiden Fällen geschieht die Hemmung, wie im Kapitel 5.4.3 beschrieben, durch das umschließende Rechteck der Objekthypothese. Für den ersten Fall kann jedoch die Objekthypothese nicht genutzt werden, da sich das Objekt wegbewegt hat und nicht mehr am Explorationsort liegt. Allerdings zeigt jetzt die größte Differenzwolke s_{max} die neue Position des Objektes an. Dazu werden alle Koordinaten der Differenzwolke auf die Hemmungskarte projiziert und wieder disjunktiv verknüpft. Das verwenden der größten Differenzwolke s_{max} hat den Vorteil, dass beim Verschieben auch nahe Objekte sich bei der Vorwärtsbewegung mitverschieben dürfen. Allein die größte Punktwolke wird gehemmt, was zur Verbesserung der Auflösung von Mehrdeutigkeiten von nahen Objekten beiträgt. Zum Schluss werden bei einer verifizierten Objekthypothese die Merkmalsdeskriptoren, der Ausschnitt des Objektes im RGB-Bild und dessen Beweglichkeit memoriert.

6 Experimentelle Evaluierung

Für die Evaluierung des Gesamtsystems wird auf eine statistische Auswertung verzichtet, da hierzu eine große Anzahl von automatisch ablaufenden Versuchsdurchgängen erforderlich sind. In realen Experimenten ist dies in angemessener Zeit nicht machbar, weil die Ergebnisse manuell verglichen werden müssten. Eine Alternative bestehe darin die Experimente automatisch im Simulator durchzuführen. Hier entsteht aber das Problem, dass die Physik exakt im Rechner simuliert werden müsste, was lediglich in Annäherung geschehen kann. Daher werden hier nur einige ausgewählte Experimente durchgeführt, die resultierenden Ergebnisse dargestellt und die Grenzen des Gesamtsystems aufgezeigt.

6.1 Verwendete Versuchsobjekte

Um möglichst eine Vielzahl an unterschiedlichen Objekteigenschaften bei den Experimenten abzudecken, wurden insgesamt 8 unterschiedliche Versuchsobjekte aus dem Alltag ausgewählt, siehe Abbildung 6.1. Jedes dieser Objekte steht repräsentativ für mindestens eine spezielle Objekteigenschaft. Die Nudel- und die Knäckebrotpackung sind relativ groß und stehen daher für alle großen Objekte, wohingegen die Figur benutzt wird, um kleine Objekte darzustellen. Der Klebestift hingegen repräsentiert alle instabilen Objekte, da der Klebstift keine stabile Standfläche hat und relativ leicht umfällt. Das Deodorant wurde ausgewählt, weil dieses eine glänzende Oberfläche hat und daher stark reflektierend ist. Das sorgt dafür, dass in viele Bereichen am Objekt keine Tiefeninformationen verfügbar sind, weil die ausgesendeten Infrarotstrahlen der Kinect-Kamera ungünstig abgelenkt werden. Daher steht das Deodorant für alle Objekte bei denen keine Tiefeninformationen ermittelbar sind. Die Tomatendose und die grüne Dose beschreiben runde Objekte, wobei die grüne Dose speziell alle Objekte mit homogenen Oberflächen charakterisiert. Das letzte Objekt, die Stierstatue, steht für alle Objekte mit stark texturierten Oberflächen und Objekte die keine primitiven Formen besitzen.

Abbildung 6.1: Verwendete Versuchsobjekte zum Evaluieren des Gesamtsystems.

6.2 Ablauf der Versuchsdurchführung

Aus den verwendeten 8 Versuchsobjekten wurden unterschiedliche Versuchskonstellationen zusammengestellt, um das Verhalten des Gesamtsystems zu analysieren. Insgesamt 20 Ergebnisse sind davon in der Tabelle 6.1 zusammengefasst, wobei die zugehörigen Versuchskonstellationen in Anhang A dargestellt sind. Vor dem Start eines jeden Experimentes wurden sämtliche memorierten Objekte in der Datenbank gelöscht. Eine Ausnahme bildet das Experiment 4, dort wurden eine vormals erstellte Objektdatenbank beibehalten, um den visuellen Vergleich mit den memorierten Objekten zu ermöglichen. Zusätzlich wurde nach jeder erfolgreichen verifizierten Objekthypothese der Anwender über eine Eingabemaske befragt, um was für ein Objekt es sich handelte. Diese semantische Zuweisung der Bedeutung soll aber in einer zukünftigen Version des Systems halb- bzw. vollautomatisch erfolgen. Desweiteren wurde zur Simulierung der nicht Beweglichkeit eines Objektes das betreffende Objekt während der Hypothesenverifikationsphase per Hand festgehalten. Ein Demonstrationsvideo, worin ein ganzer Ablauf gezeigt wird, ist in der Begleit-DVD enthalten.

6.3 Ergebnisse und Grenzen des Gesamtsystems

Das entwickelte Gesamtsystem erwies sich als sehr robust, wenn es um das Unterscheiden zwischen Beweglichkeit und nicht Beweglichkeit eines Objektes angeht, siehe Experimente 6,7 und 14. Außerdem konnte festgestellt werden, dass das System nur Orte untersucht, worin sich auch wirklich ein reales Objekt befand. Fehlgriffe ins Leere, aufgrund einer falschen Objekthypothese bzw. einer falschen Aufmerksamkeitskarte, traten bei den ganzen Experimenten zu keiner Zeit auf. Dies war auch dann nicht der Fall, wenn kein Objekt auf dem Tisch lag, wie im Experiment 12. Weiterhin konnte beobachtet werden, dass die letzten Explorationsorte immer erfolgreich komplett mit der

Nr.	Anzahl Objekte	Anfahrtsversuche	Verifizierte Hypothesen	Verworfene Hypothesen	Bemerkungen
1	4	4	4	0	—
2	3	3	3	0	—
3	3	3	3	0	Klebestift und Deodorant umgefallen.
4	3	1	3	0	Nudelpackung und Tomatendose in memorierten Objekten wiedererkannt.
5	2	2	2	0	Nudelpackung und Tomatendose direkt nebeneinander.
6	3	3	3	0	Nudelpackung und Tomatendose festgehalten.
7	3	3	3	0	Alle Objekte festgehalten.
8	1	1	1	0	Deodorant umgefallen.
9	3	2	2	1	Figurhypothese verworfen.
10	2	1	1	1	Figurhypothese verworfen.
11	1	1	1	0	—
12	0	0	0	0	leerer Tisch.
13	2	1	1	0	Figur auf Tomatendose gestellt.
14	1	1	1	0	Nudelpackung festgehalten.
15	2	1	1	0	grüne Dose auf Tomatendose gestapelt.
16	2	2	2	0	Deodorant auf grüne Dose gestapelt.
17	3	3	3	0	texturierter Hintergrund.
18	3	2	2	0	texturierter Hintergrund. Keine Figurhypothese erstellt.
19	3	2	2	1	texturierter Hintergrund. Klebestift umgefallen. Figurhypothese verworfen.
20	2	2	2	0	texturierter Hintergrund.

Tabelle 6.1: Ergebnisse der Experimente.

Hemmungskarte unterdrückt worden sind, da doppelte Anfahrtsversuche pro Objekt vermieden wurden. Bei den beiden Experimenten 5 und 16 zeigt sich auch die Leistungsfähigkeit des gesamten Systems bei Mehrdeutigkeiten. Dort wurden zwei Objekte direkt nebeneinander bzw. aufeinander gestellt. In der ersten Objekthypothese wurden beide Objekte zwar anfangs als eine Einheit ermittelt. Allerdings löste sich durch die Interaktion mit der Objekthypothese die Mehrdeutigkeit auf, indem sich nur ein Objekt wegbewegt hatte. Dieses Objekt wurde auch anschließend verifiziert. Beim stehengebliebenen Objekt geschah die Verifikation erst im nächsten Explorationszyklus durch eine erneute Interaktion. Desweiteren beeinflusst das Umfallen von instabilen Objekten während des Verschiebens nicht die korrekte Ermittlung der neuen Hemmungsposition und der Analyse der Beweglichkeit, wie in den Experimenten 3, 8 und 19 zu sehen. Zudem ist oftmals, die Segmentierung nach der Interaktion eines beweglichen Objektes wesentlich verbessert, da dort das Objekt über eine Bewegung segmentiert wird. Im Experiment 8 zum Beispiel wurde das Deodorant aufgrund seiner reflektiven Eigenschaft nur im oberen Drittel segmentiert bzw. eine Objekthypothese erstellt. Aufgrund des ermittelten hohen Anfahrtspunktes fiel das Objekt während der Interaktion um, aber anschließend wurde eine viel genauere Segementierung gewonnen. Im Experiment 4 zeigte sich, dass keine Anfahrtsversuche notwendig sind, wenn die Hypothese mit den memorierten Objekten wiedererkannt wird. Allein die grüne Dose musste nochmal interaktiv verifiziert werden, weil durch die homogene Oberfläche der Erkenner zu wenig markante Merkmale fand. Bei den letzten vier Experimenten, bei denen der Tisch mit einem texturierten Hintergrund versehen wurde, konnten trotz der Störstruktur in der Aufmerksamkeitskarte fast alle Objekte gefunden werden. Lediglich im Experiment 18 hatte die kleine Figur, wegen des texturierten Hintergrundes, einen zu geringen Aufmerksamkeitswert und somit wurde keine Objekthypothese für die Figur erstellt.

Die Grenzen des Gesamtsystems betreffen vorallem die Größe der Roboterhand und der Trajektorienplannung. Der Anfahrtspunkt konnte oftmals nicht kollisionsfrei erreicht werden. Entweder geriet die Plannung in ein lokales Minima oder es existierte aufgrund der Handgröße kein kollisionsfreier Pfad. Das Herabsetzen des Kollisionsabstandes brachte nur einen geringen Erfolg. Eine Erhöhung des erlaubten Fehlers an der Zielposition wurde auch in Betracht gezogen. Allerdings wurde die Idee wieder verworfen, weil der Fehler am Anfahrtspunkt nicht mehr tolerierbar waren. Daher wurden alle Anfahrtspunkte die einen kleineren Abstand als 3cm zur planaren Hauptfläche aufwiesen nicht angefahren. Das ist auch der Grund, warum in den Experimenten 9, 10 und 19 die erzeugte Objekthypothese der kleinen Figur verworfen wurde. Ein weiteres Problem besteht in der Auflösung von Mehrdeutigkeiten, wenn sich die Objekte während der Interaktion nicht trennen. In Experiment 13 und 15 wurden ein Objekt auf ein anderes Objekt gestapelt. Während der Interaktion fiel kein Objekt herunter oder trennte sich in irgendeiner Weise vom anderen Objekt. Daher wurden die beiden Objekte fälschlicherweise als ein zusammengehöriges Objekt wahrgenommen. Schwierigkeiten verursachen zudem Objekte, bei der die Kinect-Kamera keinerlei Tiefeninformationen ermittelt. Eine richtige Projizierung des Anfahrtspunktes in Weltkoordinanten und eine Abbildung im Simulator ist ohne Tiefeninformation nicht möglich. Desweiteren sind mit einer starren Kamera verdeckte Objekte nicht sichtbar für das System. Davor liegende Objekte müssten nach rechts oder links verschoben werden, was jedoch unter Umständen wieder andere Objekte verdeckt.

7 Zusammenfassung und Ausblick

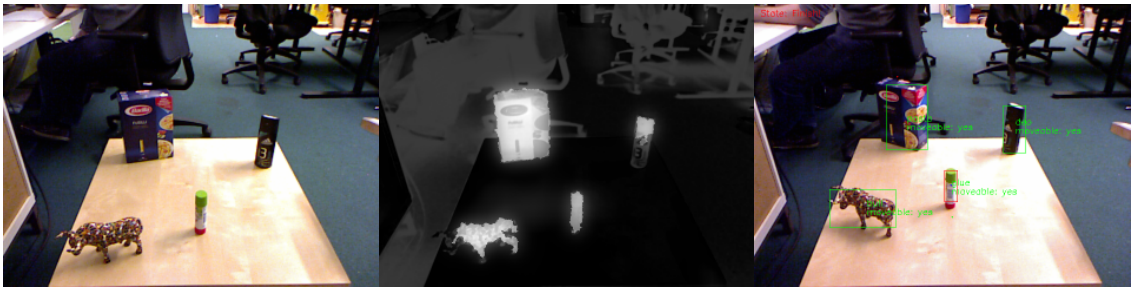
Das Ziel dieser Arbeit, ein System für autonom lernende Roboter zu entwickeln, das die Umwelt wie ein Kleinkind exploriert und die Objekteigenschaften analysiert, konnte vollständig erreicht werden. Getrieben wurde die Exploration durch eine Aufmerksamkeitskarte, welche die nächsten Explorationsorte anzeigt. Dafür wurde das Standardaufmerksamkeitsmodell von [IKN98] angepasst und um vier zusätzliche Auffälligkeitskarten Symmetrie, Größe, Tiefe und Cluster erweitert. Anschließend wurden an jedem Explorationsort, durch eine zweistufige Segmentierung, verschiedene Objekthypothesen generiert. Die Verifizierung jeder Hypothesen erfolgte entweder durch einen Vergleich mit memorierten Objekten aus früheren Explorationen oder anhand einer visuellen oder taktilen Wahrnehmung am Explorationsort durch Interaktion mit einem Roboterarm. Dabei war die Interaktion auf das Verschieben eines potentiellen Objektes ausgerichtet, wodurch gegebenenfalls gleichzeitig die Beweglichkeit eines Objektes erschlossen wurde. Die Ergebnisse der abschließenden experimentelle Evaluierung ergaben, dass das Gesamtsystem oftmals in der Lage ist, bei schwierigen Verhältnissen wie Hintergrundstörungen oder Objektmehrdeutigkeiten, verschiedenartige Objekte korrekt zu lokalisieren und deren Beweglichkeit robust zu bestimmen. Die Grenzen betrafen vorwiegend die Größe der Roboterhand. Die fast doppelte Handgröße bezüglich einer menschlichen Hand erlaubte vereinzelt keine kollisionsfreien Trajektorienpfade zum Anfahrtspunkt, wodurch die entsprechende Objekthypothese verworfen wurde.

Aufbauend auf dieser Arbeit sind weitere Analysen und Weiterentwicklungen vorstellbar. Zum Beispiel könnten, nachdem ein verifiziertes bewegliches Objekt gefunden wurde, noch mehrere Eigenschaften interaktiv mit einem Roboterarm bestimmt werden. Die Materialoberfläche, das Gewicht oder die komplette 3D-Form lassen sich durch greifen, anheben und drehen bzw. kippen des Objektes herausfinden. Zusätzlich sollten die semantische Information eines Objektes vollständig autonom bezogen werden. Ein vielversprechender Ansatz könnte hier sein, die im Kapitel 5.4 extrahierten Merkmale oder die segmentierten Bilder, mit großen semantischen Datenbanken, wie zum Beispiel von der *Stanford University* betriebene Internetseite „ImageNet“ oder dem „LabelMe“ Projekt vom *Massachusetts Institute of Technology*, zu vergleichen. Weiterhin befindet sich noch Entwicklungspotential in der Aufmerksamkeitskarte des Gesamtsystems. Hier könnten weitere Auffälligskeitskarten kombiniert und untersucht werden, die zum Beispiel zeitliche Merkmale einer Szene mit auswerten. Auch eine

Erweiterung um eine *top-down* Komponente ist für gezielte Aufgaben wünschenswert. Eine Idee für eine Anwendung könnte sein, verschiedenfarbige Objekte in einzelne Kisten einzusortieren oder gezielt Tassen, Nudelpackungen, Tomatendosen für einen Haushaltsroboter auf einem Tisch zu suchen.

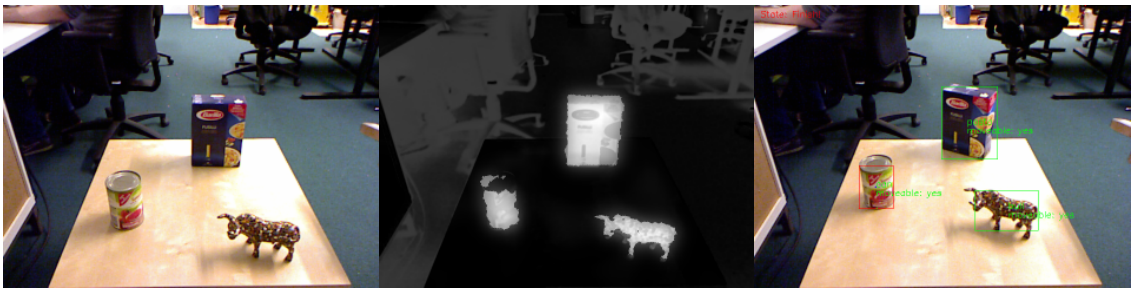
Anhang

A Darstellungen der einzelnen Versuchskonstellationen



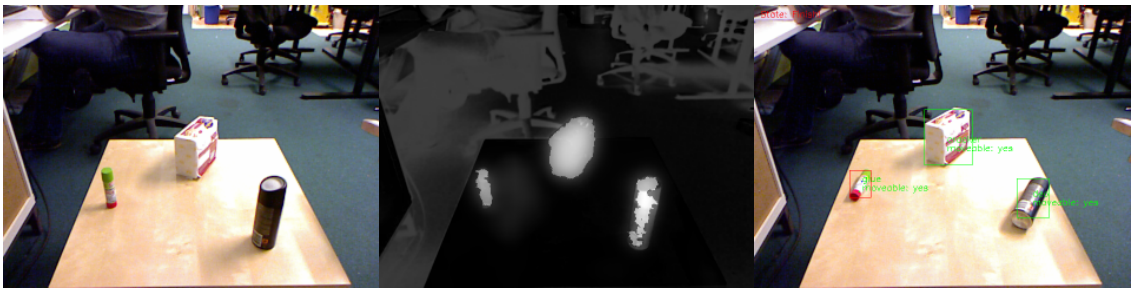
(a) (b) (c)

Abbildung A.1: (a) 1. Versuchskonstellation mit Nudelpackung, Stierstatue, Deodorant und Klebestift, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion (Man beachte die Verschiebung der Objekte). Alle Objekthypothesen wurden erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



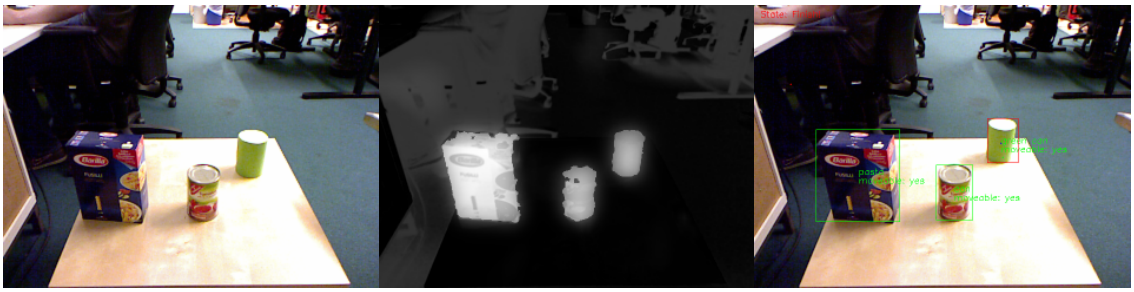
(a) (b) (c)

Abbildung A.2: (a) 2. Versuchskonstellation mit Nudelpackung, Stierstatue und Tomatendose, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Alle Objekthypothesen wurden erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



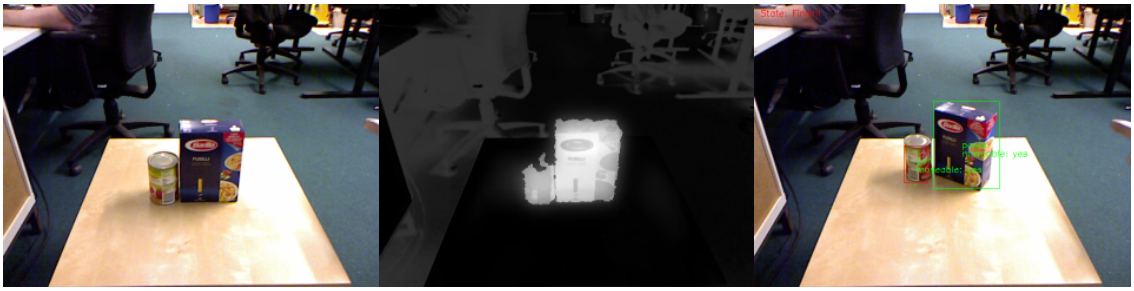
(a) (b) (c)

Abbildung A.3: (a) 3. Versuchskonstellation mit Knäckebrotpackung, Klebestift und Deodorant, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Bei der Hypothesenverifikationsphase von Deodarant und Klebestift kippten die Objekte um. Dennoch wurden alle Objekthypothesen erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



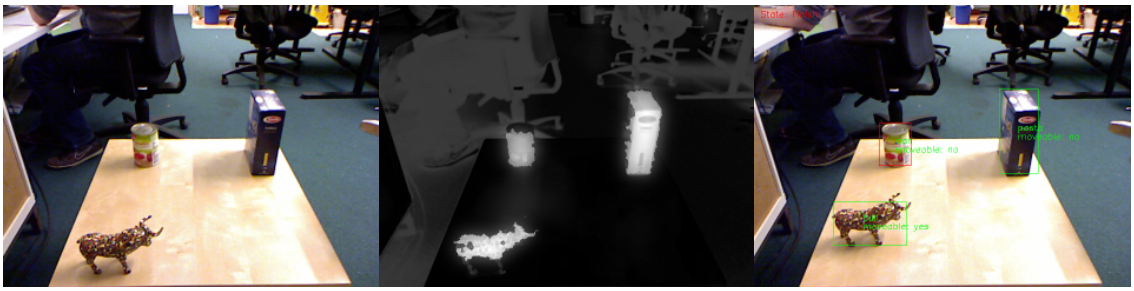
(a) (b) (c)

Abbildung A.4: (a) 4. Versuchskonstellation mit Nudelpackung, Tomatendose und grüne Dose, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Vor der Versuchsdurchführung wurden die memorierten Objekte *nicht* gelöscht. Dadurch wurde erreicht, dass die Nudelpackung und die Tomatendose bereits anhand der memorierten Objekten erfolgreich verifiziert wurden. Allein bei der Objekthypothese der grünen Dose erfolgte eine Interaktion, da diese fälschlicherweise in den memorierten Objekten nicht wiedererkannt wurde. Letztendlich wurden alle Objekthypothesen erfolgreich verifiziert.



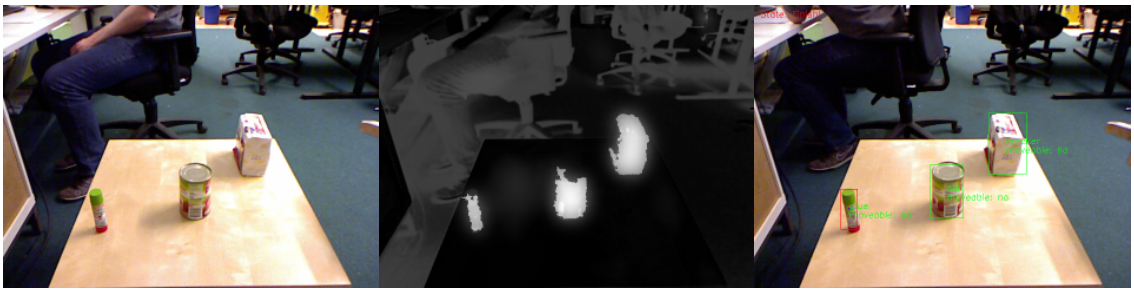
(a) (b) (c)

Abbildung A.5: (a) 5. Versuchskonstellation mit Nudelpackung und Tomatendose direkt nebeneinander, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. In der ersten generierten Objekthypothese wurden beide Objekte fälschlicherweise als zusammengehörig erkannt. Durch die Interaktion wurde die Mehreindeutigkeit erfolgreich aufgelöst und deren Beweglichkeiten richtig analysiert.



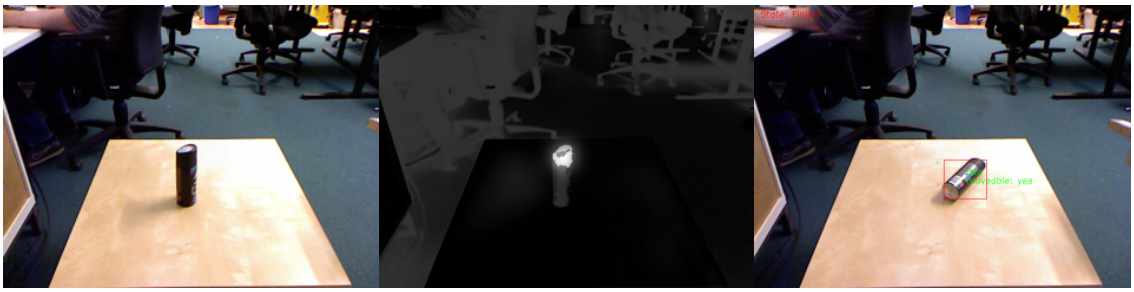
(a) (b) (c)

Abbildung A.6: (a) 6. Versuchskonstellation mit Nudelpackung, Stierstatue und Tomatendose, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Zum Simulieren eines nicht bewegbaren Objektes wurde die Nudelpackung und die Tomatendose während der Hypothesenverifikationsphase per Hand festgehalten. Alle Objekthypothesen wurden erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



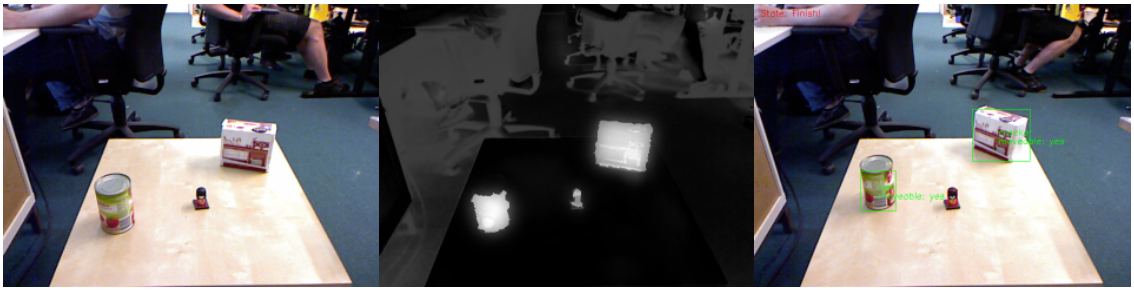
(a) (b) (c)

Abbildung A.7: (a) 7. Versuchskonstellation mit Knäckebrotpackung, Klebestift und Tomatendose, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Zum Simulieren eines nicht bewegbaren Objektes wurden alle Objekte während der Hypothesenverifikationsphase per Hand festgehalten. Alle Objekthypothesen wurden erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



(a) (b) (c)

Abbildung A.8: (a) 8. Versuchskonstellation mit Deodorant, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Bei der Hypothesenverifikationsphase vom Deodarant kippte das Objekt um, weil nur im oberen Drittel vom Objekt Tiefeninformationen ermittelbar waren und somit der Anfahrtspunkt über den Objektschwerpunkt lag. Gleichzeitig erfasste die Segmentierung zunächst, aufgrund der fehlenden Tiefeninformation, das ganze Objekt nicht. Allerdings wurde die Segmentierung des Objektes nach der Interaktion wesentlich verbessert. Zugleich wurde die Objekthypothese erfolgreich verifiziert und deren Beweglichkeit richtig analysiert.

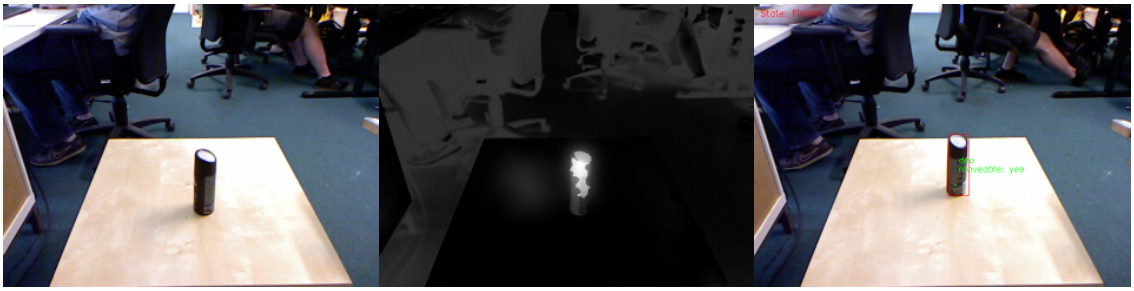


(a) (b) (c)

Abbildung A.9: (a) 9. Versuchskonstellation mit Knäckebrotpackung, Tomatendose und Figur (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Die Objekthypothese von der Figur wurde verworfen, da der Anfahrtspunkt zu niedrig war. Die Objekthypothese von der Knäckebrotpackung und der Tomatendose wurden hingegen erfolgreich verifiziert und deren Beweglichkeit richtig analysiert.

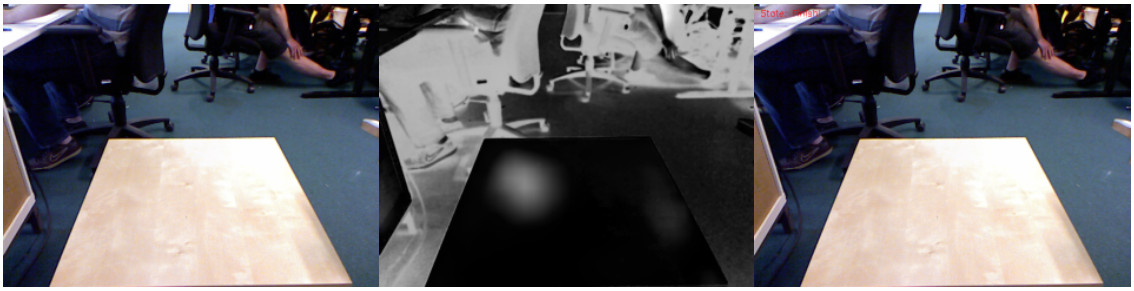
(a) (b) (c)

Abbildung A.10: (a) 10. Versuchskonstellation mit Nudelpackung und Figur (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Die Objekthypothese von der Figur wurde verworfen, da der Anfahrtspunkt zu niedrig war. Die Objekthypothese von der Nudelpackung wurde hingegen erfolgreich verifiziert und deren Beweglichkeit richtig analysiert.



(a) (b) (c)

Abbildung A.11: (a) 11. Versuchskonstellation mit Deodorant, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Die Objekthypothese wurde erfolgreich verifiziert und deren Beweglichkeit richtig analysiert.

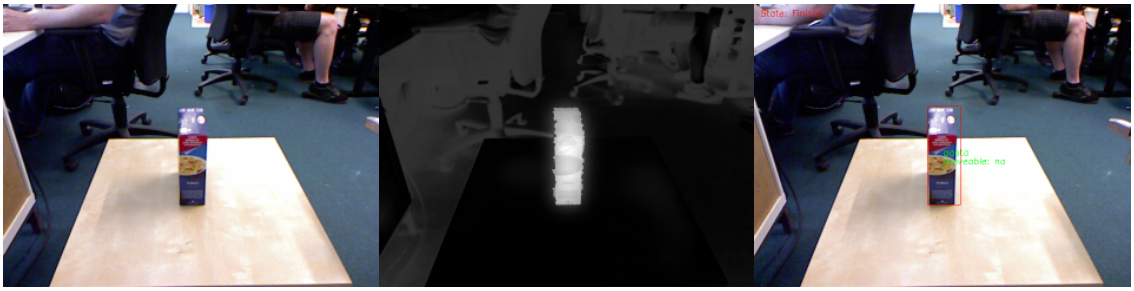


(a) (b) (c)

Abbildung A.12: (a) 12. Versuchskonstellation mit keinen Objekten, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild. Es wurde korrekterweise keine Objekthypothese erstellt.

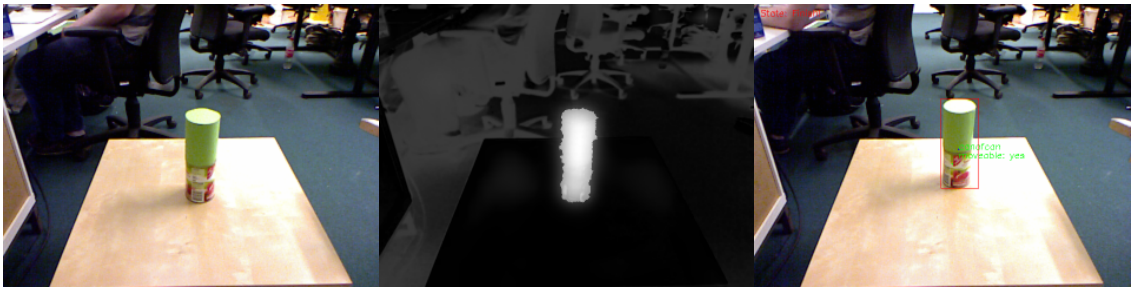
(a) (b) (c)

Abbildung A.13: (a) 13. Versuchskonstellation mit Figur auf Tomatendose gestapelt, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Die Mehreindeutigkeit konnte während Hypothesenverifikationsphase nicht aufgelöst werden, da sich die beiden Objekte nicht getrennt haben. Daher wurden die beiden Objekte fälschlicherweise als ein ganzes Objekt wahrgenommen.



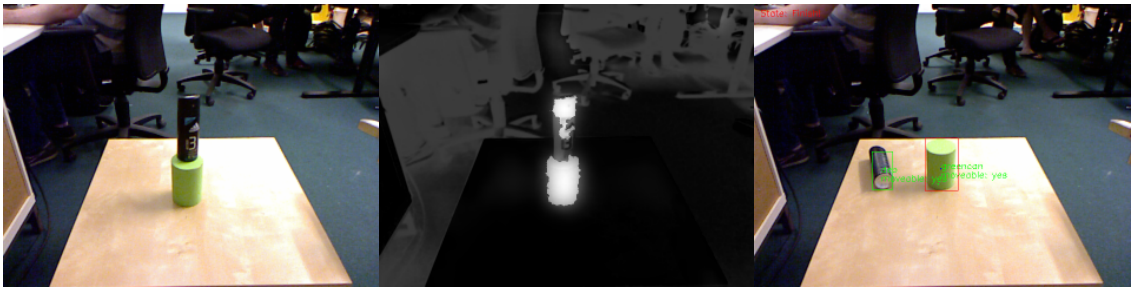
(a) (b) (c)

Abbildung A.14: (a) 14. Versuchskonstellation mit Nudelpackung (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Zum Simulieren eines nicht bewegbaren Objektes wurde die Nudelpackung während der Hypothesenverifikationsphase per Hand festgehalten. Die Objekthypothese wurde erfolgreich verifiziert und deren Beweglichkeit richtig analysiert.



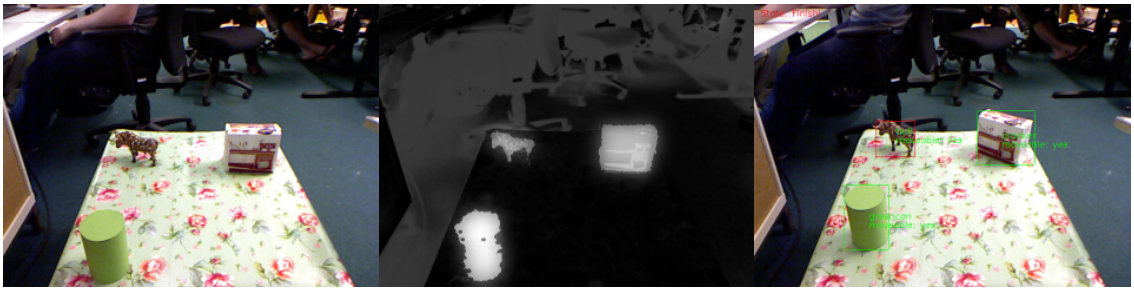
(a) (b) (c)

Abbildung A.15: (a) 15. Versuchskonstellation mit grüne Dose auf Tomatendose gestapelt, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Die Mehreindeutigkeit konnte während der Hypothesenverifikationsphase nicht aufgelöst werden, da sich die beiden Objekte nicht getrennt haben. Daher wurden die beiden Objekte fälschlicherweise als ein ganzes Objekt wahrgenommen.



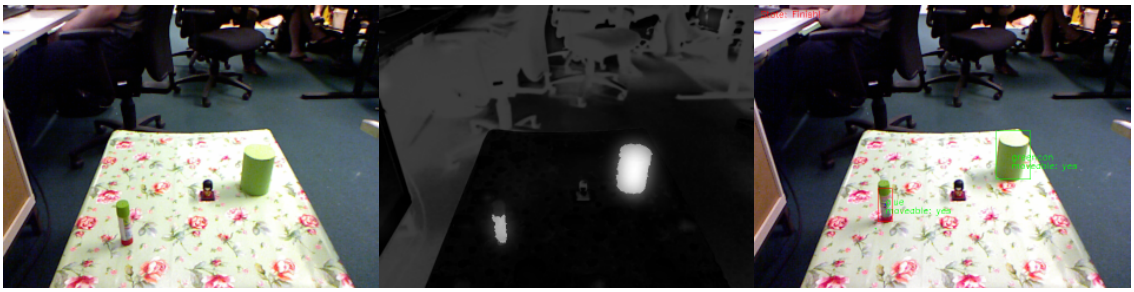
(a) (b) (c)

Abbildung A.16: (a) 16. Versuchskonstellation mit Deodorant auf grüne Dose gestapelt, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Die Mehreindeutigkeit konnte während der Hypothesenverifikationsphase aufgelöst werden, da sich beide Objekte getrennt haben. Beide Objekte wurden erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



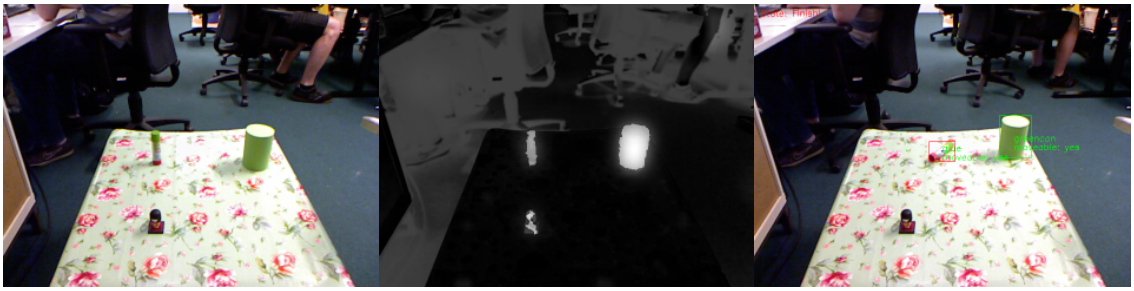
(a) (b) (c)

Abbildung A.17: (a) 17. Versuchskonstellation mit Knäckebrotpackung, Stierstatue, grüne Dose und strukturierten Hintergrund, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Alle Objekthypothesen wurden erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



(a) (b) (c)

Abbildung A.18: (a) 18. Versuchskonstellation mit Klebestift, Figur, grüne Dose und strukturierten Hintergrund, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Es wurde keine Objekthypothese für die Figur erstellt, da die Aufmerksamkeitswerte an dem Ort zu gering war. Die Objekthypothesen von dem Klebestift und von der grüne Dosen wurden hingegen erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.



(a) (b) (c)

Abbildung A.19: (a) 19. Versuchskonstellation mit Klebestift, Figur, grüne Dose und strukturierten Hintergrund, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Die Objekthypothese von der Figur wurde verworfen, da der Anfahrtspunkt zu niedrig war. Bei der Hypothesenverifikationsphase vom Klebestift kippte das Objekt um. Die Objekthypothese von dem Klebestift und von der grünen Dose wurde erfolgreich verifiziert und deren Beweglichkeit richtig analysiert.

(a) (b) (c)

Abbildung A.20: (a) 20. Versuchskonstellation mit Knäckebrotpackung, Deodorant und strukturierten Hintergrund, (b) Aufmerksamkeitskarte und (c) annotiertes Ergebnisbild nach der Interaktion. Alle Objekthypothesen wurden erfolgreich verifiziert und deren Beweglichkeiten richtig analysiert.

Abbildungsverzeichnis

Literaturverzeichnis

[AHES09] Radhakrishna Achanta, Sheila S. Hemami, Francisco J. Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[AHK+09] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida. Cognitive developmental robotics: A survey. *IEEE Transactions on Autonomous Mental Development*, 1:12–34, 2009.

[And07] John R. Anderson. *Kognitive Psychologie*. Spektrum Akademischer Verlag, 2007.

[Bac04] Gerriet Backer. *Modellierung visueller Aufmerksamkeit im Computer-Sehen: Ein zweistufiges Selektionsmodell für ein Aktives Sehsystem*. PhD thesis, Universität Hamburg, 2004.

[Bei05] Hans-Jürgen Beins. Bauen und Konstruieren als lustvolles Lernen. *kindergarten heute*, 1:6–12, 2005.

[BJ01] Yuri Y. Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *International Conference on Computer Vision*, 1:105–112, 2001.

[CDLI09] Krzysztof Charusta, Dimitar Dimitrov, Achim J. Lilienthal, and Boyko Iliev. Extraction of grasp related features by human dual-hand object exploration. In *Proceedings of the IEEE International Conference on Advanced Robotics (ICAR)*, June 22–26 2009.

[Die08] Felix Diener. *Das visuelle Verarbeitungssystem des Menschen*. GRIN Verlag, 2008.

[DOK08] A. Ude D. Omrcen and A. Kos. Learning primitive actions through object exploration. In *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Daejeon, Korea, 2008.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[GBBK10] Xavi Gratal, Jeannette Bohg, Mårten Björkman, and Danica Kragic. Scene representation and object grasping using active vision. In *IROS'10 Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics*, 2010.

[HH09] Christian Hick and Astrid Hick. *Intensivkurs Physiologie*. Urban & Fischer Verlag, 2009.

[HKMS11] H. Hagendorf, J. Krummenacher, J. Müller, and T. Schubert. *Wahrnehmung und Aufmerksamkeit*. Springer Verlag, 2011.

[HT00] Nouchine Hadjikhani and Roger B.H. Tootell. Projection of rods and cones within human visual cortex. *Human Brain Mapping*, 9(1):55–63, 2000.

[IKN98] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[KB08] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *Proceedings of the International Conference on Advanced Robotics*, pages 272–277, 2008.

[KOB10] Dov Katz, Andreas Orthey, and Oliver Brock. Interactive perception of articulated objects. In *12th International Symposium of Experimental Robotics*, pages 1–15, 2010.

[Kov97] Peter Kovesi. Symmetry and asymmetry from local phase. In *Tenth Australian Joint Converence on Artificial Intelligence*, pages 2–4, 1997.

[KYDB07] G. Kootstra, J. Ypma, and B. De Boer. Exploring objects for recognition in the real world. *Proceedings of IEEE International Conference on Robotics and Biomimetics*, pages 429–434, 2007.

[KYS+07] Yasuo Kuniyoshi, Yasuaki Yorozu, Shinsuke Suzuki, Shinji Sangawa, Yoshiyuki Ohmura, Koji Terada, and Akihiko Nagakubo. Emergence and development of embodied cognition: a constructivist approach using robots. In C. von Hofsten and K. Rosander, editors, *From Action to Cognition*, volume 164 of *Progress in Brain Research*, pages 425 – 445. Elsevier, 2007.

[LMPS03] Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. Developmental robotics: a survey. *Connection Science*, 15:151–190, 2003.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[Lö04] Frank Lömker. *Lernen von Objektbenennungen mit visuellen Prozessen*. PhD thesis, Universität Bielefeld, 2004.

[Mic11] Microsoft. Programming Guide: Getting Started with the Kinect for Windows SDK Beta 1 Draft Version 1.1, 2011. Zugriff am 20.08.2011 unter http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/.

[ML09] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application*, pages 331–340. INSTICC Press, 2009.

[NI05] V. Navalpakkam and L. Itti. Modeling the influence of task on attention. *Vision Research*, 45(2):205–231, 2005.

[PI04] Rolf Pfeifer and Fumiya Iida. Embodied artificial intelligence: Trends and challenges. In *Embodied Artificial Intelligence*, volume 3139 of *Lecture Notes in Computer Science*, pages 629–629. Springer Berlin / Heidelberg, 2004.

[Pri11] PrimeSense. Prime sensor reference design 1.08 datasheet, 2011. Zugriff am 7.06.2011 unter http://www.primesense.com.

[PS85] Franco P. Preparata and Michael I. Shamos. *Computational geometry: an introduction*. Springer Verlag, 1985.

[PS99] Rolf Pfeifer and Christian Scheier. *Understanding intelligence*. MIT Press, 1999.

[RKB04] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23:309–314, 2004.

[Rus09] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Technische Universität München, 2009.

[Sch00] Steffen Schmalz. *Entwurf und Evaluierung von Strategien zur 2D/3D-Objekterkennung in aktiven Sehsystemen*. PhD thesis, Universität Hamburg, 2000.

[Sch11] Schunk. Produktinformationen zu Robotic Hands SDH und 7DOF LWA Manipulator, 2011. Zugriff am 7.06.2011 unter http://www.schunk-modular-robotics.com.

[SPS+09] Jürgen Sturm, Vijay Pradeep, Cyrill Stachniss, Christian Plagemann, Kurt Konolige, and Wolfram Burgard. Learning kinematic models for articulated objects. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pages 1851–1856, 2009.

[SS11] Marlene Schnelle-Schneyder. *Sehen und Photographie*. Springer Verlag, 2011.

[Sto11] Alexander Stoytchev. Self-detection in robots: a method based on detecting temporal contingencies. *Robotica*, 29:1–21, 2011.

[Tou09] Marc Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1049–1056. ACM, 2009.

[vF07] Heinz von Foerster. Das Konstruieren einer Wirklichkeit. In *Die erfundene Wirklichkeit: Wie wissen wir, was wir zu wissen glauben? von Paul Watzlawick*. Piper Verlag, 2007.

[VR06] Vibha S. Vyas and Priti Rege. Automated texture analysis with gabor filter. *Journal on Graphics, Vision and Image Processing*, 6:35–41, 2006.

[Wol94] Jeremy M. Wolfe. Guided search 2.0 a revised model of visual search. *Psychonomic Bulletin and Review*, 1:202–238, 1994.

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Berlin, den 25. August 2011

_____

Oliver Erler