Die Brille mit Gedächtnis

—

Interaktives Lernen mit einem mobilen Augmented-Reality-System

Holger Bekel

The Neuroinformatics Group
Faculty of Technology
Bielefeld University
Germany

Dipl.-Biol., Dipl.-Inform Holger Bekel
AG Neuroinformatik
Technische Fakultät
Universität Bielefeld
email:hbekel@uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 21.04.2010 vorgelegt von Holger Bekel
am 29.10.2010 verteidigt und genehmigt.

Gutachter:

- Prof. Dr. Helge Ritter
- Prof. Dr. Gunther Heidemann

Prüfungsausschuss:

- Prof. Dr. Franz Kummert
- Prof. Dr. Helge Ritter
- Prof. Dr. Gunther Heidemann
- Dr. Frank Röben

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

Für meine Eltern

Danksagung

Bei der Arbeit an der vorliegenden Dissertation, die im Rahmen des Projektes VAMPIRE entstand, erhielt ich vielfältige Unterstützung, für die ich mich an dieser Stelle bedanken möchte.

Mein besonderer Dank gilt Prof. Dr. Gunther Heidemann für seine langjährige Betreuung meiner Arbeit. Die angenehme kollegiale und produktive Zusammenarbeit mit ihm und mit Prof. Dr. Ingo Bax während der gesamten Projektlaufzeit werden mir immer in guter Erinnerung bleiben.

Weiterhin möchte ich mich sehr herzlich bei Prof. Dr. Helge Ritter bedanken, dessen Begeisterung an seinem Fach bei mir das Interesse für das Gebiet des künstlichen Lernens, insbesondere der neuronalen Netze geweckt hat. Sowohl seine fachliche Kompetenz und Beratung als auch die angenehme und konstruktive Atmosphäre in seiner Arbeitsgruppe habe ich während der gesamten Zeit von der Diplomarbeit bis zum Ende des Projektes sehr geschätzt.

Ebenfalls danke ich allen Kollegen des Projektes VAMPIRE, insbesondere dem Projektleiter Prof. Dr. Gerhard Sagerer. Die vielen gemeinsamen internationalen Projekttreffen, der gemeinsame Besuch verschiedener Konferenzen wie auch die Besprechungen vor Ort waren immer wieder inspirierend und gewinnbringend. Hervorheben möchte ich die kritisch-konstruktive und gleichzeitig äußerst sympathische Arbeitsatmosphäre.

Meinem Bürokollegen Dr. Axel Saalbach möchte ich für seine ständige Bereitschaft zum Austausch und seine vielen Anregungen bedanken. Ebenso hat Stefanie Schwassmann als studentische Hilfskraft zum Gelingen dieser Arbeit beigetragen. Bemerkenswert war, wie schnell und hoch motiviert sie sich in neue Teilbereiche eingearbeitet und diese praktisch umgesetzt hat.

Nicht zuletzt möchte ich mich bei meiner Familie bedanken, die mir durch ihre langjährige Unterstützung diesen Werdegang erst ermöglicht haben. Mein besonderer Dank gilt Jun. Prof. Dr. Valerie Kastrup, die mich genauso liebevoll wie geduldig und tatkräftig unterstützt hat, mit sehr viel Akribie die Aufgabe des Korrekturlesens übernommen hat und mich zusammen mit ihrer Familie immer wieder ermuntert hat, die Arbeit trotz der neuen beruflichen Herausforderungen fertigzustellen.

Veröffentlichungen

Teile dieser Arbeit sind bereits im Voraus in folgenden Journalen und auf folgenden Konferenzen veröffentlicht worden:

Journale:

- **Interactive Online Learning**
  G. Heidemann, H. Bekel, I. Bax, H. Ritter
  Pattern Recognition and Image Analysis, 15, (1), 55-58, 2007.

- **Interactive Image Data Labeling Using Self-Organizing Maps in an Augmented Reality Scenario**
  H. Bekel, G. Heidemann, H. Ritter
  Neural Networks, 18, (5/6), 566-574, 2005.

- **Integrating Context-Free and Context-Dependent Attentional Mechanisms for Gestural Object Reference**
  G. Heidemann, R. Rae, H. Bekel, I. Bax, H. Ritter
  Machine Vision and Applications, 16, (1), 64-73, 2004.

Konferenzen:

- **SOM Based Image Data Structuring in an Augmented Reality Scenario**
  H. Bekel, G. Heidemann, H. Ritter
  Montréal, Québec, Proc. Intl Joint Conf. on Neural Networks, Sep. 2005.

- **Adaptive Computer Vision: Online Learning for Object Recognition**
  H. Bekel, I. Bax, G. Heidemann, H. Ritter
  Proc. DAGM 2004, 447-454, Eds.: C. E. Rasmussen et al., Springer, 2004.

- **Multimodal interaction in an augmented reality scenario.**
  G. Heidemann, I. Bax, H. Bekel, C. Bauckhage, S. Wachsmuth, G. Fink, A. Pinz, H. Ritter, and G. Sagerer.
  In Proc. Int'l Conf. Multimodal Interfaces ICMI 2004, pages 53-60. ACM Press, 2004.

- **Interactive Online Learning.**
  G. Heidemann, H. Bekel, I. Bax, and H. Ritter.
  In Proc. PRIA 2004, pages 44-48, 2004.

- **Hand Gesture Recognition: Self-Organising Maps as a Graphical User Interface for the Partitioning of Large Training Data Sets**
  G. Heidemann, H. Bekel, I. Bax, A. Saalbach
  Cambridge, UK, Proc. ICPR 2004, 4, 487-490, Eds.: J. Kittler, M. Petrou and M. Nixon, IEEE CS-Press, 2004.

- **Integrating Context-Free and Context-Dependent Attentional Mechanisms for Gestural Object Reference**
  G. Heidemann, R. Rae, H. Bekel, I. Bax, H. Ritter
  Graz, Austria, Proc. Intl Conf. Cognitive Vision Systems, 22–33, 2003.

- **Recognition of gestural object reference with auditory feedback.**
  I. Bax, H. Bekel, and G. Heidemann.
  In Proc. Int'l Conf. Neural Networks, pages 425-432, Istanbul, Turkey, 2003.

Abkürzungsverzeichnis

Verwendete Abkürzungen:

API	Application Programming Interface, Programmierschnittstelle
ATM	ATtention Module, Modul zur Aufmerksamkeitsberechnung
EVS	Ego Vision System, mobiles System, welches den Blickwinkel des Benutzers teilt und diesen bei Prozessen einbindet
Exset	Example Set, Datensatz für die Weiterverarbeitung in NESSY
FP	Fokuspunkt
GWF	Gradient Weight Function, Gewichtungsfunktion für die Symmetrieberechnung
LLM	Local Linear Map, neuronales Klassifikationsverfahren
PCA	Principal Component Analysis, Hauptkomponentenanalyse
LTM	Long Term Memory, Langzeitgedächtnis
NESSY	NEural viSion SYstem, Software-Umgebung
PWF	Phase Weight Function, Gewichtungsfunktion für die Symmetrieberechnung
RMI	Remote Method Invocation, Methodenfernaufruf
ROI	Region Of Interest, Region erhöhter Aufmerksamkeit
RPC	Remote Procedure Calls, Technik zur Realisierung von Interprozesskommunikation
SIFT	Scale Invariant Feature Transform, Skalierungsinvariante Bildmerkmale
SOM	Self Organizing Maps, Selbstorganisierende Karten
STM	Short Term Memory, Kurzzeitgedächtnis
VAM	Visual Active Memory, Aktives visuelles Gedächtnis
VAMPIRE	Visual Active Memory Processes and Interactive REtrieval, Name des EU-Projektes
VPL	Vector quantisation-PCA-LLM, Klassifikator aus den drei Schichten Vektorquantisierung, PCA und LLM-Netzen
XCF	XML enabled Communication Framework, middleware zur objektorientierten Übertragung von Informationen
XML	eXtensible Markup Language, Auszeichnungssprache zur Darstellung hierarchisch strukturierter Datensätze in Form von Textdaten

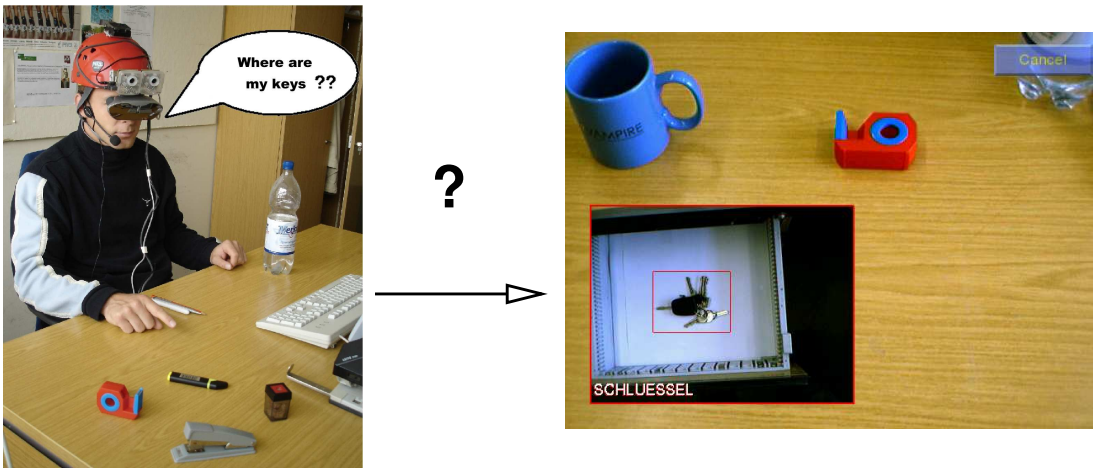
Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation für die Entwicklung einer Brille mit Gedächtnis

Eines der häufigsten Missgeschicke, die einem tagtäglich passieren, ist, dass man einen Gegenstand verlegt hat und ihn partout nicht wiederfinden kann. Wer würde sich in solchen Situationen nicht einen Assistenten wünschen, der jederzeit eine Antwort auf Fragen wie z.B. *„Wo ist mein Schlüsselbund?"* hat? Für die viele Probleme im alltäglichen Leben gibt es inzwischen computergestützte Geräte, die einem den Alltag erleichtern. Dies ist aufgrund der kontinuierlichen Steigerung der Rechnerkapazitäten bei parallel fortschreitender Miniaturisierung möglich geworden. So ist es heute völlig normal, dass man mit seinem Handy nicht nur telefonieren, sondern nebenbei noch Fotos machen und sogar Filme aufnehmen kann. Mit einem modernen Handy kann man Radio hören und MP3-Musikdateien abspielen oder man lässt sich per GPS den Weg leiten. Ein lückenhaftes Gedächtnis in Bezug auf anstehende Termine, Geburtstage und andere Daten auszugleichen, gehört für ein modernes Handy zu den selbstverständlichen Funktionen.



Dieses Beispiel zeigt, dass die Entwicklungsgeschwindigkeit in diesem Bereich die Vorstellungen von vor 20 Jahren um ein Vielfaches übersteigt. Wieso gibt es also noch kein computergestütztes Gerät, welches einem hilft, verlegte Gegenstände wiederzufinden? Warum kann man noch nicht in einen gewöhnlichen Computerdiscounter gehen und so

etwas wie eine „**Brille mit Gedächtnis**" kaufen? Eine leicht zu tragende Brille, die mich ständig begleitet und welche aufmerksam mich und meine Umgebung beobachtet und mir hilft, wenn ich etwas verlegt habe, indem sie mir dessen Standort mitteilt und somit die Lücken in meinem Gedächtnis schließt. Da es diese Gedächtnisstütze nicht gibt, resultiert daraus die Frage, inwiefern sich die Funktionalität einer Brille mit Gedächtnis von den oben genannten Funktionen eines Handys oder anderer moderner Geräte unterscheidet. Geht man dieser Frage nach, so trifft man auf die ungeheure Komplexität, die die Konstruktion eines solchen Gerätes mitbringen würde.

Eine Person mit einem guten Gedächtnis, welche einen ständig beobachtet, wäre leicht imstande, den Ort eines Gegenstandes, den man verlegt hat, mitzuteilen. Aber welche Kenntnisse und Fähigkeiten sind die Voraussetzungen dafür? Und wie kann man diese in einem computergestützten System realisieren, um solche Aufgaben zu bewältigen?

Die Liste der benötigten Fähigkeiten ist lang und basiert auf zwei essentiellen Voraussetzungen: Als erstes müsste ein solches System ein mit dem Benutzer[1] gemeinsames Verständnis von der Umgebung haben. Dieses gemeinsame Verständnis impliziert die Kenntnis des Ortes des Benutzers ebenso wie seine Blickrichtung. Dabei muss ein solches System die Objekte in der Umgebung und deren Lage im Raum kennen oder zumindest die Möglichkeit bieten, unbekannte Objekte zu erlernen.

Die zweite essentielle Bedingung für die Entwicklung einer Brille mit Gedächtnis ist, dass es einen Weg geben muss, über den das System mit dem Benutzer auf möglichst multimodale Weise und bidirektional kommuniziert. Das System muss dem Benutzer mitteilen können, wo sich verlegte Objekte befinden und falls ein neues Objekt erlernt werden muss, ist die Verständigung mit dem Benutzer über das neu zu erlernende Wissen notwendig. Diese Kommunikation zwischen Mensch und Maschine setzt dabei das gemeinsame Verständnis voraus. Dazu reicht es nicht, die verbale Frage nach dem Ort eines Gegenstandes beispielsweise mittels eines Spracherkennungssystems zu erkennen. Vielmehr liegt die eigentliche Herausforderung darin, dass das System die kontextuelle Semantik der Frage verstehen muss, nämlich dass etwas gesucht wird, um welche Art Objekt es sich handelt oder auf welches Objekt der Benutzer gerade z.B. mit einer Zeigegeste referenziert.

Die Erfüllung dieser beiden Bedingungen führt wiederum zu einer vielschichtigen Problemkomplexität, und viele dazu notwendigen Teilaufgaben sind bis zum heutigen Zeitpunkt noch nicht zufriedenstellend gelöst oder aber nicht in ein computergestütztes, mobiles System ohne weiteres integrierbar.

In Anbetracht dieser Entwicklungslücken zielt diese Arbeit auf die Entwicklung eines trainierbaren mobilen Objekterkennungssystems als Kernstück einer solchen Brille mit Gedächtnis. Der Schwerpunkt dieser Entwicklung wird dabei auf die interaktive Trainierbarkeit und Lernfähigkeit des laufenden Systems gelegt. Der Benutzer dient dabei als Experte, welcher dem System das notwendige Wissen vermittelt. Natürliche Wege zur Mensch-Maschine-Kommunikation werden als Teil des Systems vorgestellt, welches sowohl Objekte als auch Gesten und Sprache erkennt und welches fähig ist, zur Laufzeit

[1] Um eine bessere Lesbarkeit zu gewährleisten, wird hier nur die männliche Form verwendet. Die weiblichen Benutzerinnen sind damit selbstverständlich ebenfalls gemeint.

neues Objektwissen in einer Interaktionsschleife zwischen Mensch und Maschine sowohl aufzunehmen als auch mit Hilfe künstlicher neuronaler Netze zu erlernen. Es wird aufgezeigt und diskutiert, dass die Komplexität einzelner Aufgaben für die Realisierung eines solchen mobilen Systems reduziert werden muss, wofür praktikable Lösungen vorgestellt werden.

1.2 Aufbau der Arbeit

Die Arbeit beginnt mit einer Darstellung der funktionalen Anforderungen eines interaktiv trainierbaren mobilen Objekterkenners. Anschließend werden die verschiedenen Informatikdisziplinen, welche in dieser Arbeit berührt werden, aufgeführt und aktuelle Entwicklungen in den einzelnen Bereichen vorgestellt. Dabei werden Computer-Vision-Systeme, kognitive Systeme und die Entwicklung und Anwendung moderner Augmented-Reality-Systeme aufgeführt. Am Ende des ersten Teils wird das technische Setup und das Szenario vorgestellt, welches für die Entwicklung verwendet wurde.

Im anschließenden Teil folgt die Beschreibung der praktischen Umsetzung einzelner Teilaufgaben, welche für verschiedene komplexere Funktionen benötigt werden. Diese Basiskomponenten sind die Voraussetzung der komplexen Abläufe bei der Mensch-Maschine-Interaktion. Dabei werden verschiedene Modalitäten der Kommunikation und verschiedene Wege auf dem Weg zum notwendigen gemeinsamen Verständnis zwischen Mensch und Maschine dargestellt. Die visuelle Aufmerksamkeit des Computersystems verknüpft mit dem Nutzen der menschlichen Intelligenz wird als wichtiger Aspekt aufgezeigt.

Der Hauptteil dieser Arbeit befasst sich anschließend mit dem interaktiven Lernen von Objekten. Der Teil beginnt mit einer Übersicht über das Gesamtsystem und dem hierarchisch strukturierten Aufbau der realisierten Funktionen. Diese werden im einzelnen vorgestellt, wobei der Schwerpunkt dieser Arbeit auf die beiden Hauptfunktionen gelegt wird. Zum Einen wird ein online-trainierbares Objekterkennungssystem vorgestellt. Zum Anderen wird ein semiautomatisches Verfahren beschrieben, welches für das Erstellen einer Trainingsmenge für den Objekterkenner durch die Verwendung unüberwachter Lernverfahren entwickelt wurde. Bei der Bewältigung dieser Aufgaben zur Laufzeit fungiert der Benutzer als Wissensvermittler für das künstliche kognitive System.

Die Arbeit schließt mit der Vorstellung des Gesamtsystems des Projektes ab. Dazu werden die Arbeiten der Projektpartner aufgeführt und die Integration der einzelnen Komponenten und des vorgestellten Systems in das Gesamtsystem dargelegt. Am Schluss folgt ein Ausblick über mögliche weitere Entwicklungen als sinnvolle Fortführung dieses Projektes und der hier erlangten Erkenntnisse.

Teil I

Der Weg zur Brille mit Gedächtnis

In diesem ersten von vier Teilen werden die Entwicklungsanforderungen an das angestrebte System, der Stand der Forschung und die Entwicklungsumgebung vorgestellt, bevor dann in den nächsten drei Teilen die Realisierung des Systems beschrieben wird.

Der erste Teil beginnt mit einem Überblick über die Vielzahl der benötigten Funktionen, die für die Realisierung einer „Brille mit Gedächtnis" notwendig sind. Dabei werden die „Brille mit Gedächtnis" und ihre Funktionalität als Vorstufe zur Entwicklung eines persönlichen Assistentensystems vorgestellt.

Im Anschluss werden die verschiedenen Forschungsbereiche präsentiert, welche bei dieser Arbeit berührt werden. Dabei wird der Schwerpunkt auf die Bereiche des maschinellen und kognitiven Sehens, der Erweiterten Realität (Augmented Reality) und des interaktiven Objektlernens gelegt. Es wird aufgezeigt, welche Arten von verwandten Systemen bereits entwickelt wurden, um die Herausforderungen des in dieser Arbeit entwickelten Systems zu verdeutlichen.

Der erste Teil schließt mit der Beschreibung der Entwicklungsumgebung. Hierbei werden das Entwicklungsszenario, die verwendete Hardware sowie die Software-Entwicklungsumgebung vorgestellt.

Kapitel 2

Entwicklungsanforderungen

2.1 Die Brille mit Gedächtnis als Teil eines persönlichen Assistentensystems

Das Entwicklungsziel dieser Arbeit ist die Konstruktion einer Brille mit Gedächtnis gewesen. Diese Brille mit Gedächtnis ist dabei als erster Schritt der Entwicklung eines mobilen computergestützten Büro-Assistentensystems anzusehen. Das Büro-Assistenten-System dient der Erforschung aktiver visueller Gedächtnisprozesse und deren interaktive Abfrage und bildet dabei die praktische Experimentierplattform in Form eines Prototypen. Die Erforschung solch komplexer Themen ist dabei nicht von einzelnen Arbeitsgruppen zu leisten, sondern bedarf vielmehr der Zusammenarbeit von Wissenschaftlern aus verschiedenen Forschungsbereichen. Die Umsetzung dieses Forschungsvorhabens war das Ziel des von der EU geförderten Projektes VAMPIRE (Akronym für **V**isual **A**ctive **M**emory **P**rocesses and **I**nteractive **RE**trieval)(siehe VAMPIRE Consortium (2002–5)). Das zu entwickelnde mobile Vision-System soll eine Person in einem Raum beobachten und deren Umgebung wahrnehmen, um sie bei Gedächtnisprozessen zu unterstützen. Dabei sollen sowohl Gegenstände als auch Personen und Handlungen im Umfeld des Benutzers wahrgenommen, abspeichert und nach Anfrage zur Verfügung gestellt werden. Zusätzlich soll das System einer Person bei bestimmten Handlungsabläufen durch Hilfestellung assistieren. Hierfür benötigt das notwendige visuelle Gedächtnis einerseits Wissen über die Objekte in der Umgebung und deren Position im Raum, und andererseits ist ein Verständnis über Aktionen des Benutzers und ggf. von weiteren anwesenden Personen erforderlich. Somit sollte das künstliche Gedächtnis Objekt- und Aktionswissen repräsentieren und zur Verfügung stellen können. Das Ablegen des visuell Perzipierten auf unterschiedlichen Abstraktions- und Organisationsebenen in einem klar strukturierten künstlichen Gedächtnis stand dabei im Zentrum der Verknüpfung einzelner Forschungsaufgaben, über die im Kapitel IV ein Überblick gegeben wird. Das Vermitteln von Information als auch die Abfrage von Informationen aus dem Gedächtnis führt zu einem weiteren Schwerpunkt des Projektes. Dieser besteht aus der Entwicklung zweckmäßiger und natürlicher Wege der Mensch-Maschine-Interaktion für ein mobiles System.

Die Gedächtnisprozesse, welche für die Realisierung eines kognitiven Assistentensystems notwendig waren, sind die Basis des anvisierten generellen visuellen Gedächtnisses. Für diese allgemein gefasste Aufgabe wurden weitere Belange im Kontext von Videoannotation eines Tennisszenarios untersucht (Messer u. a. (2005),Christmas u. a. (2005)).

2.2 Funktionale Anforderungen an das System

Die vorliegende Arbeit zielt auf die Entwicklung des Teils des Assistentensystems, welcher als „Brille mit Gedächtnis"bezeichnet werden kann. Diese Brille mit Gedächtnis ist ein computergestütztes mobiles System, welches ein künstliches visuelles Objektgedächtnis enthält, das sowohl interaktiv erweiterbar als auch interaktiv verfügbar sein soll. Die im Gesamtsystem anvisierte Handlungserkennung anwesender Personen steht nicht im Fokus dieser Arbeit. Im Folgenden wird die erforderliche Funktionalität des während dieser Arbeit entwickelten Prototyps einer solchen Brille vorgestellt. Diese Funktionalitätsanforderungen dienen als Grundlage für die technische Realisierung der einzelnen Teilaufgaben, welche in den anschließenden Kapiteln beschrieben werden.

Bei der Entwicklung einer Brille mit Gedächtnis stehen die beiden in der Einleitung erwähnten Aspekte im Mittelpunkt. Einerseits benötigt die Brille ein gemeinsames Verständnis der Umgebung mit dem Benutzer. Andererseits muss es möglich sein, Wissen untereinander auszutauschen, d.h. das System muss auf irgendeine Art und Weise mit dem Benutzer kommunizieren. In dieser Arbeit wurde dabei Wert darauf gelegt, dass die Mensch-Maschine-Kommunikation möglichst natürlich abläuft.

Wie also kommunizieren Menschen miteinander? Zwei Sinnesmodalitäten stehen bei der Kommunikation zwischen Menschen im Mittelpunkt. Zum einen spielt die visuelle Wahrnehmung von Gestik und Mimik eine wichtige Rolle. Zum anderen werden die meisten Informationen durch die akustische Generierung und Wahrnehmung von Sprache ausgetauscht. Hierbei werden bei der Kommunikation zwischen Menschen nicht nur die Inhalte des Gesprochenen, sondern auch Informationen durch die Mimik und den Tonfall, wie z.B. über den aktuellen Gemütszustand, ausgetauscht. Dieser Aspekt soll hier jedoch keine Rolle spielen. Die sachliche Kommunikation über Sprache und Gestik steht hier im Mittelpunkt. Dabei sollen für eine möglichst natürliche Mensch-Maschine-Kommunikation keine zusätzlichen Input-Geräte notwendig sein (Oviatt und Cohen (2000)). Die Brille mit Gedächtnis soll keine Verwendung einer Computermaus oder gar eines Datenhandschuhs wie in anderen Systemen, wie z.B. im Tinmith-Projekt[1], notwendig machen. Des Weiteren soll das System für die Erfüllung der Aufgabe, an den Ort eines verlegten Gegenstandes zu erinnern, keine zusätzliche Information, wie z.B. Miniatursender an allen Gegenständen, verwenden, sondern auf möglichst natürlichem Wege ohne in die Umgebung einzugreifen, arbeiten. Nur die natürlichen Kanäle der visuellen und der akustischen Wahrnehmung sollen dem System einerseits dazu dienen, dem Benutzer Informationen mitzuteilen und andererseits Informationen aus der Umgebung und von dem Benutzer wahrzunehmen. Das Gerät benötigt somit Perzeptionskanäle für visuelle und akustische Signale und die Möglichkeit, visuelle und akustische Informationen dem Benutzer zugänglich zu machen.

Die Kernaufgabe einer mobilen Brille mit Gedächtnis ist es, die Frage zu beantworten, wo sich z.B. der verlegte Schlüssel befindet. Der natürlichste Weg einer solchen Abfrage wäre, verbal nach dem Schlüssel zu fragen. Das System muss also die Möglichkeit haben, akustische Signale aufzunehmen und die Frage aus diesem Signal heraus zu erkennen und

[1] http://www.tinmith.net/tinmith.htm

Abbildung 2.1: Links: Benutzer mit dem mobilen System. Das Kamerabild der zwei kleinen am Helm befestigten Kameras wird in das Display eingeblendet und mit Systeminformationen ergänzt. Zur akustischen Kommunikation dienen ein Funkmikrophon und Kopfhörer. Rechts: In den laufenden Videostreams eingeblendete zuletzt wahrgenommene Position des gesuchten Schlüssels in der Schreibtischschublade.

zu verstehen. Die gestellte Frage muss vom System anschließend in der Art interpretiert werden, dass es nun in seinem Gedächtnis die Information über den Aufenthaltsort des Schlüssels findet und diese dem Benutzer in einer für ihn verständlichen Art und Weise mitteilt. In der natürlichen Mensch-zu-Mensch Kommunikation hieße dies, dass man entweder den Ort verbal mitteilt oder auf den Ort des verlegten Objektes zeigt.

Das hier entwickelte System soll dem Benutzer den Ort des Objektes dadurch zeigen, dass es das gesuchte Objekt in seiner zuletzt wahrgenommenen Umgebung zeigt. Für das Zeigen dieser Information und den gesamten Austausch visueller Informationen wurde ein System verwendet, bei dem der Benutzer einen Helm trägt, auf dem zwei Kameras montiert sind, welche den Blickwinkel des Benutzers aufnehmen und diesen Bilddatenstrom dann über ein mobiles Display, welches sich direkt vor seinen Augen befindet, einblendet (siehe Abb.2.1, links). Dadurch wird ermöglicht, dass Systeminformationen überlagert werden können, wie in diesem Beispiel ein Bild von der zuletzt wahrgenommenen Position des gesuchten Schlüssels in seiner aktuellen Umgebung, der Schreibtischschublade. Diese Information über den Aufenthaltsort des Objektes sollte in der Regel genügen, um die Gedächtnislücke des Benutzers zu schließen. Durch die Positionierung der Kameras entspricht das eingeblendete Bild dem Blickwinkel des Benutzers, so dass er problemlos den Aufenthaltsort des gesuchten Objektes wieder erkennen kann. Für die Aufnahme bzw. Wiedergabe des akustischen Signals wurde ein mobiles Funkmikrofon respektive ein Kopfhörer verwendet. Das gesamte technische Setup ist von

Mitarbeitern des Projektpartners der Technischen Universität Graz[2] entwickelt worden. Technische Details werden in Kap. 4.1.2 beschrieben. Der Schwerpunkt dieser Entwicklung lag dabei nicht auf einer maximal zu erreichenden Miniaturisierung, sondern auf einer rekonstruierbaren, hohen Funktionalität mit der Möglichkeit, weitere Sensoren anzubringen, welche für andere Teilaufgaben bei der Entwicklung des Gesamtsystems, wie die 3-D-Positionserkennung (siehe Teil IV), notwendig sind.

Um den Ort des Objektes auf die in Abb. 2.1 gezeigte Art mitzuteilen, benötigt das System somit ein Gedächtnis über den aktuellen Aufenthaltsort des Schlüssels. Das künstliche visuelle Gedächtnis, welches für die Entwicklung dieses Prototypen entwickelt wurde und später als eine Teil des hierarchisch aufgebauten künstlichen visuellen Gedächtnisses eingehen soll, ist somit ein Bildergedächtnis. Dieses Bildergedächtnis muss so aufgebaut sein, dass die Bildinformation über die verbale Frage nach dem Objektnamen zugreifbar sein muss. Das Bildergedächtnis besteht somit aus einem gelabelten Bildspeicher.

Dieses visuelle Objektgedächtnis muss dem System vermittelt werden, was ebenfalls auf möglichst natürliche Art und Weise passieren soll. D.h., das System soll die Objekte in der Umgebung des Benutzers wahrnehmen können und erlernen, um welche Objekte es sich handelt. Die Information, welches das System für die Erstellung des Objektgedächtnisses und somit für das Erlernen von Objekten verwendet, soll ausschließlich der Bilddatenstrom der beiden Kameras liefern. Daher bietet sich die Verwendung eines ansichtsbasierten Objekterkenners an. Im Gegensatz zu anderen Objekterkennungsansätzen benötigt so ein System keine 3-D-Oberflächenmodellierung oder geometrische Modelle um Objekte neu zu erlernen, sondern kann die Information zum Erlernen neuer Objekte ebenfalls aus dem Bildstrom gewinnen, wodurch ansichtsbasierte Objekterkenner komfortabel zu nutzen sind (vgl. Koenderink und van Doorn (1979); Murase und Nayar (1995); Mel (1997)).

Aus diesem Bildstrom muss nun dem System mitgeteilt werden können, wo im Bild sich welche Objekte aufhalten. Es muss also möglich sein, dem System ein identisches Verständnis von Gegenständen in der Umgebung zu vermitteln. Die Frage nach dem „Wo befinden sich Objekte im Bild?", kann durch eine Aufmerksamkeitssteuerung erlangt werden, die auf ähnliche Reize im Bild reagiert, die auch beim Menschen zu einem genauen Hinschauen führen. Details zur Generierung der visuellen Aufmerksamkeit finden sich in Kapitel 5. Diese Aufmerksamkeitssteuerung soll dazu führen, dass dem System mögliche Orte von Gegenständen im Bild vorliegen. Diese Orte müssen auf geeignete Objekten untersucht werden und die entdeckten Objekte erkannt werden. Diese Information muss anschließend in das visuelle Bildergedächtnis überführt werden. Wenn Objekte im Bild nicht bekannt sind, muss das System fähig sein, diese neu zu lernen. Für einen ansichtsbasierten Objekterkenner, welcher auf einer mobilen Plattform arbeiten soll, ist es dabei notwendig, Objekte sowohl in unterschiedlicher Skalierung als auch in verschiedenen Ansichten, also aus verschiedenen Perspektiven, zu erlernen.

Für das Erlernen neuer Objekte soll das Wissen des Benutzers „angezapft" werden. Der Benutzer soll hier als Experte fungieren, um dem System die notwendige Information

[2] http://www.emt.tu-graz.ac.at/ pinz/

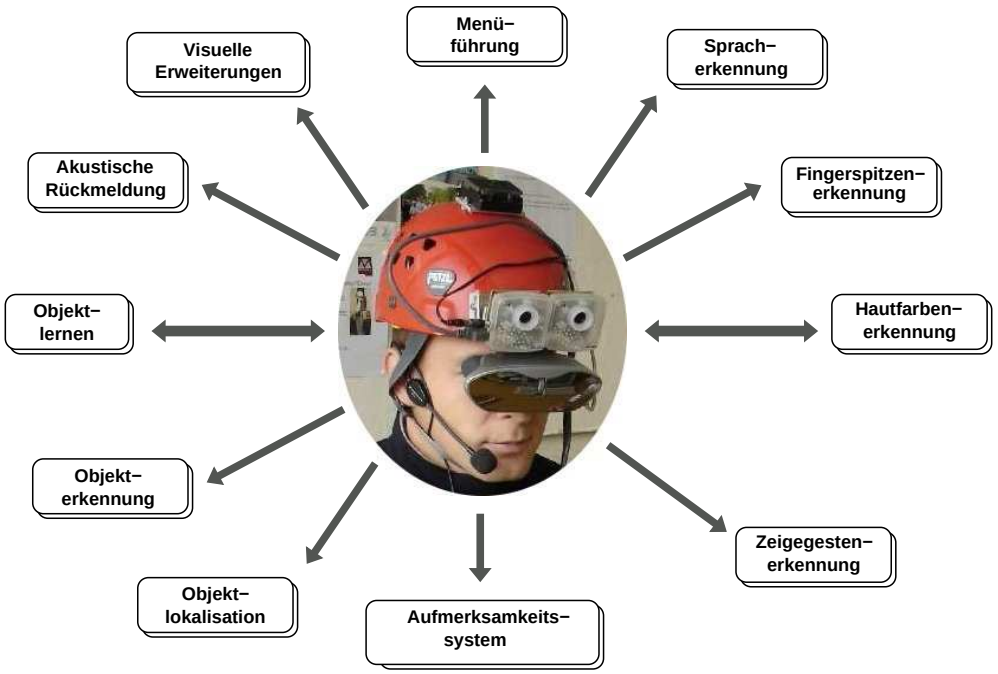


Abbildung 2.2: Notwendige Funktionalitäten für die Realisierung einer Brille mit Gedächtnis.

zu vermitteln. Das Erlernen neuer Objekte soll dabei so natürlich und komfortabel wie möglich geschehen.

Hierfür sollen zwei Möglichkeiten zur Verfügung gestellt werden. Die eine Möglichkeit besteht darin, dem System direkt Objekte in verschiedenen Ansichten zu präsentieren und diese dann per Spracheingabe zu benennen. Andererseits soll das System geeignete Objektansichten selbstständig über die Aufmerksamkeitssteuerung aufnehmen, welche dann in einer Interaktionsschleife mit dem Benutzer benannt werden können.

Für die erste Möglichkeit ist es notwendig, dass das System erkennt, welches Objekt der Benutzer dem System beibringen möchte. Das Objekt muss also von dem Benutzer referenziert werden. Um dies zu ermöglichen, soll das System Zeigegesten erkennen, so dass der Benutzer durch Zeigen mit dem Finger ein Objekt referenzieren kann. Anschließend sollen dem System verschiedene Ansichten des Objektes präsentiert und anschließend benannt werden.

Da die Performanz eines so trainierten Objekterkenners auf einer geringen Anzahl von Ansichten weniger Objekte beruht, soll diese gesteigert werden, indem komfortabel eine größere Trainingsmenge erstellt werden kann. Dazu sollen automatisch, nur über das künstliche Aufmerksamkeitssystem gesteuert, mögliche Kandidaten für Objekte aufgenommen werden. Diese umfangreiche Bildersammlung von Objekten muss komfortabel für den Benutzer zu benennen sein.

Anschließend kann der Objekterkenner sein Wissen erweitern, indem er mit den neu gewonnenen, gelabelten Daten weiter lernt.

Diese Vielzahl an benötigten Funktionen muss dabei dem Benutzer leicht verständlich verfügbar gemacht werden. Hierfür soll ein Funktionsmenü entwickelt werden, welches über das Display angezeigt wird. Dieses Menü soll einerseits über die Sprache zu bedienen sein und andererseits bei Problemen mit der Spracherkennung, z.B. bei starken Umgebungsgeräuschen, dadurch robuster gemacht werden, dass es auch über Fingerbewegungen zu steuern ist.

Robustheit soll das System ebenfalls dadurch erlangen, dass sowohl die Spracherkennung als auch die Gestenerkennung an sich ändernde Umgebungen angepasst werden können.

Diese Vielzahl an benötigten Fähigkeiten wird in Abb. 2.2 veranschaulicht.

Kapitel 3

Stand der Forschung

Nachdem vorausgehend die Entwicklungsanforderungen an eine „Brille mit Gedächtnis" vorgestellt wurden, soll nun der Stand der Forschung zu den hierzu relevanten Bereichen dargestellt werden. Die Problemkomplexität bei der Entwicklung einer Brille mit Gedächtnis spiegelt sich in der Vielfalt der Informatikdisziplinen wieder, die dabei berührt werden. Sie reichen von Maschinellem Sehen über Mensch-Maschine-Interaktion, künstliches Lernen mittels Neuronaler Netze, künstliche visuelle Gedächtnisse bis zur Sprach- und Zeigegestenerkennung.

In der vorliegenden Arbeit liegen die beiden Schwerpunkte auf der Mensch-Maschine-Interaktion und der Lernfähigkeit eines computergestützten visuellen Systems.

Das zentrale Fachgebiet in dieser Arbeit ist das Maschinelle Sehen, i.e. die **Computer Vision**. Computer Vision befasst sich mit der Be- und Verarbeitung von natürlichen und künstlichen Bildern mit Hilfe leistungsfähiger Computer. Die Grundlage der digitalen Bildverarbeitung sind hochdimensionale Daten. In modernen Systemen resultiert diese hohe Dimensionalität aus der Verwendung hochauflösender Bildern mit einer hohen Farbtiefe, die spätestens bei der Bearbeitung von Bildfolgen, beispielsweise Videosequenzen, an die Leistungsgrenzen auch moderner Rechner schon bei vergleichsweise einfachen Operationen stößt. Neben der hohen Dimensionalität der Datenbasis wächst die Problemkomplexität der zu lösenden Aufgaben im Bereich Computer Vision durch die hohe Mannigfaltigkeit aufgrund der unendlichen Anzahl von Ansichten, Skalierungen und Beleuchtungssituationen. In der Industrie finden daher vor allem Anwendungen Gebrauch, die aufgrund einer sehr eingeschränkten Domäne handhabbar sind, wie z.B. Materialüberwachungen in Produktionsstätten oder die Überwachung von Schweißnähten an Flugzeugen (Liu u. a., 2008). Einen Überblick über industrielle Anwendungen findet man bei Malamas u. a. (2003).

Im Gegensatz zu diesen industriellen Anwendungen bestehen die Herausforderungen der Zukunft an neue Technologien jedoch daraus, dass sie den Alltag des Menschen erleichtern sollen und nicht nur für einzelne Aufgaben eingesetzt werden können. Moderne Systeme sollen in einer gewissen Weise mitdenken und den Anforderungen entsprechend reagieren. Die Entwicklung solcher Systeme können verschiedenster Art sein. Ein Beispiel dafür sind die intelligenten Räume (Wisneski u. a., 1998), in denen der Wohnkomfort durch „mitdenkende", den Benutzer umgebende Medien in Möbeln und Wänden durch eine Vielzahl von Anwendungen gesteigert wird (sogenannte *ambient intelligence*, siehe Mahesh S. Raisinghani und Schmedding (2004)). So können bei Bedarf physisch erfassbare Verbindungen mit entfernten Personen hergestellt werden; die Sicherheit der

Gebäude vor Einbrechern kann mit Hilfe von Gesichtserkennung des Bewohners erhöht werden; der Kühlschrank kann sich durch autonome Bestellung selbst füllen, etc.

Im Bereich der Computer Vision versucht man bei der Entwicklung solcher Technologien für den alltäglichen Gebrauch der hohen Problemkomplexität in natürlichen Umgebungen dadurch zu begegnen, dass einerseits die Bildinformationen auf das wesentliche beschränkt werden und andererseits eine höhere Flexibilität durch die Verwendung von artifizieller Lernfähigkeit gewährleistet wird. Die Einschränkung erreicht man, indem man geschickt die benötigten Merkmale aus den gegebenen Daten extrahiert (Beispiele hierfür finden sich in Kap. 5). Im Folgenden soll besonders der zweite Aspekt der künstlichen Adaptionsfähigkeit von visuellen Systemen betrachtet werden, welcher in den Bereich der *Cognitive Vision* fällt.

3.1 Kognitives Sehen

Der Begriff *Cognitive Vision* wurde durch das Bestreben geprägt, Computer Vision-Systeme durch die Verwendung kognitiver Fähigkeiten robuster, flexibler und lernfähiger zu machen. Cognitive Vision-Systeme sollen sich robust an unvorhersehbaren Veränderungen der visuellen Umgebung anpassen können und das Auftreten von Objekten oder Ereignissen antizipieren (nach Vernon (2005)).

Cognitive Vision besteht aus einer Kombination von Computer Vision und **Kognition** (Vernon, 2008). Im Folgenden wird geklärt, was unter Kognition verstanden wird und welche Bedingungen erfüllt werden müssen, um ein künstliches, kognitives System zu realisieren.

Zur Erklärung des Begriffs Kognition wird nach Vernon (2008) zwischen der kognitivistischen und der emergenten Position unterschieden, bevor anschließend die beiden Ansätze auf ihre Notwendigkeit zur Verkörperung von kognitiven Systemen hin überprüft werden. Nach der kognitivistischen Position umfasst Kognition die Manipulation von expliziten Repräsentationen eines Zustands oder Verhaltens der externen Welt, um angemessene, erlernte, antizipatorische und effektive Interaktionen zu ermöglichen. Das hieraus hervorgegangene, gespeicherte Wissen ermöglicht in zukünftigen Situationen ein begründetes Schlussfolgern. Für diese Sichtweise der Kognition ist die Verkörperung keine notwendige Voraussetzung für ein künstliches System. Im Gegensatz dazu zeigt Vernon, dass für die emergente Position Kognition einer Verkörperung des Systems bedarf. Nach der emergenten Position ist Kognition ein Prozess von Selbstorganisation, wobei sich das System kontinuierlich in Echtzeit aufgrund der System-Umgebungsinteraktion rekonstituiert. Für die Verkörperung entscheidend ist dabei die Fähigkeit zur Interaktion und Perzeption.

Eine ähnliche Sichtweise vertritt auch Christensen (2003), der Kognition als das Generieren von Wissen, basierend auf bestehenden Modellen, Lernen, logischem Denken oder Schlussfolgern und der Perzeption versteht. Demnach ist Kognition nicht nur ein passiver, sondern auch ein aktiver Prozess, der sowohl Kommunikation als auch Interaktion mit der Umgebung beinhaltet. Die Aufnahme, das Speichern und das Abfragen von Wissen sind dazu notwendige Komponenten. Ohne Perzeption und der Möglichkeit,

das Wahrgenommene zu verarbeiten oder zu manipulieren, kann kein Wissen aufgebaut werden. Neben der Möglichkeit des Lernens impliziert kognitive Vision eine Art Aufmerksamkeitskontrolle und eine Möglichkeit, die Prioritäten bei der Speicherung von Wissen zuzuordnen. Dies erfordert kontextabhängiges Schlussfolgern und Kategorisieren, also eine Art kontextuelles Bewusstsein. Die Voraussetzungen hierfür können nach Christensen, ebenso wie bei Vernon, nur erfüllt werden, wenn das System in irgendeiner Art und Weise verkörpert wird.

Diese Verkörperung eines kognitiven Systems bezieht sich auf die Annahme, dass das System eine Form von Körper mit einer sensomotorischen Apparatur besitzt, durch welche es in einer virtuellen oder realen Umwelt perzipieren und operieren kann (Ziemke, 2005; Anderson, 2003a,b).

Diese Verkörperung kann in Form von Robotern oder anderen mechatronischen Teilen gegeben sein. Eine weitere Möglichkeit, die eben genannte Voraussetzung des sogenannten Perzeptions-Aktions-Kreislaufes zu erfüllen, kann nach Bauckhage u. a. (2005) durch Interaktion mit einem mobilen System erreicht werden.

Das Entwicklungsziel der vorliegenden Arbeit, eine „Brille mit Gedächtnis" zu konstruieren, impliziert automatisch das Verwenden eines mobilen Systems. Der Benutzer steht bei dem Lernprozess des Systems in einer Art Rückkopplungs- oder Interaktionsschleife, in der das System von dem Wissen des Benutzers einerseits profitieren soll und andererseits Wissenslücken des Benutzers ausgleichen kann (im Englischen oft als *human in the loop* bezeichnet, vgl. Bauckhage u. a. (2005)). Bei fehler- oder lückenhaften Kenntnissen des Systems kann der Benutzer korrigierend eingreifen.

Sollen die kognitiven Aspekte der Interaktionsfähigkeit und des Lernens, also dem Generieren von Wissen aus Modellen heraus, in solch ein System integrierbar sein, muss das System einerseits die Umgebung in vergleichbarer Art und Weise wie der Benutzer wahrnehmen und andererseits dem Benutzer das eigene Wissen vermitteln können. Eine Möglichkeit, sowohl die von dem Benutzer wahrgenommene visuelle Information wahrzunehmen als auch sogleich visuell künstlich Information dem Benutzer anzuzeigen, bietet die sogenannte Augmented Reality.

3.2 Erweiterte Realität - Augmented Reality

Der Begriff der Augmented Reality (abgekürzt AR), also der Erweiterten Realität, grenzt sich von dem Begriff der Virtual Reality, der virtuellen Realität (abgekürzt VR), ab. Die VR projiziert den Benutzer in eine künstliche virtuelle Welt. VR-Entwicklungen sind vor allem für die Spieleentwicklung von großem Interesse, wodurch der Begriff deutlich stärker verbreitet ist als die AR.

In der AR wird die eigentliche Sinneswahrnehmung der realen Umgebung des Benutzers durch künstlich erzeugte Überlagerungen erweitert. Diese künstlichen Erweiterungen können sich auf alle Perzeptionskanäle beziehen. Am weitesten verbreitet ist jedoch die visuelle Verwendung. Die überlagerte Information kann dabei sowohl aus den natürlichen menschlichen Perzeptionskanälen resultieren als auch für Menschen nicht zugängliche Informationen visualisieren. Der Benutzer nimmt also die reale Welt wahr

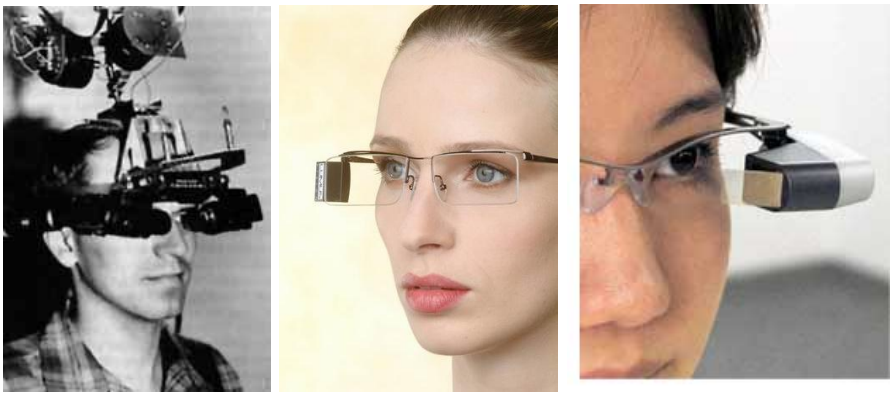


Abbildung 3.1: Links: Ivan Sutherland mit dem ersten HMD. Mitte und rechts: Moderne mobile Displays, welche an handelsüblichen Brillengläsern montiert werden können (Mitte von Limus, rechts von Brother).

und bekommt zusätzlich von einem künstlichen computergestützen System Informationen, die den Benutzer bei bestimmten Prozessen unterstützen. Dabei werden vor allem 2- und 3-D-Modelle visuell überlagert, die so mit der realen Welt koexistieren.

Meist geschieht dies mittels am Kopf befestigter oder in der Hand gehaltener Displays. Das erste AR-Interface wurde von Ivan Sutherland 1965 entwickelt. Er entwickelte zwei am Kopf tragbare Miniaturbildröhren in Verbindung mit einem mechanischen Tracker. Dieses erste am Kopf befestigte Display (HMD - engl. für *Head Mounted Display*) konnte das virtuelle Drahtmodell eines Würfels der realen Welt überlagern. Abb. 3.1 zeigt den Prototypen im Vergleich zu modernen Displays.

Im Gegensatz zur VR bieten AR-Anwendungen verschiedene Vorteile. Kiyokawa (2000) verglich die Performanz von VR mit AR mit Hilfe einer einfachen Zeige-Aufgabe. Die Benutzer mit AR waren signifikant schneller, da sie sich mit Hilfe von nicht verbaler Kommunikation besser verständigen konnten. In der AR ist der Arbeitsraum, z.B. bei dem Hantieren von Objekten auf einem Tisch, der gleiche wie der Kommunikationsraum. Billinghurst und Poupyrev (2000) machten einen Versuch mit einer einfachen Puzzleaufgabe unter drei verschiedenen Konditionen: *a)* Zusammenarbeit in natürlicher Umgebung einander gegenüber sitzend, *b)* Zusammenarbeit mittels AR mit virtuellen Objekten und *c)* gemeinsames Arbeiten an virtuellen Objekten, die auf einen Bildschirm projiziert waren. Auch wenn die Benutzer die Bedingungen als stark unterschiedlich empfanden, so verwendeten sie doch die annähernd gleiche non-verbale Kommunikation bei der Verwendung der AR. Auch die deiktischen Sprachmuster waren in der AR-Umgebung und in der realen Umgebung identisch. Die Manipulation von Objekten wurde in realer und in der AR-Umgebung als gleich leicht empfunden. Billinghurst untersuchte ebenfalls das Empfinden von Versuchspersonen bei einer AR-Version von Telekonferenzen, in der die abwesende Person in 3-D eingeblendet war (Billinghurst und Kato, 2000). Auch hier stellte Billinghurst fest, dass non-verbale Kommunikation im Gegensatz zur 2-D Telekonferenz aufgrund der wahrgenommenen Präsenz des Gegenübers verstärkt eingesetzt

und registriert wurde. Des Weiteren ist die Anzahl der Kommunikationspartner nicht so eingeschränkt.

Weitere Vorteile der Verwendung von AR-Systemen im Kontext der Kollaboration wurden in dem Projekt *Studierstube* sichtbar. Dabei wurde das kollaborative Arbeiten an virtuellen 3D-Objekten untersucht. Sie benannten die folgenden fünf wesentlichen Vorteile von kollaborativem Arbeiten mittels AR-Apparaturen:

- Virtualität: Objekte, die nicht existieren, können untersucht und bearbeitet werden;
- Erweiterung: reale Objekte können durch virtuelle Annotation erweitert werden;
- Kooperation: Benutzer können sich sehen und miteinander arbeiten;
- Unabhängigkeit: jeder Benutzer kontrolliert seine Blickrichtung unabhängig von den anderen;
- Individualität: überlagerte Information kann für jeden einzelnen Benutzer unterschiedlich sein.

Diese Vorteile bieten die Möglichkeit einer Vielzahl von Anwendungen. Die meisten davon sind jedoch auf die Forschung eingeschränkt und nur wenige haben bisher kommerzielle Anwendung gefunden. Eine Beispielanwendung entwickelten Billinghurst und Kato (2001) mit dem „Magic Book". Dieses Buch im normalen Format ermöglicht mittels eines in der Hand gehaltenen AR-Displays Szenen des Buches realer durch Einbettung von 3-D Szenen zu erleben. Ein weiteres Feature ermöglicht es, dass verschiedenen Personen das Buch gemeinsam betrachten, jede aus dem eigenen Blickwinkel und jede Person wird von den anderen als eine virtuelle Person am Rande wahrgenommen.

Der Vorteil für den Benutzer, nicht sichtbare Informationen der Realität visuell zu überlagern, bietet in vielen Anwendungsbereichen neue Perspektiven. Eine der ersten und besonders zweckmäßigen Anwendungsfelder der AR ist die Medizin. Bereits 1997 stellte Azuma in seinem Überblick über AR-Anwendungen eine medizinische Anwendung vor, in der einem Chirurgen bei einer Operation durch ein semi-transparentes Display die Möglichkeit gegeben wurde, Röntgenaufnahmen der behandelten Organe ortsgetreu eingeblendet zu bekommen und so den Operationserfolg durch für ihn sonst nicht sichtbare Information zu optimieren (Azuma, 1997a). Ein fortgeführtes Projekt des BMBF, MEDARPA, ermöglicht minimalinversive Herzchirurgie, in der schwere Operationen durch die zusätzlich eingeblendete Information von CT- und Röntgenaufnahmen ohne das Öffnen des Brustkorbes durchgeführt werden können (siehe Abb. 3.2[1]).

Weitere Anwendungen finden sich im Flugzeugbau, um Leitungen im Flugzeug zu verlegen (Curtis u. a., 1998) oder um für Bauprojekte Strukturen hinter Wänden zu visualisieren, wie im Vibal-Projekt des Fraunhofer IAIS [2].

[1] aus http://www.igd.fraunhofer.de/igd-a7/projects/medarpa/medarpa.html
[2] http://www.iais.fraunhofer.de/805.html


Abbildung 3.2: Beispielhafte Anwendungen von AR. Links: Das MEDARPA- System assistiert einem Chirurgen bei minimalinversiver Herzchirugie. Rechts: Das System des Projektes *Studierstube* dient als Navigationshilfe und Stadtführer.

Eines der größten Probleme all dieser Anwendungen ist das Ausrichten bzw. Abgleichen der virtuellen Welt bzw. der künstlichen Erweiterungen mit der realen Welt im dreidimensionalen Raum (von Azuma (1997b) als *registration issue* bezeichnet).

In dem Projekt STARMATE wird durch sehr exaktes Tracking und Nutzen von Laserpointern als Zeigern versucht, dieses Problem zu reduzieren (Schwald u. a., 2003). Doch selbst mit modernster 3D-Trackertechnologie bleibt dieses Problem bestehen. Wenn hingegen die Perzeptionskanäle des künstlichen Systems die gleichen sind, wie die des Benutzers, kann ein Großteil des Problems der Ausrichtung von realer und erweiterter Welt umgangen werden. Ein Beispiel hierfür ist das bereits genannte Projekt Studierstube (Reitmayr und Schmalstieg, 2004; Schmalsteig, 1996). Die hier verwendete Apparatur, in der dem Benutzer das Bild von zwei über seinen Augen befindlichen Kameras in ein mobiles Display vor seinen Augen übertragen wird, ist Vorbild bzw. Vorläufer für das in dieser Arbeit entwickelte System (siehe Kap. 4.1.2). Abb. 3.2 zeigt das verwendete System als Navigationshilfe und Stadtführer.

3.3 Interaktives Objektlernen

In den beiden letzten Abschnitten wurden die Notwendigkeit der Verkörperung für ein kognitives System aufgezeigt und die Anwendungsmöglichkeiten und Vorteile von AR-Systemen wiedergegeben. Betrachtet man beide Aspekte gemeinsam, ergibt sich, dass die Voraussetzungen von Kognition im Sinne der Verkörperung durch die Verwendung eines mobilen AR-Systems erfüllt werden können. Für ein AR-System, welches die gleiche Blickrichtung wie der Benutzer verwendet, wie in Abb.3.2 des Projektes Studierstube gezeigt, führt Hanheide den Begriff *Ego-Vision-System* (EVS, Hanheide (2006)) ein. Da hier die visuelle Perzeption von Mensch und Maschine einander entspricht, kann dieser Kanal ebenfalls für die Einblendungen von Informationen verwendet und somit

die Kommunikation zwischen Mensch und System erleichtert werden. Durch die Aktivität des Benutzers wird der Blickwinkel des Systems gesteuert. Wird zusätzlich vom System auf die sich wechselnden Bedingungen reagiert, z.B. durch das automatische Anpassen von Parametern an variierende Beleuchtungssituationen, wird das System als *active vision system* bezeichnet. Durch den Aufbau eines EVS wird die Konstruktion eines kognitiven Systems ermöglicht, in welchem der Mensch, wie oben beschrieben, in einer Interaktionsschleife die Lernfähigkeit des künstlichen Systems steigert und steuert. Diese Adaptionsfähigkeit kann sich dabei sowohl auf die Anpassung an sich ändernde Verhältnisse als auch auf das Erlernen von neuen Gegebenheiten beziehen.

In der vorliegenden Arbeit soll ein System entwickelt werden, dessen Adaptionsfähigkeit sich insbesondere auf die Erkennung und das Erlernen von Objekten bezieht und welches fähig sein soll, interaktiv zur Laufzeit neue Objekte zu erlernen.

Das Erlernen und Erkennen von Objekten ist für den Menschen eine so selbstverständliche Fähigkeit, dass es einem kaum bewusst wird, wie robust diese Aufgabe selbst in schwierigen Situationen auf eine Art und Weise erledigt wird, die bisher nicht ansatzweise von künstlichen Systemen erreicht werden kann und somit noch zu den großen Herausforderungen künstlicher intelligenter Systeme gehört.

Die künstliche **visuelle Objekterkennung** verfolgt dabei grob zwei Richtungen bezüglich der verwendeten Wissensrepräsentation: einerseits modellbasiert und andererseits ansichtsbasiert (Riesenhuber und Poggio, 2000b). Grundlage der ersten Vorgehensweise sind vorher erlernte geometrische 2- oder 3-D-Modelle von Objekten, welche bei dem Erkennungsvorgang mit dem real aufgenommenen Bild verglichen werden und so bei hinreichender Übereinstimmung zur Erkennung von Objekten führen. Der bekannteste Ansatz basiert dabei auf einer Objektrepräsentation, welche aus einer Dekomposition der Objekte in basale geometrische Formen besteht, wie bei Hummel und Biederman (1992). Der Vorteil des modellbasierten Ansatzes besteht darin, dass durch die vorhandenen Modelle die Erkennungsleistung weitgehend unabhängig von dem jeweiligen Blickwinkel und der Skalierung ist. Der Nachteil liegt allerdings in der sehr aufwändigen Erstellung exakter 3D-Modelle. Des Weiteren ist die Erkennungsleistung dadurch begrenzt, dass die Modelle in einer künstlichen laborähnlichen Umgebung erstellt werden und sich somit für natürliche Umgebungen und einer teilweisen Verdeckung nur bedingt eignen. Eine Möglichkeit, dieses Handicap zu umgehen, zeigt Mian u. a. (2006), indem er zwar ebenfalls offline unter großem Rechenaufwand 3D-Modelle erstellt, diese jedoch auf Tensoren basieren, die eine Erkennung selbst in teilverdeckten Szenarien ermöglicht. Den Nachteil der aufwändigen Modellerstellung versuchten Khan u. a. (2007) durch einen hybriden Ansatz aus modellbasierter und ansichtsbasierter Objekterkennung zu umgehen, indem sie durch homographische Transformation ihre geometrischen Modelle aus Trainingsansichten von Objekten erstellten.

Der zur Zeit dominierende Ansatz ist die ansichtsbasierte Objekterkennung (vgl. (Belongie u. a., 2001; Carmichael und Hebert, 2002; Viola und Jones, 2001). Erste wertvolle Beiträge aus dem Bereich der Gesichtserkennung lieferten Turk und Pentland (1991). Ein generellerer Ansatz zur ansichtsbasierten Objekterkennung wurde von Murase und Nayar (1995) vorgestellt. Der große Vorteil der ansichtsbasierten Objekterkennung ist darin zu

sehen, dass man ohne eine Modellierung auskommt und direkt über die aufgenommenen Bilder die Trainingsmenge für das Objekterkennungssystem gewinnt. Dadurch ist die Beschreibung durch ein geometrisches Modell nicht notwendig, und schon einzelne Ansichten können von Anfang an zum Erlernen neuer Objekte verwendet werden. Die hohe Mannigfaltigkeit von Objektansichten, wie Verdeckungen und Beleuchtungsvariationen, werden direkt über die Bildtrainingsmenge mitgelernt.

Das einfachste und zugleich aufwändigste Verfahren, welches heute vor allem zu Vergleichen der Klassifikationsperformanz herangezogen wird, ist der *nearest-neighbor*-Klassifikator. Dieser arbeitet auf dem gesamten zu betrachtenden Bildausschnitt im Rohformat, also mit der Dimension des Bildes (*Pixelbreite* x *Pixelhöhe* x *Anzahl der Farbkanäle*), wodurch er einen sehr hohen Bedarf an Speicherplatz und Rechenzeit hat. Die Funktionsweise beruht darauf, dass ein zu klassifizierender Bildausschnitt mit allen Bildausschnitten seiner Datenbasis verglichen wird und der Klasse zugewiesen wird, von welcher der Auschnitt den minimalsten euklidischen Abstand zu einem der Bilder aus der Datenbasis besitzt. Neben dem Nachteil des hohen Rechen- und Speicherplatzaufwands ist das Problem bei der Verwendung dieser Art von Klassifikation, dass für gute Klassifikationsergebnisse eine sehr umfangreiche Datenbasis zur Verfügung stehen muss, um die Vielzahl von Objektansichten durch unterschiedliche Skalierung, Translationen, Rotationen und Änderungen der Lichtverhältnisse abzudecken.

Um diese Probleme zu umgehen, sucht man nach geeigneten Repräsentationen der Bilder von Objekten, um die für ein Objekt charakteristischen Merkmale zu extrahieren. Einerseits geht man dabei rein künstliche Wege, andererseits benutzt man die Erkenntnisse aus der Neurophysiologie des Menschen, um die Art und Weise, wie der Mensch diese Aufgaben erledigt, zumindest vom Grundprinzip her zu rekonstruieren.

Eine der prominentesten Ansätze entwickelte Lowe (1999a) mit den sogenannten SIFT-Features (engl. für *Scale Invariant Feature Transform*). Diese Bildmerkmale haben ähnliche Eigenschaften wie die Neuronen des inferioren Temporallappens, welche bei Primaten für die Objekterkennung zuständig sind und eine von der Skalierung weitgehende unabhängige Repräsentation ermöglichen.

Die Merkmalsextraktion wird zunehmend in Objekterkennungsarchitekturen integriert. Die neurophysiologischen und psychophysikalischen Erkenntnisse über die visuelle Verarbeitung bei höheren Vertebraten, welche nach Hubel und Wiesel (1959) in einem hierarchischen Netzwerk organisiert wird, sind die Grundlage von aktuellen Forschungsansätzen. Das hierarchische Neuronennetz besteht demnach aus einzelnen Schichten. Zum einen sind dies Schichten aus den sogenannten *simple cells*, welche als einfache Merkmalsextraktoren für Balken in verschiedenen Ausrichtungen agieren. Die Information dieser Schichten wird in den darauffolgenden Schichten aus den *complex cells* weiter verarbeitet. Als Vorreiter für biologisch motivierte hierarchische Feedforward-Modelle gilt Fukushima (1980). Er stellte mit dem *Neocognitron* den ersten künstlichen Objekterkenner vor, welcher mit künstlichen neuronalen Netzen diese Struktur und Funktionsweise rekonstruierte. Riesenhuber und Poggio (2000a) entwickelten diesen Ansatz mit dem HMAX-Modell weiter, welches als Grundlage für die Arbeiten von Wersing und Körner (2003) einging.

Auf diesen Modellen basierend entwickelte Bax (2007) parallel zu dieser Arbeit im Rahmen des VAMPIRE-Projektes ein generalisiertes Hierarchisches Feedforward-Objekterkennungsmodell, welches eine Vielzahl an Objekten aufgrund von Bildausschnitten selbst bei Verdeckungen, unterschiedlichen Skalierungen und Translationen und vor einem völlig konfusen Hintergrund mit einer hohen Performanz erkennt. Als Erweiterung zeigte er, dass die Flexibilität des Modells unter Zuhilfenahme einer zusätzlichen Schicht es ermöglicht, simultan neben der Objekterkennung mit hoher Performanz unter schwierigen Bedingungen ebenfalls die Objekte zu lokalisieren.

Bei den meisten künstlichen Objekterkennungsansätzen geht der eigentlichen Anwendung eine rechenaufwändige Lernphase des Klassifikators mit Hilfe einer großen Datenbasis als Trainingsmenge voraus. Im Vergleich dazu wird Objektwissen beim Menschen nach den heutigen Kenntnissen in der Neurologie und Psychologie etappenweise abgespeichert. Das verbreitetste Gedächtnismodell geht bereits auf Hebb (1949) zurück und teilt das Gedächtnis in Kurzzeitgedächtnis (STM engl. für *short term memory*) und Langzeitgedächtnis (LTM, engl. für *long term memory*) ein, welche sich durch unterschiedliche Dauer der Speicherung von Entitäten und der Kapazität unterscheiden. Vorgeschaltet befindet sich das sogenannte sensorische Gedächtnis, welchem allerdings noch keine bewussten Gedächtnisprozesse zugeschrieben werden. Das STM ermöglicht schnelles Lernen schon mit einer geringen Wissensbasis. Das Erinnerungsvermögen ist allerdings nur von kurzer Dauer und es können nur eine geringe Anzahl von Gedächtnisinhalten gespeichert werden. Das STM wird auch als Arbeitsgedächtnis bezeichnet und teilt sich wiederum in ein zentrales Kontrollsystem auf, welches von dem visuell-räumliche Skizzenblock und der phonologischen Schleife unterstützt wird (Baddeley, 2003). Der visuelle Teil des STM ermöglicht das sofortige Erkennen von Objekten nach nur einmaligem Präsentation (Norman und O'reilly, 2003). Diese Art des Objektlernens, bei denen nur eine Ansicht eines Objektes zum Lernen verwendet wird (im engl. *one-shot-learning*), führte zu den ersten technischen Ansätzen des sogenannten **Online-Lernens** (Carpenter u. a., 1991). Das Online-Lernen bezieht sich auf die Fähigkeit eines künstlichen Systems, zur Laufzeit neues Objektwissen zu erlangen und anwenden zu können. Das Online-Lernen ermöglicht eine interaktive Kontrolle und Korrektur des Lernprozesses zur Laufzeit. Um dies zu ermöglichen, benötigen solche Lernsysteme neben der Klassifikationsfähigkeit die Fähigkeit der Interaktion zwischen Mensch und Maschine.

Steels und Kaplan (2000) präsentierten ein Beispiel für einfaches Objektlernen durch verbale Mensch-Maschine-Interaktion, indem sie dem Roboterhund Aibo drei simple Objekte beibrachten. Durch die einfache Struktur und Farbe der Objekte war es möglich, diese online auf Basis von geometrischem Hashing zu erlernen. Ein weiteres Beispiel für interaktives Objektlernen stellte Makihara vor, in dem ein Roboter aufgefordert wird, bestimmte Objekte aus einem Kühlschrank zu holen. Ist das Objekterkennungsergebnis falsch, werden durch direkte verbale Kommunikation mit dem Roboter die Parameter des Objekterkennungssystems auf Basis von Farbfiltern online angepasst (Makihara u. a., 2004, 2005). Diese Ansätze weisen jedoch nur eine geringe Kapazität des Objektgedächtnisses auf.

Beim Menschen muss für die anhaltende Speicherung von einer Vielzahl von

Gedächtnisinhalten das Wissen vom STM in das LTM überführt werden, welches nahezu unendliche Kapazitäten aufweist. Die Überführung benötigt jedoch Zeit, und Untersuchungen weisen darauf hin, dass diese Überführung vor allem im Schlaf passiert (Maquet, 2001).

In dieser Arbeit soll ebenfalls zwischen robustem und umfangreichem Langzeitgedächtnis und schnell erlernbarem Kurzzeitgedächtnis unterschieden werden (siehe Teil III). Das System kopiert die Funktionalität des menschlichen Gedächtnisses in Bezug auf die Funktion. D.h., dass es auf Basis weniger Ansichten Objekte neu erlernen kann und ihm dieses Wissen schnell zur Verfügung steht. In beiläufigen Prozessen wird dann, wenn das System nicht voll ausgelastet ist, das LTM erstellt, welches eine deutlich größere Anzahl an Objekten robust erkennt. Die beiden Gedächtnisprozesse ergänzen sich jedoch nicht inkrementell. Eine Weiterentwicklung dieser Idee stellten Kirstein u. a. (2008) mit einem Objekterkennungssystem vor, welches online durch Interaktion seine Objekterkennungsperformanz erhöhen kann und dabei inkrementell auf den vom STM erstellten Repräsentationen von Objekten das LTM aufbaut.

Kapitel 4

Entwicklungsumgebung

Im vorangegangenen Kapitel ist der Stand der Forschung dargelegt worden. Darauf aufbauend wird nun die Entwicklungsumgebung des im Rahmen dieser Arbeit entstandenen Systems vorgestellt. Die Entwicklungsumgebung besteht im Einzelnen aus dem verwendeten Szenario, der Hardware sowie der verwendeten Software-Entwicklungsumgebung.

4.1 Szenario und Systemaufbau

Für die Entwicklung eines trainierbaren Objekterkenners für alltägliche Anwendungen sollte eine Umgebung gewählt werden, die sowohl rekonstruierbar ist als auch verschiedene Möglichkeiten und Schwierigkeitsgrade bietet. Hier wurde ein gewöhnliches Büro gewählt, welches diese Bedingungen erfüllt. Ein Büro bietet einerseits die Möglichkeit für die ersten Entwicklungsschritte unter Labor ähnlichen, eingeschränkten Bedingungen zu arbeiten, wie z.B. unter kontrolliertem künstlichen Licht bei heruntergelassenen Vorhängen auf einem aufgeräumten Schreibtisch und mit statischer Kamera. Andererseits werden bei einfallendem Tageslicht und in einem „normalen" unaufgeräumten Büro ohne Einschränkung des Blickfeldes mit einer mobilen Kamera sehr große Anforderungen an ein solches System gestellt.

Nicht nur die Umgebung, sondern auch die Vielfalt an Objekten in einem Büro birgt für die Entwicklung eines Objekterkenners eine große Spanne an Herausforderungen, die es zu bewältigen gilt. Dabei bleibt die Entwicklungsumgebung nachvollziehbar bzw. rekonstruierbar, da es eine Vielzahl an mehr oder weniger typischen Objekten gibt. Diese Objekte unterscheiden sich stark im Hinblick auf verschiedene Parameter, wie Form, Größe, Umriss etc.

In gewöhnlichen Büros geht die Spanne von kleinen kompakten Objekten, wie z.B. einem Anspitzer, einer Tasse oder einem Tacker, bis hin zu einem Schreibtisch oder einem Regal. Letztere sind allein durch ihre Größe immer nur teilweise sichtbar oder teilweise verdeckt und liefern dadurch eine andere Anforderung an ein solches System. Ebenso bieten die Umrisse von Objekten sehr unterschiedliche Schwierigkeitsgrade. Auch hier reicht die Spanne von kleinen kompakten Objekten zu sehr kompliziert geformten Objekten, die dazu auch noch eine große Variabilität im Hinblick auf den Blickwinkeln liefern, wie z.B. größere Zimmerpflanzen.

Aufgrund dieser unterschiedlichen Anforderungen wurde das System in zwei Schritten entwickelt. Die ersten Schritte für die Entwicklung des Subsystem zur Aufmerksamkeits-

steuerung und für die Entwicklung und Evaluation des Subsystems zur Zeigegestenerkennung, wie in Kap. 5 bzw. Kap. 7.2 beschrieben, wurden in einem statischen Szenario durchgeführt. Die fortgeschrittene Entwicklung und die mobilen Fähigkeiten wurden in der gleichen Umgebung (vgl. 4.1.1), allerdings mit einer mobilen AR-Ausrüstung durchgeführt (vgl. 4.1.2).

4.1.1 Szenario mit statischer Kamera

Die ersten Schritte der Entwicklung fanden in einem Büroszenario statt, in dem der Benutzer vor seinem Schreibtisch sitzt und mit typischen Büroobjekten hantiert. Wie in Abb. 4.1 gezeigt liegen auf dem Schreibtisch vor dem Benutzer Objekte, wie ein Anspitzer, ein Tacker, ein Telefon etc. Im Regal hinter dem Benutzer ist eine statische Kamera so montiert, dass sie die Region vor dem Benutzer in etwa aus seiner Blickrichtung aufzeichnet.

Sowohl in dem statischen als auch in dem mobilen Scenario läuft das System unter dem Betriebssystem *Linux*, welches eine stabile Unterstützung von IEEE 1394 Treibern für den Betrieb von Firewire-Kameras bietet.

Im statischen Szenario wurde die digitale Firewire-Kamera *DFW VL 500* von *SONY* verwendet, die dem IEEE1394 Standard entspricht. Sie liefert eine Wiedergaberate von 30 Bildern pro Sekunde mit einer VGA-Auflösung (640x480) in nicht komprimiertem YUV (4:2:2). Die Kamera zeichnet sich durch ein 12-fach-Zoomobjektiv mit einstellbaren Parametern, wie Zoom, Iris und Fokus, aus. Die benutze Treibersoftware basiert auf den Standardbibliotheken *libdc1394* und *libraw1394*. Die Bibliothek *libdc1394* wurde entwickelt, um eine hochwertige Programmierschnittstelle für die Kontrolle von IEEE 1394 Kameras, welche die 1394 basierte Digitale Kamera Spezifikation erfüllen, anzubieten. Die Bibliothek *libraw1394* liefert direkten Zugriff auf den IEEE 1394 Bus. Die verwendeten Linux-Treiber sind *ieee1394*, *raw1394* und *ohci1394*.

4.1.2 Mobiles Szenario

Im Zentrum der Arbeit stand die Entwicklung des mobilen Systems, welche ebenfalls im Büro stattfand. Im Unterschied zum statischen Szenario ist der Benutzer nicht darauf eingeschränkt, sich in dem von der statischen Kamera beobachteten Bereich zu bewegen, sondern kann sich frei im Büro bewegen. In diesem Szenario trägt der Benutzer eine AR-Apparatur wie die in Abb. 4.2. Die Basis dieser Apparatur besteht aus einem handelsüblichen Bergsteigerhelm. Dieser kann verhältnismäßig bequem getragen werden und bietet die Möglichkeit zur Montage der benötigten elektrischen Geräte. Auf dem Helm sind zwei „künstliche Augen" in Form von zwei kleinen Kameras angebracht, die die Umgebung annähernd in Blickrichtung des Benutzers aufnehmen. Die aufgenommene Bildsequenz der Kameras, bei Bedarf überlagert mit Systeminformationen, wird direkt in ein Display eingeblendet, welches sich unterhalb der Kameras und somit direkt vor den Augen des Benutzers befindet. Daneben kommt ein Funkmikrofon als ein Weg zur Kommunikation mit dem System zum Einsatz.

Abbildung 4.1: Links: Der Benutzer sitzt vor seinem Schreibtisch und zeigt auf ein Objekt vor ihm (hier ein Anspitzer). Rechts: Die am Regal montierte Kamera hinter und oberhalb des Benutzers im statischen Szenario.

Das Stereo-Kamerapaar besteht aus *FIRE-I* CCD Webcams der Firma UNIBRAIN. Für die mobile Anwendung sind sie aufgrund der leichten Bauweise gut geeignet. Zudem liefern sie ein gutes Bild in einer VGA-Auflösung mit einer Bildwiederholrate von bis zu 30 Bildern pro Sekunde (YUV 4-1-1). Die Treibersoftware basiert ebenfalls auf den Bibliotheken *libdc1394* und *libraw1394*.

Für die Darstellung des Bildes wird das 3-D I-Visor 4400VPD Display benutzt. Die Auflösung des wiedergegebenen Bildstroms ist SVGA, also mit einer Auflösung von 800 mal 600 Pixeln in Stereo mit 60, 70 und 75 Hz im VESA-Standard-Modus. Das Blickfeld für beide Augen hat jeweils eine Ausdehnung von 31 Grad.

Für die Spracheingabe wurden Komponenten der kabellosen Serie *ew 100 G2* von *SENNHEISER* verwendet. Zum einen sind das das mobile, am Kopf befestigte Mikrofon *ME3*, verbunden mit dem tragbaren Receiver *EK 100 G2*, und zum anderen der statische Empfänger *EM100G2*.

Das mobile System wurde in zwei Varianten entwickelt: einerseits als selbstständiges Demonstrationssystem und andererseits als Komponente eines mit kabellosem Netzwerk verbundenen Gesamtsystems. Im selbständigen System werden alle Berechnungen, die Visualisierung, die Verarbeitung des Sprachinputs etc. auf einem 1.8 GHz Laptop-Computer der Firma DELL ausgeführt, welcher auf einem eigens dafür modifizierten Rucksack getragen werden kann. Durch diese Einschränkung der verfügbaren Rechnerkapazität und der Voraussetzung der Echtzeitfähigkeit sind einige Algorithmen für die Demonstrationsversion angepasst worden. Z.B. wurde das Aufmerksamkeitsmodul, wie in Kapitel 5 beschrieben, weitestgehend minimiert, soweit das die Erhaltung der wesentlichen Charakteristik zuließ.

Abbildung 4.2: Benutzer mit dem mobilen System. Das Kamerabild der zwei kleinen am Helm befestigten Kameras wird in das Display eingeblendet und mit Systeminformationen ergänzt.

Als Komponente des Gesamtsystems wurde ein Datenaustausch zwischen dem mobilen System und einem Rechnernetzwerk mittels kabelloser Netzwerkverbindung gewährleistet. In diesem Falle kamen die optimierten Algorithmen zum Einsatz (die genaue Beschreibung folgt in den einzelnen Kapiteln).

4.2 Die Software-Entwicklungsumgebung NESSY

Als Programmierumgebung dient das von Gunther Heidemann entwickelte NESSY[1]. NESSY steht für „NEural viSion SYstem". Diese Softwareumgebung bietet dem Entwickler eine große Anzahl an Computervision und Neuronale Netze Bibliotheken in der Programmiersprache C, eine hoch entwickelte Programmierumgebung, um vielschichtige Prozessarchitekturen zu konstruieren und eine mächtige grafische Benutzeroberfläche mit der Möglichkeit laufende Prozesse zu kontrollieren und ggf. Fehler zu beseitigen. Die Bibliotheken bieten eine Vielzahl an implementierten Funktionen und Algorithmen. Diese reichen von einfachen Vektor- und Matrixoperationen über Bildfilter, Schnittstellen für den Zugriff auf Kameras bis hin zu Implementierungen von umfangreichen Algorithmen,

[1] http://www.techfak.uni-bielefeld.de/ags/ni/projects/compvis/Nixdorf/hp/NEssyEn.html

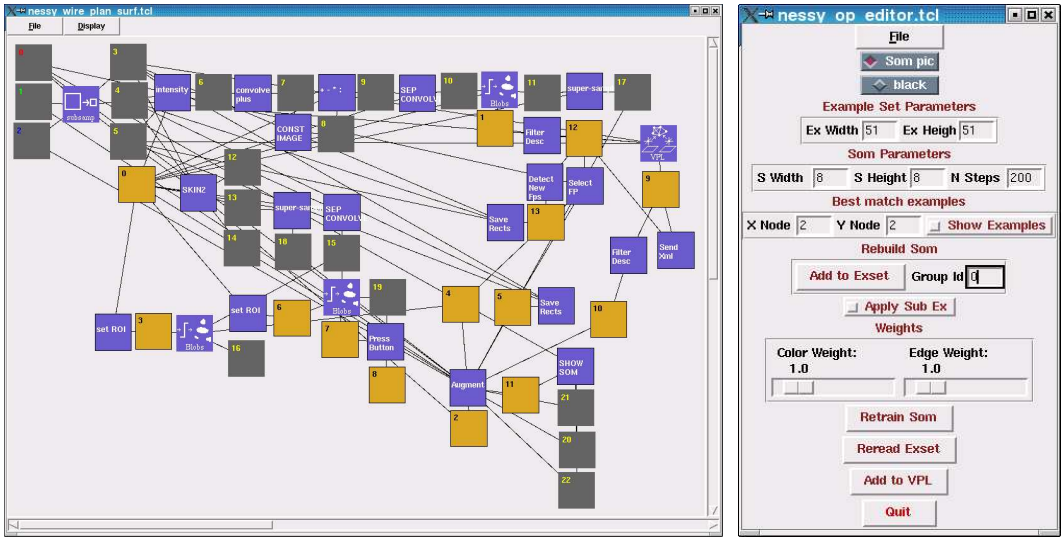


Abbildung 4.3: Grafische Oberfläche der Programmierumgebung NESSY. Links: Im Schaltplanfenster stehen einzelne Operatoren über Drähte als Symbol für stattfindenden Datenaustausch in Verbindung. Rechts: Im laufenden System können die Parameter jedes dieser Operatoren über eine grafische Benutzerschnittstelle verändert werden.

wie dem in Kapitel 6 beschriebenen VPL-Klassifikator.

Abb. 4.3 zeigt ein Beispiel für die grafische Oberfläche mit der Visualisierung des sogenannten NESSY-Schaltplanes", welcher den Verarbeitungsablauf der Bild- und Merkmalsverarbeitung widerspiegelt. Es gibt in dem typischen NESSY-Schaltplan drei Arten von grafischen Symbolen, die jeweils mit Drähten als Symbol für stattfindenden Datenaustausch verbunden sind. Die blauen Symbole stehen für einzelne Operatoren mit dahinter liegender unterschiedlicher Komplexität, die von einfachen Bildfiltern bis zum Klassifikator reichen. Die grauen Symbole geben die einzelnen Bilder von den Eingabebildern einer oder mehrerer Kameras über die Zwischenergebnisse der Operatoren bis zu den Ergebnisbildern wieder. Neben der Möglichkeit, Ergebnisse von Operatoren in Form von Bildern zu übergeben, symbolisieren die orangen Icons die Übergabe von Parametern anderer Datenformate. Klickt man auf die Operatorensymbole, erscheint ein Fenster (siehe Abb. 4.3, rechts), mit dem zur Laufzeit des Programmes Parameter geändert werden können und so die Operatoren an die gegebenen Anforderungen anpasst werden. Wählt man die Bildsymbole, kann man sich die einzelnen Bilder, z.B. das Ergebnis einer Faltungsoperation, in beliebiger Größe darstellen lassen. Zusätzlich können in diesen Fenstern errechnete Ergebnisse, wie z.B. Fokuspunkte, eingeblendet werden.

Durch diesen modularen Aufbau ist NESSY insbesondere für die Entwicklung komplexer Systeme sehr gut geeignet. Die Funktionalität ermöglicht es, in einem komplexen System die Verarbeitungsergebnisse einzelner Module zur Laufzeit zu sehen und somit sowohl die Funktionsweise als auch die Parameter je nach Notwendigkeit für das Gesamtsystem zu variieren. Die im weiteren Verlauf vorgestellten Module des Systems basieren

dabei zum Teil aus einer Kombination mehrerer Operatoren und zum Teil aus einzelnen Operatoren. Diese Operatoren stammen entweder aus den umfangreichen Bibliotheken oder sie wurden speziell für dieses System entwickelt.

Ein weiterer Vorteil von NESSY ist das komfortable Zusammenstellen des Schaltplans durch eine einfach aufgebaute Konfigurationsdatei. Dadurch kann der Verarbeitungsablauf in kürzester Zeit verändert oder es können einzelne Module ausgetauscht werden. Aufgrund dieser Vielseitigkeit wurde NESSY gewählt, da es sich bereits in vorherigen Projekten als stabiles, variables und komfortables Werkzeug in verschiedensten Bildverarbeitungs- und Neuronale-Netze-Anwendungen gezeigt hat und somit als Basis für die Entwicklung des Online-Lernsystems sehr gut geeignet ist.

Teil II

Basiskomponenten

Nach der Darlegung der Anforderungen an eine „Brille mit Gedächtnis" in Teil I der Arbeit, werden in diesem Teil die Basiskomponenten des Gesamtsystems vorgestellt, damit in dem anschließenden Teil das Zusammenwirken dieser Komponenten für die komplexen Mensch-Maschine-Interaktionen des lernfähigen Systems beschrieben werden kann.

Das in dieser Arbeit beschriebene mobile Augmented-Reality-System wurde entwickelt, um zur Laufzeit visuelles Objektwissen zu erlernen und dieses einem Benutzer verfügbar zu machen. Für die komfortable Verwendung eines trainierbaren, mobilen Systems ist die Basis eine robust funktionierende natürliche Mensch-Maschine-Kommunikation. Diese wird genutzt, um durch die intelligente Verknüpfung des Expertenwissens des Benutzers die Leistungsfähigkeit der Maschine nach dem bereits vorgestellten Prinzip des „human-in-the-loop" zu steigern. Das menschliche Expertenwissen wird dabei zum einen bei der Steuerung der visuellen Aufmerksamkeit und zum anderen bei den Lernprozessen verwendet. Da weder eine Tastatur noch eine Maus oder andere zusätzliche Eingabegeräte verwendet werden sollen, ist für die Mensch-Maschine-Kommunikation die Verwendung von Sprache und Gestik obligatorisch. Dabei sind drei Kategorien von Aufgaben zu bewältigen:

- *Die Kommunikation mit dem System über die Umgebung, d.h. vor allem das Lenken der Aufmerksamkeit auf Objekte und die Anfrage nach bereits ins Gedächtnis eingeprägten Objekten.*
- *Die Kontrolle des Systems, wie z.B. das Wechseln zwischen den einzelnen Systemfunktionen und den dazugehörigen Anzeigemodi.*
- *Die Readaptation der Mensch-Maschine-Schnittstelle für den Fall, dass durch externe Faktoren, wie veränderte Beleuchtungsbedingungen oder starke Umgebungsgeräusche, die Kommunikation gestört wird.*

In diesem Teil werden die Basiskomponenten für die Mensch-Maschine-Interaktion beschrieben und in Teilen bereits evaluiert. Das robuste Funktionieren dieser Basiskomponenten ist die Voraussetzung für die später beschriebenen komplexen Mensch-Maschine-Interaktionen, die Hauptaufgaben des Systems, also das Iterative Labeln von Bilddaten, das Online-Lernen von Objekten und deren Abfrage. Die Basiskomponenten des Systems sind die Module für die visuelle Aufmerksamkeit, welche die Referenzierung von Objekten mit Hilfe von Zeigegestenerkennung integriert, und die durch Sprache und Gestik gesteuerte Menükontrolle. Durch die Anforderung an das System, in Echtzeit zu arbeiten, müssen Wege gefunden werden, um mit der hochdimensionalen Mannigfaltigkeit im Pixelraum durch das vielgestaltige Bild von Objekten unter unterschiedlichsten Bedingungen und Ansichten umzugehen. Bei der Beschreibung der einzelnen Module werden daher an verschiedenen Stellen Verfahren vorgestellt, mit denen man die Komplexität des Problems erheblich reduzieren kann. Im darauf folgenden Teil wird dann das Zusammenwirken dieser Komponenten für die komplexen Aufgaben des Systems beschrieben. Die Abb. 4.4 zeigt die verschiedenen Mensch-Maschine-Kommunikationskanäle des entwickelten Systems, welche sich aus den Systemanforderungen ergeben.

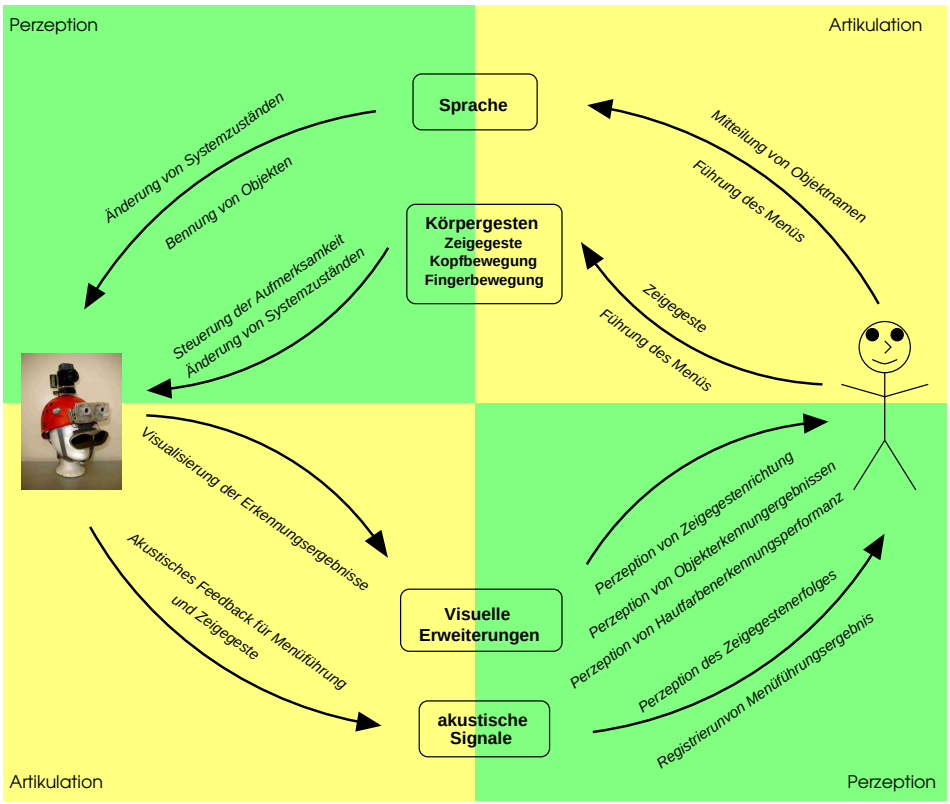


Abbildung 4.4: Überblick über die verschiedenen Mensch-Maschine-Kommunikationskanäle.

Kapitel 5

Benutzer gesteuerte visuelle Aufmerksamkeit

Grundlage für die Funktionalität der anvisierten Brille mit Gedächtnis ist das Speichern von Wissen über die Aufenthaltsorte von Objekten in der Umgebung des Benutzers. Die Erstellung eines visuellen Objektgedächtnisses teilt sich in den meisten aktuellen künstlichen Objekterkennungsarchitekturen in die beiden Komponenten der Objektlokalisation und der Objektklassifikation. Objektlokalisation, also das Auffinden von Objekten im Bild, wird dabei von einem artifiziellen Aufmerksamkeitssystem bewerkstelligt, um geeignete Orte im Bild zu finden, welche dann von dem Erkennungssystem auf das Vorkommen von bekannten und unbekannten Objekten untersucht wird.

Wie bereits in Kaptitel 3 beschrieben gibt es neuere Ansätze (wie z.B. Bax (2007)), welche versuchen, beide Aufgaben mit dem gleichen künstlichen neuronalen System zu bewerkstelligen. In dieser Arbeit wird jedoch aufgrund von verschiedenen Funktionen, die benötigt werden, insbesondere für die Integration des menschlichen Expertenwissens, diese Funktionalität aufgeteilt in ein künstliches Aufmerksamkeitssystem und ein auf neuronalen Netzen basierenden Objekterkenner. Im Gegensatz zu integrierten Systemen dient das Aufmerksamkeitssystem nicht nur als Lieferant für mögliche Orte von Objekten im Bild, sondern ermöglicht zum einen die Steuerung der Aufmerksamkeit durch den Benutzer und zum anderen die Erkennung von Bewegungen des Benutzers, welche sowohl für die Interaktion mit dem System als auch für die semiautomatische Erstellung einer Bilddatenbank (siehe Kaptitel 10) benötigt wird.

5.1 Künstliche visuelle Aufmerksamkeit

Die künstliche visuelle Aufmerksamkeit ist ein komplexes Gebiet in der Informatik und wird auf unterschiedliche Art und Weise in künstlichen Systemen realisiert. Die Art der Realisierung einer künstlichen Aufmerksamkeit ist in den meisten Systemen biologisch motiviert und basiert auf dem Wissen von der visuellen Aufmerksamkeit des Menschen, bzw. von Primaten. Diese kann man mit einem Scheinwerfer vergleichen, der von einem Ort zum anderen wechselt, dabei Regionen überspringt, um dann eine Zeit auf einer interessanten Region zu verweilen.

Diese Aufmerksamkeit ist nicht gleichzusetzen mit der Augenbewegung. Die Verschiebung der visuellen Aufmerksamkeit ist etwa viermal schneller. Jedoch ist nach der

Independence-Hypothese, welche von der Mehrheit der Aufmerksamkeitsforscher vertreten wird, die Verschiebung der visuellen Aufmerksamkeit zwar ein eigenständiger Prozess, sie ist aber nicht völlig unabhängig von der Augenbewegung (Sheperd, 1986).

Für die Perzeption von Objekten spielt die visuelle Aufmerksamkeit eine entscheidende Rolle. Nach dem heutigen Stand der Forschung besteht die Perzeption von Objekten beim Menschen aus drei hintereinander ablaufenden Schritten:

- Ein Bild von einem Objekt und seiner Umgebung wird auf das visuelle System projiziert. Atomare Merkmale werden schon hier detektiert (Graham, 1980).
- Ein Objekt wird aus der Umgebung heraus isoliert, indem nicht relevante Merkmale verworfen werden.
- Die übrig gebliebenen relevanten Merkmale werden in einer für den Beobachter zugänglichen Form integriert (Treisman, 1980).

Diese drei Verarbeitungsstufen sind notwendig und laufen in dieser Reihenfolge ab. Bei allen drei Stufen spielt die Aufmerksamkeit eine Rolle. Die bekannteste und anerkannteste Theorie über die visuelle Aufmerksamkeit ist die „Feature Integration Theorie" (FIT) von Treisman (Treisman, 1980, 1988, 1982, 1992). Nach der zentralen Prämisse dieser Theorie ist für die korrekte Wahrnehmung von Objekten die fokale Aufmerksamkeit essentiell (Treisman, 1980). Die FIT wurde in neueren Untersuchungen immer wieder diskutiert (wie z.B. in Duncan (1992)) und leicht geändert (Treisman, 1992), die grundsätzlichen Eigenschaften der Theorie sind aber weitestgehend unbestritten. Die Theorie basiert auf der Unterteilung der visuellen Aufmerksamkeit in die endogene und die exogene Aufmerksamkeit. Die exogene Aufmerksamkeit wird nur durch die Stimuli im visuellen Feld hervorgehoben, ist also rein datengetrieben und wird daher auch als präattentive oder „Bottom-up"-Aufmerksamkeit bezeichnet. Ihr steht die endogene Aufmerksamkeit gegenüber, die willentlich getrieben ist, und auch als „Top-down"-Aufmerksamkeit bezeichnet wird. Die Trennung der beiden Arten von Aufmerksamkeit gelang durch die sogenannten „Precuing" Experimente. Hierbei wird Aufmerksamkeit von Probanden auf ein Target gelenkt, nachdem man bei ihnen eine gewisse Erwartung für bestimmte Bereiche geweckt hat (Schneider, 1977).

Die Grundlage der FIT ist, dass für die visuelle Aufmerksamkeit präattentiv einzelne Merkmale parallel vom visuellen System ausgewertet und in Form von intradimensionalen und retinotopen Merkmalskarten (im engl. als *feature maps* bezeichnet) verwaltet werden. Darauf basierend folgt dann die endogen beeinflusste Komponente.

Durch diese Art der Verarbeitung findet die Theorie ebenfalls Einzug in viele künstliche visuelle Systeme, da Merkmale direkt aus dem Bild heraus und modular verarbeitet werden. Somit besteht die Möglichkeit, vorhandene Bildverarbeitungsmethoden zu verwenden und miteinander und mit neuen Methoden zu kombinieren.

Das in dieser Arbeit entwickelte System beruht ebenfalls auf der Feature Integration Theorie und basiert auf Arbeiten von Rae (2000) und Heidemann (1998a). Es ähnelt in der Verarbeitung der datengetriebenen Komponenten dem Aufmerksamkeitssystem von Backer u. a. (2001) für die Blickrichtungskontrolle eines Active Vision Systems. Ähnliche

Architekturen verwenden Walther u. a. (2002a) und Itti u. a. (1998), welche teilweise auf dem hierarchischen Modell von Walther u. a. (2002b) basieren.

Die hier vorgestellte Architektur ermöglicht es, sich einer der größten Herausforderungen in der Mensch-Maschine-Kommunikation zu stellen, nämlich dem Erzielen eines gemeinsamen Aufmerksamkeitsfokus von Mensch und Maschine.

Ein künstliches Aufmerksamkeitssystem, welches über die Integration verschiedener Merkmalskarten arbeitet, muss für die Objektlokalisation zuerst die Regionen im Bild finden, die sich als potentielle Kandidaten für den Aufenthaltsort von Objekten eignen, die sogenannten „regions of interest" (kurz ROI). Diese ROIs werden dabei über die Auswertung verschiedener Merkmale, wie Farbe, Ecken, Kanten, Symmetrien etc., bestimmt. In diesen Regionen müssen dann stabile Fokuspunkte (kurz FP) gefunden werden, welche als Zentren für die Erkennung von Objekten aufgrund von Bildausschnitten dienen. Diese FPs können einerseits Maxima von Merkmalskarten oder aber die Zentren von ROIs sein (für Details vergleiche (Heidemann, 1998a)).

5.2 Referenzierung von Objekten

Die rein datengetriebene Fokussierung der Aufmerksamkeit wird in der hier vorgestellten Architektur über die Erkennung von Zeigegesten und deren Integration gesteuert, um eine gemeinsame Aufmerksamkeit von Mensch und Maschine zu erreichen. Im Gegensatz zu den meisten anderen Systemen wird diese Aufgabe hier nicht über die spätere Verarbeitung auf einer symbolischen Ebene erreicht, sondern bereits auf einem subsymbolischen Level. Es wird im Folgenden ein Aufmerksamkeitssystem vorgestellt, welches neben dem Auffinden von interessanten Bildregionen durch die Auswertung von einfachen, bottom-up-verarbeiteten Bildmerkmalen auch die Steuerung der Aufmerksamkeit top-down durch Zeigegesten des Benutzers ermöglicht. Durch die Integration des Expertenwissens des Menschen, welcher die Aufmerksamkeit des Systems durch eine Geste steuern kann, auf einem subsymbolischen Level, werden verschiedene Nachteile beim maschinellen Auswerten von Zeigegesten umgangen. Wenn eine Person einer anderen etwas zeigt, wird die Geste nicht mit extrem hoher Präzision gemacht, sondern der Zeigende setzt ein gewisses Verständnis des Kommunikationspartners über die gemeinsam betrachtete Umgebung voraus. Des Weiteren sind deiktische Gesten ohne sprachliche Erläuterungen nicht präzise (McNeil, 1992; Wexelblat, 1998; Quek u. a., 2002). Wenn mehrere Objekte auf einem Tisch stehen und auf eines gezeigt wird, dann reicht es dem Gegenüber, wenn klar ist, in welche ungefähre Richtung gezeigt wird, wenn also klar für ihn erkennbar ist, dass die Zeigerichtung einem Objekt am nächsten ist. Es ist dadurch nicht notwendig, dass die Zeigerichtung in einer maximal zu erreichenden Genauigkeit erkannt wird, da sie einerseits nicht so genau ausgeführt wird und andererseits durch Nutzen von Kontextinformationen nicht so exakt interpretiert werden muss. Der Kontext besteht hier aus der Kenntnis der Orte von Objekten, die möglicher Weise gemeint sein könnten. Dadurch wird die Auswertung der Zeigegeste auf einige wenige diskrete Werte beschränkt. Diese Art des Umgangs mit menschlichen Zeigegesten wird hier durch die subsymbolische Integration der ausgewerteten Zeigegeste erreicht. Diese Integration

wird über die Repräsentation des visuellen Aufmerksamkeitsfokus in Form einer sogenannten kortikalen Karte erreicht. Die kortikale Karte ermöglicht eine Verarbeitung der visuellen Aufmerksamkeit, die der FIT von Treisman ähnelt, und hat folgende Vorteile:

- Durch die rein datengetriebene Bestimmung möglicher Aufenthaltsorte von Objekten erhält das System ein basales Verständnis von der Umgebung. Dies ermöglicht die Einschränkung der kontinuierlichen Zeigerichtungsmöglichkeiten auf einen diskreten Satz. Kontextfreie Aufmerksamkeitsmechanismen, wie die Detektion von Entropy, Symmetrie und von Kanten, liefern mögliche ROIs. Dadurch „antizipiert" das System mögliche Ziele der Zeigegesten.

- Die Verarbeitung der Aufmerksamkeit durch zweidimensionale Karten ermöglicht ebenfalls eine einfache Integration symbolischer Information. So können durch Sprachsteuerung mit Kommandos, wie „weiter links" oder „dahinter", Teilregionen in Bezug auf den aktuellen Aufmerksamkeitsfokus im Bild ausgeblendet werden.

- Die subsymbolische Integration der Zeigegeste ermöglicht eine bessere Mensch-Maschine-Kommunikation, da die subsymbolische Verarbeitung gut durch ein akustisches Feedback rückgemeldet werden kann und somit dem Benutzer hilft, das System zu steuern.

Im Folgenden wird zuerst ein Überblick über die Verarbeitungsarchitektur für die benutzergesteuerte visuelle Aufmerksamkeit gegeben. Anschließend werden die einzelnen Merkmalskarten und deren Integration in die kortikale Karte vorgestellt und die Anpassung der Merkmale an verschiedene Domänen beschrieben. Erst danach wird die Erkennung der Zeigegesten erläutert, welche gemeinsam mit weiteren möglichen endogenen benutzergesteuerten sogenannten Manipulatorkarten in die visuelle Aufmerksamkeitsberechnung einfließen kann.

5.3 Verarbeitungsarchitektur des Aufmerksamkeitssystems

Die Abbildung 5.1 zeigt die Architektur für die Verarbeitung der benutzergesteuerten visuellen Aufmerksamkeit. Auf Basis der Kamerabilder werden zuerst drei Merkmalskarten berechnet. Die Merkmalskarten können grundsätzlich variiert werden. Hier hat sich eine Kombination von Merkmalskarten unter der Verwendung der lokalen Entropy, der Symmetrie und der Grauwertkanten bewährt. Das ATM-Modul (engl. für ***At****tention* ***M****odul*) berechnet aus den basalen Merkmalskarten eine gewichtete Summe über eine adaptive Gewichtungsfunktion. Die Maxima dieser Berechnung korrespondieren mit den Bereichen, die möglicher Weise für das System interessant sind, und dienen als mögliche Ziele für die Referenzierung des Benutzers durch eine Zeigegeste.

Parallel dazu läuft die Zeigegestenerkennung. Diese beginnt mit der Hautfarbensegmentierung. Für die Entwicklung der Zeigegestenerkennung wurde einerseits die vorgestellte AR-Apparatur verwendet und andererseits eine kleine an einer handelsüblichen Brille befestigte Kamera, welche eine geringere Brennweite hat und somit einen größeren

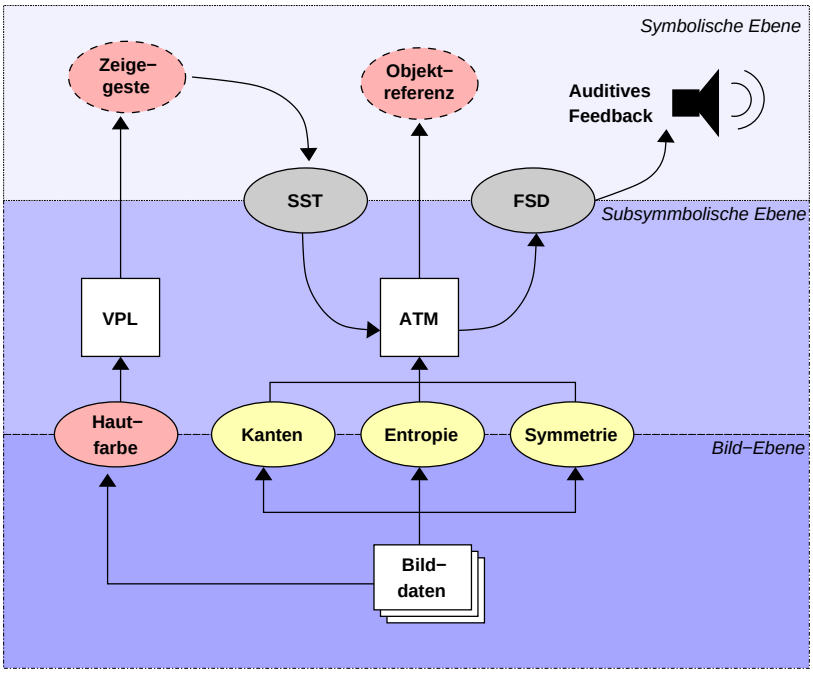


Abbildung 5.1: Prozessarchitektur für die Integration der Zeigerichtung in die Aufmerksamkeitssteuerung inklusive auditivem Feedback.

Bildausschnitt liefert. Je nach verwendeter Kamera kann somit einerseits die komplette Hand im Bild sein und andererseits nur der zeigende Finger. Für beide Fälle wurde eine Erkennung der Zeigegeste entwickelt. Der Klassifikator (als VPL bezeichnet, siehe Kap. 6) klassifiziert dann, ob es sich bei dem gefundenen Bildausschnitt um eine Zeigegeste handelt oder um ein anderes Objekt, bzw. eine nicht zeigende Hand handelt. Wenn eine zeigende Hand erkannt wurde, ist das Klassifikationsergebniss eine Schätzung der 2D-Zeigerichtung. Die gefundene Richtung wird anschließend durch den SST (Akronym für engl. ***S**ymbol **S**ignal **T**ransformer*) wieder auf die subsymbolische Ebene transformiert. Dabei wird eine sogenannte Manipulatorkarte erstellt, welche multiplikativ auf der gewichteten Summe der berechneten Merkmalskarten im ATM-Modul überlagert wird. Dadurch werden die Aufmerksamkeitsmaxima in Zeigerichtung hervorgehoben, während andere Maxima unterdrückt werden. Um dies zu erreichen, besteht die Manipulatorkarte aus einem Konus mit hoher Gewichtung in Zeigerichtung, die zu beiden Seiten gaussförmig abnimmt und im Zentrum der zeigenden Hand beginnt. Die so bestimmte Objektreferenz wird in der AR-Apparatur bei Bedarf hervorgehoben. Das Wechseln von einem bottom-up-verarbeiteten Aufmerksamkeitsmaximum des ATM-Moduls zum Nächsten bei variierender Zeigegeste wird im FSD-Modul (Akronym, engl. für ***F**ocus **S**hift **D**etection*) verarbeitet und akustisch wiedergegeben.

5.4 Merkmalskarten

Im Folgenden werden die im Prototypen verwendeten Merkmalskarten vorgestellt. Diese Karten wurden gewählt, da sie ein breites Spektrum an für den Menschen wichtigen Bildmerkmalen abdecken und somit je nach erforderlichen Aufgaben bestimmte Merkmale hervorgehoben werden können. Da die Verarbeitung auf einzelnen Karten basiert, ist eine Ergänzung bzw. ein Austausch leicht möglich.

5.4.1 Entropie

Lokale Entropie als Kriterium für die Interessantheit eines Bildes zu verwenden, basiert auf der Informationstheorie, die von Shannon eingeführt wurde (Shannon, 1948). Die zugrunde liegende Annahme ist, dass sich interessante Bereiche in einem Bild durch ein hohes Maß an Entropie auf der Pixelebene auszeichnen. Die hier benutzte Methode basiert auf den Arbeiten von Kalinke und von Seelen (1996). Dieser Algorithmus wurde schon erfolgreich in anderen Bildverarbeitungsarchitekturen angewendet (Kalinke und Handmann, 1997; Handmann u. a., 2000). Der Berechnung der Entropiekarte M_E liegen die Grauwerte, gewöhnlich in niedriger Auflösung, zugrunde. Die Berechnung folgt der Gleichung:

$$M_E(x,y) = -\sum_q P(x,y,q) \cdot \log P(x,y,q), \tag{5.1}$$

$$P(x,y,q) = \frac{\mathcal{N}(x,y,q)}{\sum_{q'} \mathcal{N}(x,y,q')}. \tag{5.2}$$

Hierbei bezeichnet $\mathcal{N}(x,y,q)$ das Histogramm der Grauwerte innerhalb eines $n_E \times n_E$-Fensters um das Pixel (x,y) (wobei gelten muss $n_E \geq 3$ und ungerade) mit q als Index für den Histogrammeintrag. Die Histogrammauflösung sollte so gewählt werden, dass die Anzahl der Gruppierungen N_B nicht zu groß gewählt wird, um zu gewährleisten, dass $(n_E)^2$ Werte der Pixel im Fenster noch zu einer guten Approximation der Grauwertverteilung führen können. Details zur Parametrisierung werden in Kapitel 5.5.1 vorgestellt.

Der entscheidende Parameter bei der Berechnung ist die Fenstergröße in Verbindung mit der Auflösung des Intensitätsbildes. Diese Relation bestimmt die Ausdehnung von Mustern, die dadurch hervorgehoben werden. Wenn ein sehr kleines Fenster gewählt wird, tendiert die Entropiekarte wie ein Kantendetektor dazu, nur kleine Strukturen, bzw. Grauwertänderungen mit geringer Ausdehnung hervorzuheben. Diese Aufgabe übernimmt in diesem System der Harrisdetektor (siehe Kap. 5.4.2). In den meisten Anwendung wird hier eine Fenstergröße von $n_E = 7$ und eine Quantisierung von $N_B = 4$ gewählt um die Aufmerksamkeit auch noch auf größere Objekte in dem Szenario zu lenken. Die Variabilität der Verwendung der Entropiekarte wird in Abbildung 5.2 wiedergegeben. Die Abbildung zeigt zwei unterschiedliche Parametrisierungen, welche dazu führen, dass in einem Fall große Objekte, wie eine Person oder eine Zimmerpflanze, hervorgehoben werden und im anderen Fall kleine Objekte auf einem Schreibtisch, wie ein

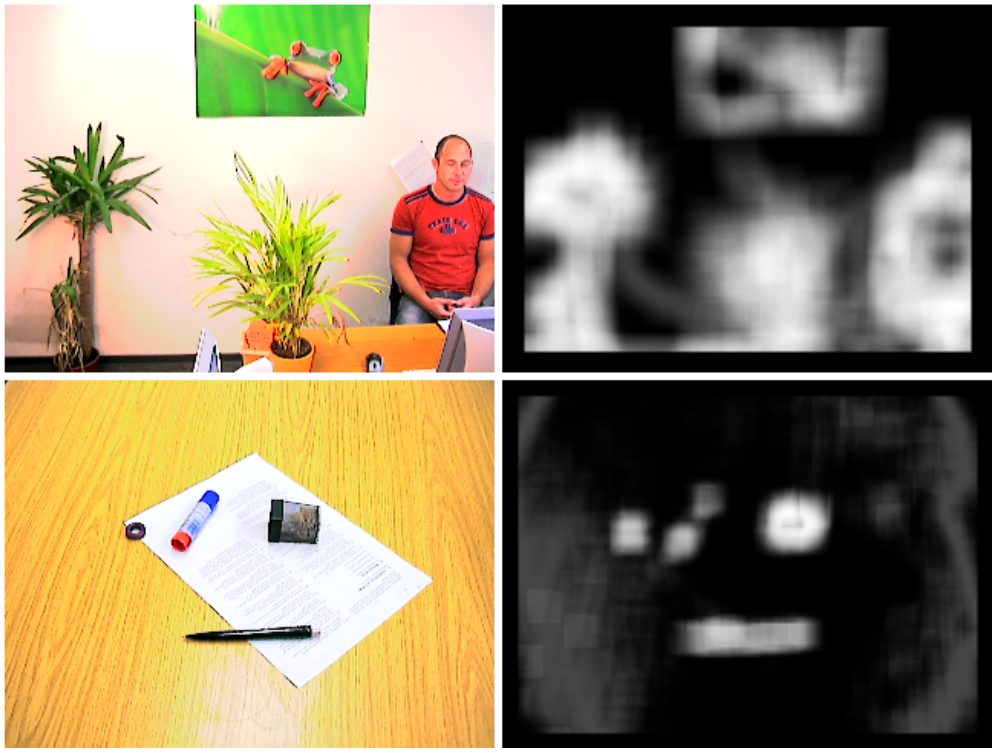


Abbildung 5.2: Entropiekarten für verschiedene Domänen. Oben: Segmentierung großer Objekte (Pflanze, Poster, Person) in einer Büroumgebung (Bildgröße 320×240, zweifach verkleinert, $n_E = 15$, $N_B = 3$). Unten: Die Parameter sind so gewählt, dass die Entropiewerte der Objekte groß und die der dahinter liegenden Textur (Holz, Skript) vernachlässigt werden. (Bildgröße 320×240, zweifach verkleinert, $n_E = 11$, $N_B = 2$).

Stift oder Anspitzer. Bei der späteren Berechnung der ROIs durch den Grassfirealgorithmus liefern diese Karten jeweils ganz unterschiedlich dimensionierte zusammenhängende Regionen (siehe Kap. 5.7).

Grundsätzlich kann Entropieberechnung auch auf Farbbilder erweitert werden. Dies führt allerdings zu einer sehr viel aufwändigeren Quantisierung. Da N_B, die Anzahl der Pixel, die zur Berechnung herangezogen werden, bei solch einer Anwendung klein ist, kann keine sinnvolle Quantisierung für den hochdimensionalen Pixel-Farbraum gefunden werden, sondern bedarf wiederum aufwändiger Anpassungen an die jeweilige Situation. Dies müsste je nach Applikation durch weitere Parameter erreicht werden, was neben der erheblichen Erhöhung des Rechenaufwands ein weiterer Grund war, weshalb hier die Entropie nur auf dem Intensitätsbild ausgewertet wird.

5.4.2 Harris

Die Merkmalskarte für die Hervorhebung von sehr kleinen Objekten und Objektdetails sowie für Kanten und Ecken resultiert aus dem von Harris und Stephens vorgeschlagenen Detektor (Harris und Stephens, 1988). Der Harrisdetektor wurde gewählt, da er in zahlreichen Applikationen erfolgreich verwendet wird und sich im Vergleich zu anderen Kantendetektoren als besser erwiesen hat (Schmid u. a., 2000). Er basiert auf der Schätzung der Autokorrelationsfunktion des Signals:

$$A(p) = \begin{pmatrix} \langle I_x^2 \rangle_{W(p)} & \langle I_x I_y \rangle_{W(p)} \\ \langle I_x I_y \rangle_{W(p)} & \langle I_y^2 \rangle_{W(p)} \end{pmatrix}, \tag{5.3}$$

mit $\langle \cdot \rangle_{W(p)}$ für die gewichtete Mittelung über ein Fenster $W(p)$ mit Pixel p als Zentrum. Als Gewichtsfunktion innerhalb des Fensters wird eine Gauß-Funktion verwendet. Der Wert des Detektors ist hoch, wenn beide Eigenwerte von A groß sind.

Zum Einsparen von Rechenzeit wird die Merkmalskarte wie folgt berechnet:

$$M_{Harris}(p) = \det(A) - \alpha \cdot (\text{spur}(A))^2 . \tag{5.4}$$

Die Varianz der gaußschen Gewichtsfunktion für die Komponenten von A innerhalb W ist $\sigma = 2$. Der Wert für die Konstante α beträgt 0.06, wie von Schmid u. a. (2000) vorgeschlagen.

5.4.3 Symmetrie

In einer künstlichen von Menschen erschaffenen Umgebung bietet sich die Symmetrie als weiteres Merkmal für die Lenkung der Aufmerksamkeit geradezu an. Auch dieses Merkmal ist biologisch motiviert. Bereits Bruce und Morgan (1954) und Locher und Nodine (1987) zeigten durch gezielte psychologische Experimente, dass die visuelle Aufmerksamkeit des Menschen stark durch symmetrische Strukturen gelenkt wird. In dem hier vorgestellten System wird die lokale Grauwert-Symmetrie, wie von Reisfeld et al. vorgestellt (Reisfeld u. a., 1995), verwendet. Während lokale Entropie für die Lokalisation größerer Objekte unabhängig von ihrer Struktur verwendet werden kann, führt die Symmetrie-Karte M_{Sym} zu einer stärkeren Fokussierung auf Objektdetails, welche lokale Symmetrien aufweisen. Dass der Algorithmus von Reisfeld u. a. (1995) die künstliche visuelle Aufmerksamkeit vergleichbar mit der menschlichen auf symmetrische Objekte lenkt, konnte bereits durch Eyetracking-Experimente von Privitera und Stark (2000) gezeigt werden. Die Symmetriekarte zeichnet sich weiterhin durch eine relative Unempfindlichkeit gegenüber unterschiedlichen Blickwinkeln, Beleuchtungsänderungen und leichten Bildstörungen aus, wie von Heidemann (2004b) gezeigt wurde. Für die Berechnung der Symmetriekarte wurde der Algorithmus in einer effizienteren Version als das Original von Reisfeld verwendet.

Das Ergebnis der Merkmalskarte $M_{Sym}(p)$ an einem bestimmten Pixel $p = (x, y)$ basiert auf den beiden Ableitungen $I_x(p), I_y(p)$, aus dem der Grauwertgradient $G_I(p) = \sqrt{I_x(p)^2 + I_y(p)^2}$ und seine Richtung $\theta_I(p) = \arctan\left(I_y(p)/I_x(p)\right)$ berechnet werden kann. Die Ableitungen I_x, I_y werden über 5×5 – Sobel-Filter bestimmt (vgl. Jähne

(1991)). Der Symmetriewert $M_{Sym}(p)$ ist die Summe über alle Pixelpaare (p_i, p_j), die in einer kreisförmigen Umgebung $\Gamma(p)$ um das Pixel p mit dem Radius R liegen:

$$M_{Sym}(p) \;=\; \sum_{(i,j)\in\Gamma(p)} \mathrm{PWF}(i,j)\cdot\mathrm{GWF}(i,j), \tag{5.5}$$

$$\Gamma(p) \;=\; \{(i,j) \mid (p_i+p_j)/2 = p \;\wedge\; \|p_i - p_j\| \le 2R\} \tag{5.6}$$

Hierbei steht PWF für die Phasengewichtungsfunktion (engl. für *Phase Weight Function*) und GWF für die Gradientengewichtungsfunktion (engl. für *Gradient Weight Function*). Die PWF ist ein Maß für den Likelihood-Schätzung, dass die Gradienten γ_i, γ_j bei p_i, p_j zu der Kontur eines symmetrischen Objektes gehören:

$$\mathrm{PWF}(i,j) \;=\; [1-\cos(\gamma_i+\gamma_j)]\cdot[1-\cos(\gamma_i-\gamma_j)]\,, \tag{5.7}$$

Dabei bezeichnen γ_i, γ_j die Winkel zwischen der Linie $\overline{p_i p_j}$, welche p_i und p_j verbindet und dem dazugehörigen Gradienten bei p_i respektive p_j. Die komplexe geometrische Bedeutung der PWF wird ausführlich in der originalen Arbeit beschrieben (Reisfeld u. a., 1995). Die GWF gewichtet die Werte für die Pixel (p_i, p_j) höher, wenn beide auf Kanten liegen, da Kanten auf Objektgrenzen hindeuten:

$$\mathrm{GWF}(i,j) \;=\; \log(1+G_I(p_i)) \;\cdot\; \log(1+G_I(p_j)). \tag{5.8}$$

Der Logarithmus führt zu einer Abschwächung des Einflusses von sehr starken Kanten. Die Abb. 5.3 zeigt ein Beispiel für die Symmetriekarte M_{Sym} mit unterschiedlicher Parametrisierung von R, bei dem die Symmetriemaxima einerseits auf den kleinen symmetrischen Knöpfen der Fernbedienung und andererseits im Zentrum der größeren Knöpfe auf der Tastatur liegen.

5.5 Domänenanpassung

Das Aufmerksamkeitssystem dient zur Selektion möglicher Orte von Objekten bzw. Objektteilen. Das Ausrichten der Aufmerksamkeit für die Selektion von Objekten oder die Referenzierung von Objekten durch Zeigegesten bei der Kommunikation zwischen zwei Personen wird durch Hintergrundwissen wesentlich gesteuert. Die semantische Bedeutung der Aufforderung „Gib mir mal den Anspitzer!" im Vergleich zu „Nimm dir den roten Stuhl!" führt durch die endogene Aufmerksamkeitskomponente in einem Fall dazu, dass das Gegenüber nach einem kleinen, vorzugsweise auf einem Schreibtisch stehenden Objekt sucht und in dem anderen Fall nach einem großen, roten auf dem Boden stehenden Objekt. Dieses Vorwissen führt zu einem wesentlich zielgerichteteren Suchen. Das Vorwissen über die Objektgröße kann in dem vorgestellten System vom Menschen durch Variation der Merkmalskartenparameter auf das Aufmerksamkeitsmodul übertragen werden. Auf den ersten Blick wirkt dies womöglich als wesentliche Einschränkung. Wie in dem oben geschilderten Beispiel führt es jedoch dazu, dass bei dem künstlichen System das Fehlen des semantischen Verständnisses einer Szene durch die Interaktion mit

Abbildung 5.3: Domänenanpassung der Symmetriekarten. Die Bilder haben eine Auflösung von 320×240. Ein Symmetrieradius von $R = 2$ hebt die Knöpfe der Fernbedienung hervor (oben), während die Symmetrie der größeren Objekte ignoriert wird. $R = 5$ führt zu Symmetriewertmaxima auf den einzelnen Tasten der Tastatur (unten).

dem menschlichen Experten kompensiert wird und somit die Kommunikation zwischen Mensch und Maschine zielgerichteter ablaufen kann. Diese Parameteradaptation wird durch die Verarbeitung der verschiedenen Merkmalskarten in dem Aufmerksamkeitsmodul ermöglicht. Der Benutzer kann durch einfache Mensch-Maschine-Interaktion (siehe Kap. 9) eine überschaubare Menge von voneinander unabhängigen Parametern variieren und somit die Aufmerksamkeit des Systems bei Bedarf an die gewünschte Objektdomaine anpassen. Der zweite Aspekt des oben genannten Beispiels, die Fokussierung auf ein Objekt in einer bestimmten Farbe, kann ebenfalls durch die Integration von geeigneten Merkmalskarten und deren manipulierbarer Gewichtung hervorgehoben werden.

5.5.1 Parametrisierung der Merkmalskarten

Alle Parameter der drei vorgestellten Merkmalskarten aus den Kapiteln 5.4.1–5.4.3 dienen zur Ausrichtung der Aufmerksamkeit auf unterschiedliche Skalierungen. Die Skalierungsparameter von M_E und M_{Sym} sind die Fenstergröße n_E und der Radius R. Die Abbildungen 5.2 und 5.3 zeigen jeweils zwei Beispiele, wie sich unterschiedlich gewählte

Parameter zur Anpassung an bestimmte Domänen auf die zu fokussierenden Objekte auswirken. Große und möglicherweise nicht symmetrische Objekte werden am besten durch die Entropiekarte hervorgehoben. Demgegenüber eignet sich die Symmetrie dazu, die Aufmerksamkeit auf kleine, symmetrische Objekte bzw. Objektteile zu lenken. Da beide Merkmalskarten nur einen zu variierenden Parameter haben, ist die Selektion einfach. Da die Entropiekarte vorzugsweise zur Detektion größerer Objekte dient, eignet sich zur Berechnung ein runterskaliertes Eingabebild. Dies hat den Vorteil, dass dadurch erheblich an Rechenzeit gespart werden kann. Es muss nur beachtet werden, dass ggf. für die Detektion notwendige Texturen durch die niedrigere Skalierung nicht verloren gehen, sondern sichtbar bleiben.

Grundsätzlich kann auch die Harriskarte M_{Harris} durch Variation der Standardabweichung σ der Gaußschen Gewichtungsfunktion an die Domäne angepasst werden. Da aber die lokale Berechnung besonders für die Detektion von kleinen Ecken und Kanten entworfen wurde, wird sie auch hier für ihren ursprünglichen Zweck verwendet. Dabei wird σ so klein wie möglich gewählt, so dass es eben noch keinen Effekt auf die Kantendetektion hat.

5.5.2 Integration anderer Merkmalskarten

Das System ermöglicht die Integration jeder möglichen Art von Merkmalsdetektoren für unterschiedlichste Applikationen, solange diese in Form einer Merkmalskarte $M^F(x, y)$ abgebildet werden können. Dabei ist eine Normalisierung nicht notwendig, da diese durch die Gleichungen des Algorithmus (Gl. 5.11, 5.12) der kortikalen Karte automatisch durch den Adaptationsprozess ausgeglichen wird. Es können sowohl völlig neue Merkmalsdetektoren integriert werden als auch weitere Instanzen der vorgestellten Merkmalskarten in unterschiedlichen Parametrisierungen. So könnten verschiedene Entropiekarten mit Eingabebildern in verschiedenen Auflösungen für die Hervorhebung unterschiedlich großer Objekte parallel verwendet werden.

Merkmalskarten, die auf Farbe basieren, können ebenfalls gut in das System integriert werden. Ein Beispiel dafür wäre die Verwendung einer von Heidemann (2004b) vorgestellten Farbsymmetriekarte, welche besonders bei mangelndem Schwarz-Weiß-Kontrast zum Einsatz kommen könnte. Da die Verarbeitung jedoch sehr rechenaufwändig ist, ist sie eher perspektivisch einzusetzen, wenn zukünftig neue Rechnergenerationen eine gesteigerter Rechnerperformanz bieten. Eine Möglichkeit, die Aufmerksamkeit auf auffällige Farbpartien zu lenken, wird bei Fislage u. a. (1999) vorgestellt. Diese Farbmerkmale wären besonders für Referenzierungen, wie im obigen Beispiel auf einen roten Stuhl, von großem Nutzen.

5.6 Manipulator-Karten

Die Manipulatorkarten dominieren das Verhalten der Merkmalskarten durch die multiplikative Überlagerung bei der Berechnung der Kortikalen Karte (siehe Kap. 5.7).

SST – *Symbol Signal Transformer*

Wie in der Abb. 5.4 der Prozessarchitektur des Aufmerksamkeitssystems gezeigt, liefert die Hautfarbensegmentierung und der VPL-Klassifikator die Position der Hand (x_H, y_H) und die Zeigegestenrichtung α (die ausführliche Beschreibung folgt in Kap. 7.2 bzw. Kap. 6). Diese beiden Informationen befinden sich auf der symbolischen Ebene und werden durch die Manipulatorkarte M_m^M zurück auf die sub-symbolische Ebene transformiert. Diese Karte besteht aus einem konusförmigen Strahl in Richtung der erkannten Zeigerichtung mit gaußschem Abfall zu beiden Seiten. Dieser hebt die Werte der Merkmalskarten M_i^F ähnlich wie der Lichtkegel eines Scheinwerfers im Zentrum stärker hervor, wird zu beiden Seiten schwächer und unterdrückt außerhalb die Werte:

$$M^M(x,y) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(\arctan(\frac{y-y_H}{x-x_H}) - \alpha)^2}{\sigma_c^2}\right), \tag{5.9}$$

hier zur Vereinfachung in der Form für den ersten Quadranten (vgl. Abb. 5.4). Der Konus gewichtet den Bereich in Zeigerichtung höher und hebt somit Aufmerksamkeitspunkte in dieser Region hervor.

Um die Selektion von Objekten unterschiedlicher Größe zu ermöglichen, kann die Standardabweichung des gaußschen Konus σ_c durch die Art der Benutzerbewegung gesteuert werden.

Die Zeigerichtung α und die Handposition (x_H, y_H) wird dazu über die letzten sechs Frames aufgezeichnet. Variiert sie stark, wird davon ausgegangen, dass der Benutzer auf ein größeres Objekt zeigen, bzw. eine größere Region hervorheben möchte. Um dies im Aufmerksamkeitsmodul zu berücksichtigen, wird σ des SST erhöht. Wenn der Benutzer längere Zeit präzise und still in eine Richtung zeigt, wird vorausgesetzt, dass der Benutzer sehr genau auf ein kleineres Objekt zeigen möchte. Dazu wird σ reduziert, so dass die Zeigerichtung als „virtueller Laserpointer" in die Berechnung einfließt. Beispiele für unterschiedliche Wahlen des Parameters σ zeigt die Abb. 5.4, rechts. Zusätzlich wird die Grob-, bzw. Feinselektion dadurch unterstützt, dass die apriori Gewichte ξ_i der Gleichung 5.12 für Referenzierung auf größere Regionen bzw. Objekte für die Entropiekarte M_E erhöht werden und im Gegensatz dazu die Gewichte für die Symmetriekarte M_{Sym} und für den Harrisdetektor M_{Harris} für die Fokussierung auf Details angehoben werden.

5.7 Kortikale Karte

Die adaptive Integration der Merkmalskarten $M_i^F(x,y)$, $i = 1, \ldots, N_F$ und der Manipulatorkarten $M_i^M(x,y)$,$i = 1, \ldots, N_M$ findet, wie in Abb. 5.4 dargestellt, im Aufmerksamkeitsmodul statt. Die grundlegende Idee zur Integration und Gewichtung geht zurück auf die Arbeiten von Rae (2000). Im vorgestellten System bestehen die vorher genannten Merkmalskarten aus der Entropiekarte, der Symmetriekarte und der Harriskarte. Somit ist die Anzahl der Merkmalskarten $N_F = 3$. Je nach aktuell zu bearbeitender Aufgabe werden unterschiedliche Manipulatorkarten, wie hier der SST, multiplikativ überlagert.

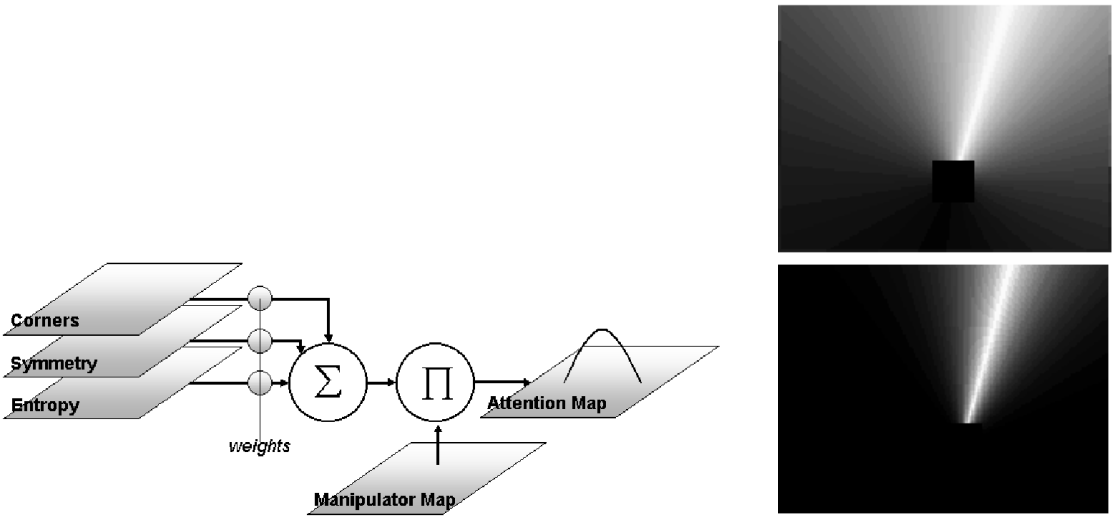


Abbildung 5.4: Links: Die Verarbeitungsreihenfolge des ATM-Moduls (der zentralen Komponente aus Abb. 5.1). Die kortikale Karte (*attention map*) C ist das Resultat aus der adaptiv gewichteten additiven Überlagerung der Merkmalskarten mit der multiplikativen Überlagerung der Manipulatorkarten. Rechts: Beispiele für die Manipulatorkarte des SST als „Lichtkegel der Aufmerksamkeit". Ein weiter Kegel wird verwendet, wenn der Benutzer auf große Objekte oder Regionen die Aufmerksamkeit lenken möchte, ein schmaler Kegel dient zur Referenzierung von Details.

Das Ergebnis des Aufmerksamkeitsmoduls wird hier kortikale Karte genannt. Deren Ergebnisbild $C(x, y)$ wird berechnet als eine gewichtete Summation der Eingabemerkmalskarten und das Produkt der verwendeten Manipulatorkarten:

$$C(x,y) \;=\; \sum_{i=1}^{N_F} w_i \cdot M_i^F(x,y) \;\cdot \prod_{j=1}^{N_M} M_j^M(x,y), \tag{5.10}$$

wobei negative Werte von $M_i^F(\cdot,\cdot)$ abgeschnitten werden. Das Maximum des Outputs der Kortikalen Karte $C(\cdot,\cdot)$ dirigiert das System auf den nächsten zu fokussierenden Punkt als möglichen Ort eines Objektes. Wird die Manipulatorkarte für die Erkennung der Zeigegeste verwendet, ist das Maximum der kortikalen Karte der mit dem Benutzer geteilte Fokus der Aufmerksamkeit in Richtung der erkannten Zeigegeste und somit das vermeintliche Ziel auf welches der Benutzer zeigt.

Um den Einfluss der verschiedenen Merkmalskarten in gleicher Weise zu berücksichtigen, wird die globale Kartenaktivität S_i als Summe über alle Pixel jeder einzelnen Karte M_i^F berechnet. Um eine annähernde Gleichgewichtung der S_i zu erreichen, werden die Kartengewichte w_i iterativ in Richtung der Zielgewichte w_i^s adaptiert:

$$w_i(t+1) = w_i(t) + \epsilon(w_i^s(t) - w_i(t)), \quad 0 < \epsilon \leq 1, \tag{5.11}$$

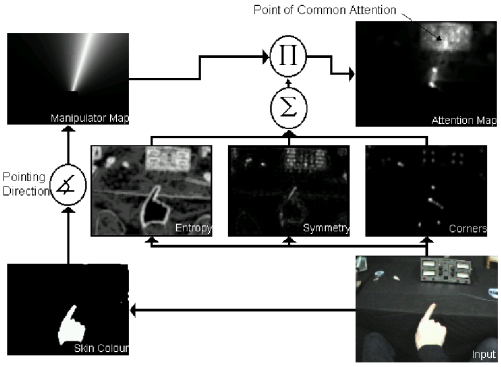


Abbildung 5.5: Darstellung der einzelnen Verarbeitungsschritte bei der Berechnung der gemeinsamen Aufmerksamkeit in der Laborumgebung. Der Benutzer referenziert per Zeigegeste auf einen Knopf eines Netzgerätes (*point of common attention*).

Als Zielgewichte werden die Gewichtes w_i^s gewählt:

$$w_i^s = \frac{1}{{N_F}^2} \cdot \frac{\sum_{k=1}^{N_F} S_k}{S_i} \quad \text{mit} \tag{5.12}$$

$$S_i = \frac{\sum_{(x,y)} (M_i^F(x,y) + \gamma)}{\xi_i}, \tag{5.13}$$

dabei erzwingt γ eine Grenze für die Gewichtszunahme. Die automatische Gewichtsanpassung wird detailliert in Fislage u. a. (1999) beschrieben. Die Parameter ξ_i können verwendet werden, wenn einzelne Merkmalskarten a priori höher gewichtet werden sollen, um z.B. für die Suche nach bestimmten Objekten durch eine höhere Gewichtung die Charakteristik einzelner Merkmalskarten hervorzuheben. Ein Beispiel hierfür wurde bereits im Kapitel 5.5.1 beschrieben, indem Entropy beim Zeigen auf große zusammenhängende Gegenstände höher gewichtet wird und im Gegensatz dazu beim Zeigen auf Objektdetails, wie einzelne Knöpfe eines technischen Gerätes, niedriger gewichtet wird.

Gewinnung der Fokuspunkte

Das Ergebnis der kortikalen Karte hängt von der jeweiligen Aufgabe und somit von den aktuell verwendeten Manipulatorkarten ab. Das vermeintliche Ziel einer Zeigegeste ist das Maximum der kortikalen Karte nach der Multiplikation mit der SST-Manipulatorkarte. Einen Überblick über die einzelnen Verarbeitungsschritte für die Erlangung des gemeinsamen Fokuses von Mensch und Maschine durch die Referenzierung eines Schalters auf einem Netzgerät per Zeigegeste zeigt Abb. 5.5.

Für andere Aufgaben bedarf es einer Lokalisation möglichst aller im Bild befindlichen Objekte. Da aufgrund der Verarbeitung der kortikalen Karte, mehrere Maxima auf ein und demselben Objekt in unmittelbarer Nähe liegen können, wird zur Objektlokalisation ein mehrschrittiges Verfahren angewendet. Zu Beginn wird die kortikale Karte mit einem

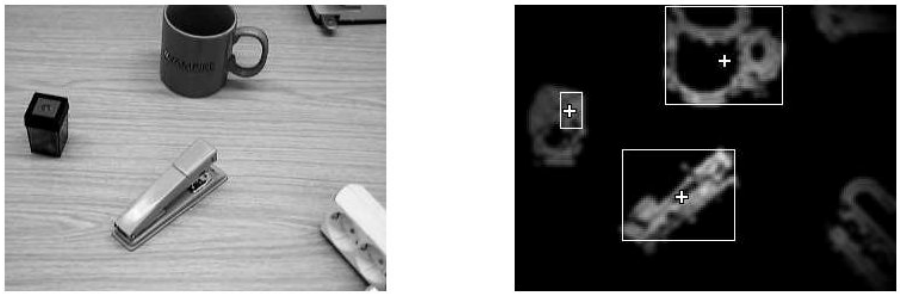


Abbildung 5.6: Links: Bildausschnitt aus dem Kamerabild. Rechts: Output des Aufmerksamkeitsmoduls. Der weiße Rahmen umgibt die berechneten ROIs. Das Kreuz markiert den Schwerpunkt der ermittelten Regionen.

5×5-Gaußkernel gefaltet. Die Varianz des Kernels wird so gewählt, dass Objektgrenzen, welche im Ergebnis der kortikalen Karte erkennbar sind, miteinander verschmelzen. Anschließend wird das Bild binarisiert. Auf diesem binarisierten Bild werden mit Hilfe des Grassfire-Algorithmus zusammenhängende Regionen ermittelt. Die Schwerpunkte dieser Regionen sind dann die Fokuspunkte des Systems. Dies hat den Vorteil, dass die FPs im Gegensatz zu einzelnen Maxima der kortikalen Karte bei leichten Beleuchtungsvariationen bzw. Bewegungen nicht hin- und herspringen, sondern verhältnismäßig stabil sind. Die x- und y-Ausdehnung dieser zusammenhängenden Regionen begrenzen dann die vom System ermittelten ROIs. Abbildung 5.6 zeigt das Ergebnis der Objektlokalisation mit den ermittelten ROIs und den dazugehörigen Fokuspunkten.

Kapitel 6

Das neuronale Klassifikationssystem

Für die Klassifikation von Objekten und von Zeigegesten wird in dem vorgestellten System ein auf neuronalen Netzen basierender Klassifikator verwendet. Wie bereits im Kap. 3 beschrieben, gibt es viele verschiedene Ansätze Objekte visuell zu klassifizieren. Für das vorliegende Vorhaben ist die Wahl auf den von Heidemann (1998b) entwickelten VPL-Klassifikator gefallen, weil dieser den Vorteil hat, dass er für die vielseitigen Funktionen des Systems auf unterschiedliche Art und Weise eingesetzt werden kann. Die im Folgenden beschriebene Trennung der einzelnen Schichten der Klassifikatorarchitektur erweist sich vor allem beim online-Lernen, wie in Kap.12 genauer beschrieben, als sehr hilfreich. Aufgrund seiner Robustheit gegenüber variierenden Bedingungen und einfachen und unempfindlichen Parametrisierung wurde er bereits in vielfältigen Anwendungen erfolgreich eingesetzt (z.B.Heidemann und Ritter (2003)).

VPL

Der VPL-Klassifikator kombiniert die Extraktion visueller Merkmale mit der Klassifikation. VPL steht für die drei Verarbeitungsschritte **V**ektor Quantisierung, Hauptkomponentenanalyse, (**P**CA, engl. für principal component analysis) und **L**LM (engl. für local linear maps). Abb. 6.1 zeigt die drei Verarbeitungsschritte. Die Eingaben für den Klassifikator sind die Bildausschnitte im Rohformat um die Fokuspunkte, welche das Aufmerksamkeitsmodul liefert. Das sind je nach Anwendung die Bildausschnitte der zeigenden Hand oder von vermeintlichen Objekten. Diese Bilddaten $x \in \mathbb{R}^D$ werden im ersten Schritt durch eine Vektorquantisierung auf N_V Referenzvektoren $\vec{r}_i \in \mathbb{R}^M, i = 1 \ldots N_V$ partitioniert. Für die Vektorquantisierung wurde der *Activity Equalization*-Algorithmus aus Heidemann und Ritter (2001) verwendet. Dieser Algorithmus zeichnet sich dadurch aus, dass die Verteilung der Daten auf die Referenzvektoren durch die Berücksichtigung der Aktivität vermeidet, dass einzelne Referenzvektoren unangetastet bleiben. Im Anschluss wird auf jede Partition der Daten eine lokale PCA berechnet. Die lokale PCA (Tipping und Bishop, 1999) mit dem vorherigen Vektorquantisierungsschritt kann als nicht-lineare Erweiterung der einfachen globalen PCA angesehen werden (Jolliffe, 1986).

Die Hauptkomponentenanalyse basiert auf dem von Sanger (1989) vorgestellten Verfahren. Zu jedem Referenzvektor $\vec{r}_i$ wird ein einschichtiges Feedforward-Netz für die sukzessive Berechnung der Hauptkomponenten (PCs) verwendet, welches die Eingabevektoren $\vec{x}$ auf die $N_P < D$ PCs mit den größten Eigenwerten projiziert:

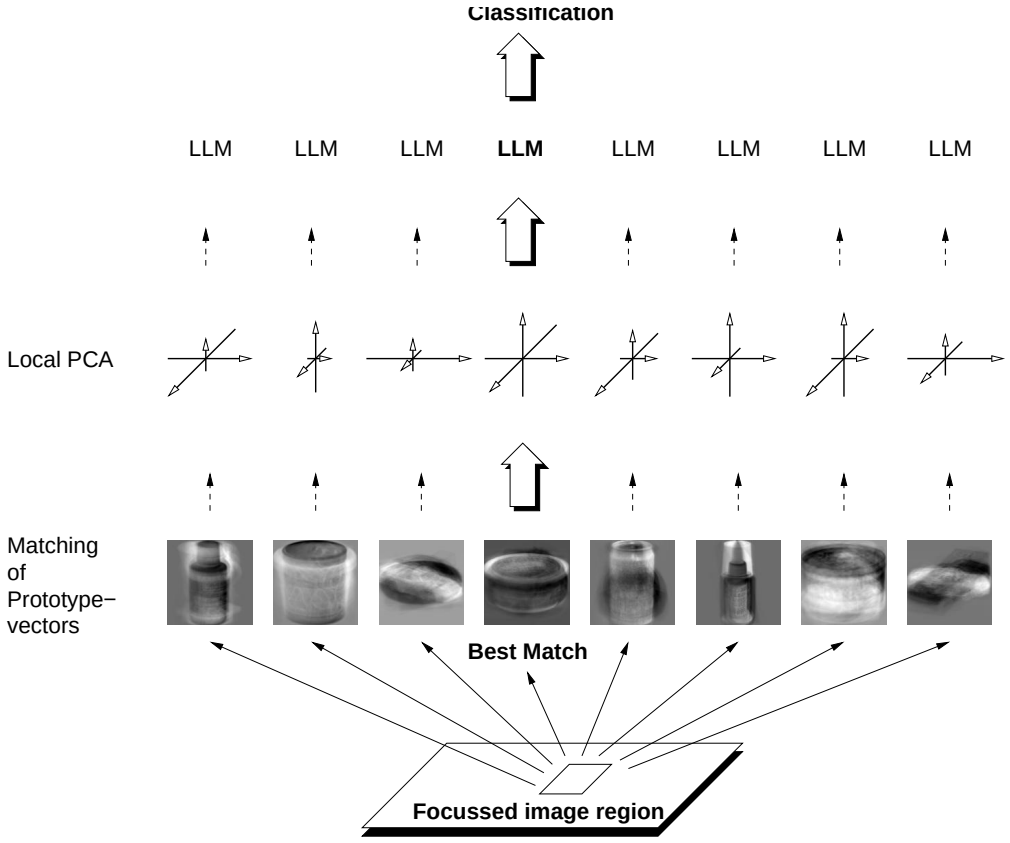


Abbildung 6.1: Architektur des Klassifikationssystems (nach Heidemann u. a. (2005)).

$\vec{x} \rightarrow \vec{p}_l(\vec{x}) \in \mathbb{R}^{N_P}, l = 1 \ldots N_V$. Diese erhebliche Dimensionsreduktion ermöglicht es, für jedes einzelne der N_V verschiedenen PCA-Netze einen eigenen Klassifikator als Experte für diese Teilmenge zu verwenden. Der Klassifikator ist vom LLM-Typ. Die Funktionsweise kann hier nur kurz beschrieben werden, für Details siehe z.B. Ritter u. a. (1992). Das LLM-Netz bildet dann als letzten Schritt mit $\vec{p}_l(\vec{x}) \rightarrow \vec{y} \in \mathbb{R}^N$ in den Ausgaberaum ab. Das LLM-Netz ist verwandt mit den Selbstorganisierten Karten von (Kohonen, 1984) und dem BRBF-Ansatz von Moody und Darken (1988). Das LLM-Netz ermöglicht eine Approximation nichtlinearer Funktionen durch einen Satz von lokal gültigen linearen Abbildungen. Dazu wird der Eingaberaum in unüberwacht trainierte Voronoizellen partitioniert. Und anschließend werden für die einzelnen Zellen überwacht erlernte lokal gültige Abbildungen in den Ausgaberaum vollzogen.

Die drei Verarbeitungsschritte werden sukzessiv trainiert, zuerst die beiden unüberwacht trainierten Schritte der Vektorquantisierung und der Ermittlung und Projektion auf die Hauptkomponenten und anschließend das überwachte Training des LLM-Netzes. Für die Klassifikation eines Eingabevektors $\vec{x}$ wird zuerst der best-match Referenzvektor $\vec{r}_{n(\vec{x})}$ bestimmt, dann wird $\vec{x}$ auf die zugehörigen $\vec{p}_{n(\vec{x})}(\vec{x})$ Hauptkomponenten mit dem jeweiligen PCA-Netz projiziert und schließlich werden die $\vec{p}_{n(\vec{x})}(\vec{x})$ auf die Aus-

gabe y durch das LLM mit $\vec{p}_{n(\vec{x})}(\vec{x}) \rightarrow \vec{y}$ abgebildet. Der Ausgabevektor $\rightarrow \vec{y}$ variiert dabei je nach Aufgabe. Details dazu sind bei den einzelnen Anwendungen beschrieben.

Für Objekterkennungsaufgaben liegt der Hauptvorteil des VPL-Klassifikators darin, dass man eine große Anzahl von hochgradig spezifischen Merkmalsdetektoren zur Verfügung hat, diese aber nur $N_V + N_P$ Filteroperationen pro Klassifikation benötigen. Der Klassifikator wurde bereits vielfach erfolgreich eingesetzt (z.B. in Heidemann und Ritter (2003)). In Heidemann (2004a) wurde gezeigt, dass die Merkmalsextraktion des VPL ebenfalls in der Domäne des *content based image retrieval* sehr effizient zur Repräsentation einer großen Menge von sehr unterschiedlichen Bilddaten eingesetzt werden kann.

In den bisherigen Anwendungen konnte ebenfalls gezeigt werden, dass sich die Generalisierungsfähigkeit und die Klassifikationsperformanz bei der Änderung der Hauptparameter, also N_V, N_P und die Anzahl der Knoten im LLM-Netz N_L, gutmütig verhält (Heidemann u. a., 2000).

Kapitel 7

Handgestenerkennung

Die Erkennung von Handgesten ist ein weites Feld in der Informatik. Dabei stellt sich die Frage, was genau als Geste bezeichnet wird. In der Mensch zu Mensch Kommunikation ist eine Geste eine Bewegung, die einen kommunikativen Aspekt für das Gegenüber trägt und nicht nur eine reine manipulierende Handlung, wie das Manipulieren eines Reglers an einem elektronischen Gerät (vgl. G. Kurtenbach (1990)). Nach Quek (1995) ist es in der Mensch-Maschine-Kommunikation durchaus sinnvoll, auch manipulierende Handbewegungen als Gesten aufzufassen. Die Verwendung von Handbewegungen in dieser Arbeit stützen diese These, da die Gesten sowohl einen kommunikativen als auch manipulierenden Charakter haben. Daher wird im Folgenden bei allen für die Mensch-Maschine-Kommunikation relevanten, ausgewerteten Handbewegungen von Gesten gesprochen.

Aufgrund der hohen Anzahl an Freiheitsgraden der Hand ist die Erkennung von Handgesten eine schwierige Aufgabe. Glücklicherweise wird bei den vom Menschen verwendeten Gesten nur ein geringer Anteil dieser theoretischen Mannigfaltigkeit verwendet. Für dieses System werden lediglich zwei Arten von Handgesten benötigt. Somit wird die Mannigfaltigkeit für die hier verwendete ansichtsbasierte Erkennung handhabbar und es wird keine weitere Hardware, wie Datenhandschuhe, benötigt. Zum einen werden Handgesten für die Steuerung der visuellen Aufmerksamkeit vom Benutzer verwendet, bei der Zeigegesten ausgewertet werden, die ein Objekt referenzieren und somit ein gemeinsames Verständnis der Umgebung von Mensch und Maschine gewährleisten. Andererseits muss der Benutzer die verschiedenen Funktionen des Systems auswählen können. Dazu wurde ein Menü entwickelt, dass im Kap. 9.2 detailliert vorgestellt wird. Die Menüführung soll intuitiv ohne die Verwendung von zusätzlicher Hardware, allein durch das Drücken von virtuellen, im Display eingeblendeten Knöpfen realisiert werden. Hierzu wird das Auftreten und die Bewegung einer Fingerspitze ausgewertet. Sowohl für die Zeigegestenerkennung, als auch für die Fingerspitzenerkennung ist eine Erkennung der Hautfarbe essentiell.

7.1 Hautfarbenerkennung

Für die Handgestenerkennung müssen die im Bild befindlichen Finger bzw. Hände vom System gefunden werden. Hautfarbenerkennung dient hier zur Selektion von geeigneten Bildregionen zur Erkennung von Fingern bzw. der Hand. Realisiert ist sie durch einen

zum Aufmerksamkeitsmodul parallel laufenden Verarbeitungsstrang. Da die Hände nicht als erlernbare Objekte dem System dienen und die Erkennung der Hände nicht von den vom System gefundenen interessanten Regionen abgelenkt werden sollen, sind die zwei Verarbeitungsstränge von einander getrennt.

Die Hautfarbenerkennung ist stark abhängig von den jeweiligen Beleuchtungsbedingungen. Um bei wechselnder Beleuchtung die Hautfarbe korrekt zu segmentieren, gibt es verschiedene, zum Teil sehr rechenaufwändige Verfahren, wie z.B. das von Lömker (2004) vorgestellte, bei dem zusätzlich zur Farbinformation die Bewegung ausgewertet wird. Durch eine Kombination von einer rein farbbasierten Regionenbewertung mit einer bewegungsbasierten Regionenbewertung über Wahrscheinlichkeitsmodelle und mit Hilfe von Kallmanfiltern werden so die Handtrajektorien verfolgt und dadurch die Parameter zur Hautfarbensegmentierung angepasst.

In dem vorgestellten, mobilen System wurde ein weniger rechenaufwändiges Verfahren verwendet, welches durch die einfache Interaktion mit dem Benutzer eine Adaptation an geänderte Beleuchtungsbedingungen zulässt. Dazu wurde ein Hautfarbenerkenner auf der Basis des Modells von M. Stoerring und Granum (2001) verwendet. Danach gibt es im $r-g$-Farbraum einen muschelförmigen Bereich der Hautfarbenverteilung, *skin locus* genannt. Der $r-g$-Farbraum ist durch $r = R/(R+G+B)$ und $g = G/(R+G+B)$ festgelegt. Dieser Farbraum lässt sich gut durch zwei Parabeln eingrenzen und somit identifizieren (Soriano u. a. (2000)). Beim Start des Systems werden die vordefinierten Standardparameter für die Parabeln verwendet. Wenn diese nicht mehr den aktuellen Bedingungen genügen und die Farbsegmentierung fehlerhaft wird, können sie durch den Benutzer, wie im Kap. 14 beschrieben, durch einfache Mensch-Maschine-Interaktion an die veränderte Beleuchtungssituation angepasst und der *skin locus* adaptiert werden.

7.2 Zeigegestenerkennung

Für die Realisierung einer Brille mit Gedächtnis wurden zwei Verfahren zur Zeigegestenerkennung verwendet. Abhängig von der Brennweite der Kamera unterscheiden sich die Anforderungen an die Auswertung der Zeigegesten erheblich. Da zu Beginn der Arbeiten die Art der später verwendeten Kamera noch nicht feststand und um eine höhere Flexibilität zu gewährleisten, wurde die Zeigegestenerkennung sowohl für Kameras mit kleiner als auch mit großer Brennweite entwickelt. In beiden Fällen sollte die Kamera aus der Perspektive des Benutzers die Szene aufzeichnen. Somit waren bei kleiner Brennweite ein größerer Teil der Szene und bei Zeigegesten die gesamte Hand im Bild zu sehen, wie in Abb. 5.5 gezeigt. Bei großer Brennweite war der Bildausschnitt entsprechend geringer, so dass für die Mensch-Maschine-Interaktion nur der im Bild befindliche Finger und nicht die gesamte Hand verwendet wurde, da es sonst zu großen Verdeckungen der Szene geführt hätte. Diese Handhabung des Systems wurde intuitiv von allen Benutzern je nach Kameratyp in dieser Art und Weise gemacht.

Für jegliche Erkennung der Handgesten ist die Hautfarbensegmentierung ein notwendiger Schritt, um die Finger bzw. die Hand zu lokalisieren. Durch die vorher vorgestellte Integration der erkannten Zeigerichtung in das Aufmerksamkeitsmodul auf einer

subsymbolischen Ebene ist jedoch zum einen die Anforderungen an die Präzision der Richtungsauswertung nicht sehr hoch, da aus dem kontinuierlich möglichen Zeigerichtungsbereich nur eine diskrete Anzahl an Zeigerichtungen wahrscheinlich ist, die durch die Objektlokalisation gefunden wurden. Zum anderen erfolgt auf den farbsegmentierten Regionen, genau wie bei der Objektlokalisation, die Analyse der ROIs durch die Erkennung der zusammenhängenden Regionen und der FPs als Schwerpunkt dieser Region. Diese FPs liefern das Zentrum der Bildausschnitte, welche später für die Klassifikation zur Verfügung stehen. Bei sich verändernden Beleuchtungsbedingungen fransen diese Regionen vor allem am Rand aus. Die Schwerpunkte der erkannten Regionen bleiben aber bis zu einem gewissen Grade stabil, so dass sich erst starke Veränderungen negativ auf die Gestenerkennung auswirken.

Durch die aufgrund der Brennweite der Kamera unterschiedlichen Anforderungen wurden zwei Verfahren zur Erkennung der Zeigegesten entwickelt:

- *i)* bei kleiner Brennweite wird die Zeigerichtung aufgrund des Bildausschnitts der gesamten Hand mit Hilfe des VPL klassifiziert,
- *ii)* bei großer Brennweite dient die ermittelte hautfarbene Region in dem entsprechenden Bildbereich zur Klassifikation der Geste und der Zeigerichtung.

7.2.1 Zeigegestenerkennung bei kleiner Kamerabrennweite

Im ersten Fall bei kleiner Brennweite der Kamera ergibt sich das Problem, für die ansichtsbasierte Erkennung eine ausreichende Trainingsmenge zu bekommen. Der überwacht lernende, ansichtsbasierte VPL-Klassifikator benötigt in der Trainingsphase eine große Menge an gelabelten Bilddaten. Die Bilddaten von der Hand sind über das Aufmerksamkeitsmodul in Kombination mit dem Hautfarbenerkennung leicht zu erhalten. Um jedoch zu erkennen, ob die Hand zeigt und vor allem in welche Richtung die Hand zeigt, müssen diese Bilddaten gelabelt werden. Für diese Aufgabe wurde ein unüberwachtes Lernverfahren, die sogenannten Selbstorganisierende Karten (engl. **s**elf **o**rganizing **m**aps, Akronym SOM) für die Vorstrukturierung der Bilddaten und dem anschließenden Labeln der Daten verwendet. Dieses Verfahren diente als Vorstufe für das Labeln von Bilddaten im mobilen System und wird daher im Kap. 10 detailliert vorgestellt.

Der über die Hautfarbensegmentierung ermittelte Bildausschnitt mit der hautfarbenen Region im Zentrum dient dem Klassifikator als Eingabe $\vec{x}$. Der VPL bildet $\vec{x} \rightarrow \vec{y}, \vec{x} \in \mathbb{R}^D, \vec{y} \in \mathbb{R}^N$ ab. Dabei entspricht die Eingabedimension D der Anzahl der Pixel des gefundenen Ausschnitts. Der Vektor x wird auf den dreidimensionalen Ausgabevektor $\vec{y} \in \mathbb{R}^3$ projiziert: Die ersten zwei Ausgabedimensionen stehen für die Klasse, die dritte für die Zeigerichtung. Die Klasse wird in den ersten beiden Komponenten in der Form $(1, 0)$ für eine zeigende Hand und in der Form $(0, 1)$ für ein anderes Objekt kodiert. Die dritte Komponente besteht aus einem kontinuierlichen Wert für die Zeigerichtung in Form eines Winkels $\alpha = y_3$ zur Horizontalen des Kamerabildes. Die Klassifikation unbekannter Ausschnitte wird durch die Klasse k mit dem maximalen Output der beiden

Abbildung 7.1: Zeigegeste auf der mobilen AR-Apparatur. Links: Originalbild. Mitte: Ergebnis der Hautfarbensegmentierung. Rechts: Überlagerung des Kamerabildes mit dem Ergebnis der Zeigegestenerkennung, welche hier allein auf Basis des zeigenden Fingers erkannt wird.

Komponenten $k = \arg\max_{i=1,2}(\vec{y}_i(\vec{x}))$ bestimmt. Nur wenn eine Zeigegeste vorliegt, ist der Winkel α relevant.

Der Klassifikator wird auf den handgelabelten Bildausschnitten der zeigenden Hand und Bildausschnitten von Objekten einer Rückweisungsklasse trainiert. Die Trainingsmenge der Rückweisungsklasse enthält entweder Bildausschnitte anderer Objekte oder anderer Handgesten. Ein Nachteil des verwendeten Klassifikators ist, dass er nur mit vorher erlernten Objekten zu korrekten Klassifikationsergebnissen kommt. Sollte ein vollkommen unbekanntes Objekt die Hautfarbensegmentierung passieren, könnte dies zur Fehlklassifikation kommen, da es keine universelle Rückweisungsklasse gibt. Der Klassifikator ist dadurch bis zu einem gewissen Grade auf das bekannte Szenario eingeschränkt.

7.2.2 Zeigegestenerkennung bei großer Kamerabrennweite

Bei großer Brennweite der verwendeten Kamera kann kein Bildausschnitt der gesamten Hand verwendet werden. Bei der Evaluation des Systems hat sich gezeigt, dass der Benutzer intuitiv nur mit einem Finger in das Bild geht, um nicht zu große Teile der Szene zu verdecken. Die Abbildungen 7.1 und 7.2 zeigen jeweils Kamerabilder, welche mit den Kameras des Prototypen aufgenommen wurden. Man erkennt, dass nur der zeigende Finger im Bild ist. Der Benutzer ermöglicht hier durch sein an das System angepasstes Verhalten, dass mit sehr geringem Rechenaufwand die Zeigerichtung ermittelt wird. Dazu werden nur im unteren Bereich des Bildes hautfarbene Regionen ausgewertet. Auf dem farbsegmentierten Bildausschnitt wird die zusammenhängende Region mit seiner Ausdehnung und seiner Hauptachse bestimmt. Entspricht die Ausdehnung der eines länglichen Fingers, wird die Hauptachse der Region als Zeigerichtung gedeutet. Diese Einschränkungen ermöglichen eine sehr schnelle Auswertung und sind nur gegenüber unten im Bild befindlichen hautfarbenen und länglichen Objekten empfindlich, was sich in der Praxis nie störend auswirkte. Abb. 7.1 zeigt das Rechenzeit sparende Verfahren. Links ist das Originalbild zu sehen, in dem die Hautfarbe segmentiert wird (Mitte). Im Display erscheint dann die erkannte Zeigerichtung als Lichtstrahl, welcher dem Bild überlagert wird.

7.3 Erhöhung der Bedienungsperformanz durch Systemfeedback

In diesem Kapitel wird auf die Bedeutung und Funktionalität von akustischem und visuellem Feedback für die anwendungsfreundliche Bedienung eines mobilen Systems eingegangen. Die Funktionalität von künstlichen Systemen wird durch multimodale Mensch-Maschine-Interaktionen deutlich erhöht (Stiedl, 2006). Das Systemfeedback für den Benutzer besteht bei diesem System einerseits aus der Überlagerung der Verarbeitungsergebnisse als visuelle Erweiterungen im Display und andererseits aus akustischem Feedback, welches für verschiedene Ereignisse jeweils ein prägnantes Systemgeräusch verwendet. Dies erhöht die Benutzerfreundlichkeit und die Bedienungsperformanz. Exemplarisch wurde für das Zeigen auf Objekte eine Evaluation durchgeführt, welche den Nutzen von Multimodalität und Systemrückmeldungen belegt. Bei einem System, in dem der Mensch mit der Maschine kommunizieren kann, ist es möglich, dass der Benutzer eine fehlerhafte Verarbeitung des Systems durch angepasstes eigenes Handeln ausgleicht. Die folgende Evaluation verdeutlicht diesen Aspekt. In den vorigen Kapiteln wurde bereits die Zeigegestenerkennung beschrieben. Sie dient dazu, Objekte per Zeigegeste zu referenzieren und so ein gemeinsames Verständnis der Umgebung von Benutzer und Maschine zu erreichen, indem das System erkennt, welches Objekt vom Benutzer gemeint ist. Nehmen wir an, dass der Benutzer in Abb. 7.2 auf die Espressotasse zeigen möchte, das System aber die Zeigerichtung falsch auswertet und dadurch den gemeinsamen Fokus auf den Anspitzer setzt. Sobald der Benutzer dies zurückgemeldet bekommt, kann er darauf reagieren. In der Abbildung wird durch den hervorgehobenen Lichtkegel die Zeigerichtung visualisiert und durch den gelben Rahmen das Objekt hervorgehoben, welches das System als das refenzierte erkannt hat. Würde in diesem Fall der Rahmen auf dem Anspitzer liegen, könnte der Benutzer seine Zeigerichtung entsprechend leicht verändern, bis das gewünschte Objekt im Fokus des Systems liegt, und somit Ungenauigkeiten bei der Auswertung der Zeigerichtung ausgleichen.

Nicht nur visuelle Erweiterungen, sondern auch ein akustisches Feedback kann bereits den Benutzer bei der Bedienung des Systems unterstützen. Bei der Zeigegeste ist die Information, wann das System den Fokus von einem Objekt zum nächsten wechselt, für den Benutzer sinnvoll. Wie im Kap. 5.3 vorgestellt, befindet sich im Aufmerksamkeitssystem integriert das FSD-Modul auf der subsymbolischen Ebene. Dieses Modul stellt plötzliche räumliche Sprünge des ermittelten gemeinsamen Fokus fest, wenn also das zu referenzierende Objekt wechselt (Bax u. a., 2003). Dieses Ereignis wird durch einen „bop"-Sound akustisch untermalt, wenn der Fokus zu einem anderen Maximum $\hat{s}_j$ der kortikalen Karte C wechselt. Solch ein Ereignis wird detektiert, wenn gilt:

$$\left| \left(\frac{1}{\Delta t} \sum_{i=1}^{\Delta t} \hat{s}^*(t-i) \right) - \hat{s}^*(t) \right| > d, \tag{7.1}$$

wobei $\hat{s}^*(t)$ das Maximum von C bezeichnet, welches dem aktuellen gemeinsamen Fokus der Aufmerksamkeit zum Zeitpunkt t am nächsten liegt. Der Parameter Δt muss

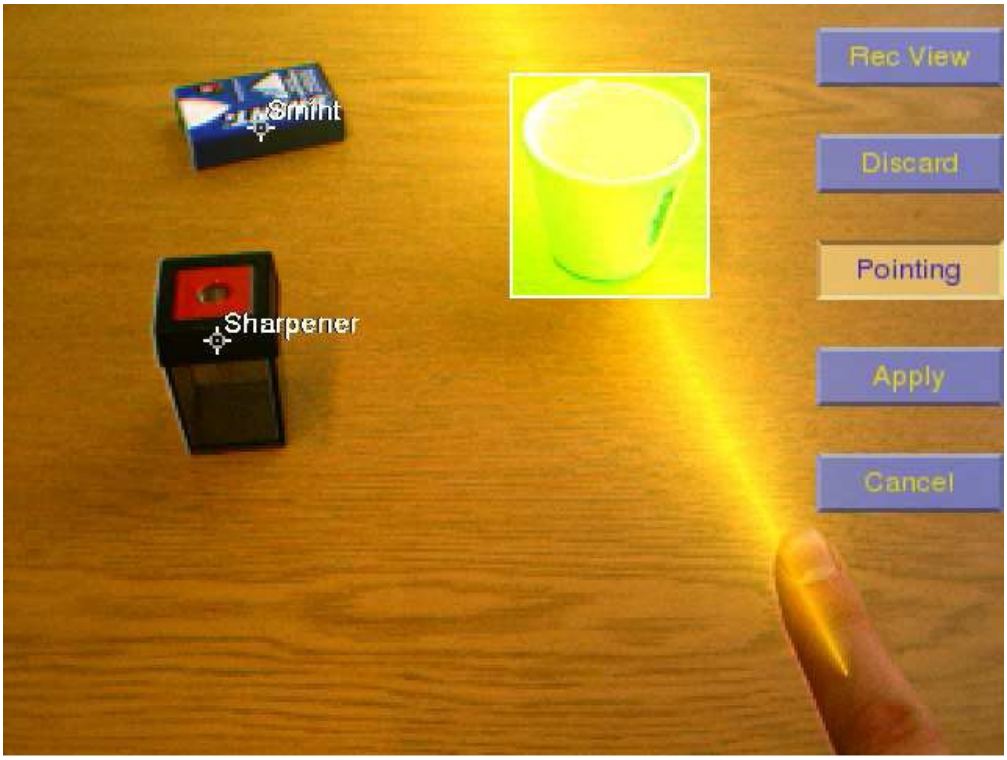


Abbildung 7.2: Zeigegeste auf der mobilen AR-Apparatur.

aufgrund der Verarbeitungsgeschwindigkeit an die Framerate angepasst werden. Der Schwellenwert d kann auf Basis der Distanzmatrix der Maxima $\hat{s}_i^*$ abgeschätzt werden.

Evaluation

Um die Systemperformanz in Abhängigkeit von verschiedenen Feedbacks zu evaluieren, wurde eine leicht nachzuvollziehende einfache Zeigeaufgabe verwendet. Dazu sollte ein Proband, wie in Abb. 7.3 links gezeigt, einen von sechs weißen Kreisen auf schwarzem Hintergrund per Zeigegeste referenzieren. Der Abstand zwischen der Hand und den Zielkreisen beträgt ca. 40 cm. Somit haben die Kreise eine Ausdehnung von ca. 1,7°. Der Abstand zwischen den Kreismittelpunkten wird von 4° bis 28° variiert. Um die Genauigkeit der Zeigegestenbestimmung für das Zeigen auf Details zu ermitteln, wurde ergänzend mit kleineren Kreisen mit einem Durchmesser von 0,9° und einem Abstand von 2° ein zusätzlicher Experimentteil durchgeführt.

Bei dem Experiment wurde der Proband durch Nennung der Nummer aufgefordert, auf einen der sechs Kreise zu zeigen. Diese Nummer wurde zufallsgeneriert. Um Randeffekte zu vermeiden, wurden nur die inneren Kreise verwendet. Ein Treffer wurde gezählt, wenn der Proband innerhalb von 3 Sekunden erreicht hat, dass der Fokus des Systems auf dem Kreis liegt. Das Experiment wurde unter folgenden drei Bedingungen wieder-

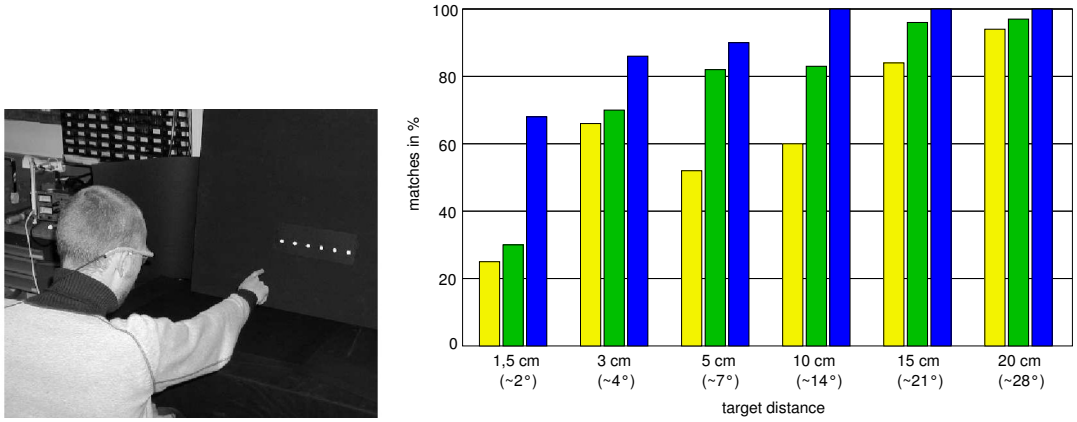


Abbildung 7.3: Links: Evaluationsumgebung. Rechts: Ergebnisse der Untersuchung zur Exaktheit der Zeigegestenerkennung.Die blauen Balken geben die Ergebnisse ohne jedes Feedback wieder. Die Ergebnisse werden durch akustisches Feedback (grün) und besonders stark durch visuelles Feedback (blau) erhöht.

holt: Im ersten Fall erhielt der Proband keinerlei Rückmeldung über den Erfolg seiner Zeigegeste bzw. über die Systemergebnisse. Bei der Bedingung *auditory feedback* erhielt der Benutzer das oben geschilderte akustische Signal eines „bop"-Sounds, wenn das System den Fokuspunktwechsel von einem Zielpunkt auf den nächsten erkannt hat. Bei *full visual feedback* sah der Proband das Ergebnis der Auswertung auf einem Display eingeblendet. Abb. 7.3 zeigt, dass die besten Ergebnisse erreicht wurden, wenn der Proband die Erkennungsergebnisse visualisiert bekommt. Die Ergebnisse ganz ohne Feedback sind am schwächsten. Bereits bei einem Abstand der Kreise von 14° sinkt der Anteil der Treffer auf unter 60%. Unter Verwendung des akustischen Feedbacks werden bei Abständen ab 7° vergleichbar gute Werte erreicht wie mit visuellem Feedback. Erst bei geringeren Abständen führt das akustische Feedback zu keiner signifikanten Steigerung.

Als wesentliche Rückschlüsse, die man aus diesen Testergebnissen schließen kann, sind zu nennen, dass eine Systemrückmeldung die Systemperformanz wesentlich steigern kann, da der Benutzer

1. einzelne Zeigegesten auf einen Zielpunkt adjustieren und

2. sich an das Systemverhalten anpassen kann.

Durch die Anpassung seines Verhaltens an das System kann somit die effektive Auflösung der Zeigegestenerkennung signifikant gesteigert werden, da sie nicht allein von der Genauigkeit der Systemantwort abhängt, sondern vom Benutzer gesteuert werden kann. Diese Verhaltensanpassung des Benutzers konnte schon allein bei der Verwendung des akustischen Feedbacks beobachtet werden, da hier die Probanden beim Hin- und Herzeigen sinnvolle Informationen vom System zurückgemeldet bekommen haben, mit denen sie ihre Zeigegesten anpassen konnten. Der Mensch in einer Interaktionsschleife

mit dem künstlichen System steigert somit deutlich die Performanz, da Systemungenauigkeiten aufgrund der Systemrückmeldungen vom Benutzer ausgeglichen werden können.

7.4 Fingerspitzenerkennung

7.4.1 Bedienung des Menüs

Für die Auswahl der einzelnen Systemfunktionen wurde ein Menü entwickelt, welches im rechten Teil des Displays als visuelle Erweiterung eingeblendet wurde. Dieses besteht aus einzelnen Menüpunkten, welche als halbtransparente, rechteckige Tasten mit dem entsprechenden Label der Funktion (meist in abgekürzter Form) dem aktuellen Kamerabild überlagert werden. Für die Bedienung des Menüs sollten keine zusätzlichen Hilfsmittel verwendet werden, vielmehr sollte die Navigation allein durch Fingergesten oder durch Sprache gesteuert werden können. Um eine verlässliche Bedienung des Menüs zu gewährleisten, müssen alle Tasten zuerst selektiert werden, bevor sie gedrückt bzw. aktiviert werden. Die Selektion kann per Sprache durch das Kommando „Wähle Knopf x" oder durch eine Geste erfolgen. In diesem Fall wählt der Benutzer eine Taste aus, indem er den Finger auf die virtuelle Taste legt (vgl. Abb .7.4).

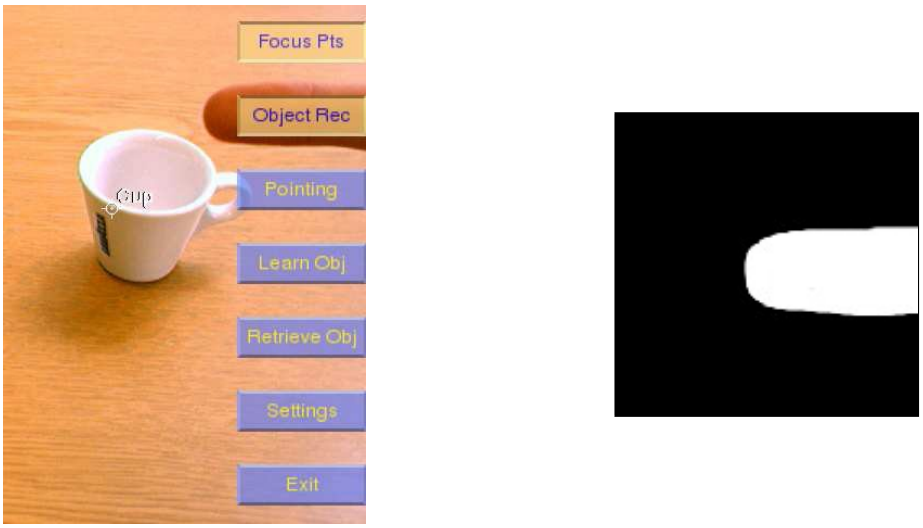


Abbildung 7.4: Links: Menükontrolle per Fingerspitze: Der Benutzer hat bereits den Menüpunkt für die Darstellung der Fokuspunkte gewählt und führt nun eine „Drückbewegung" aus, um die Objekterkennung anzuschalten. Das System analysiert die Position und die Bewegungstrajektorie, um die Ereignisse „Taste selektiert" und „Taste gedrückt" zu erkennen. Rechts: Fingerspitzenschablone, mit der die hautfarbensegmentierte Region gematched wird.

Nach der Selektion kann die Taste gedrückt werden, indem der Benutzer den Finger nach innen gerichtet bewegt. Per Spracherkennung muss nach der Selektion das Kom-

mando „Drücke Knopf x" gegeben werden. Die Menüoperationen werden sowohl durch visuelles als auch durch akustisches Feedback für die Selektion und das Drücken wiedergegeben. Die Tasten wechseln je nach Zustand ihre Farbe, so dass der Benutzer jederzeit über den Zustand informiert ist und bei der Bedienung des Menüs ein Feedback bekommt, ob das System die Auswahl eines Menüpunktes entweder per Fingerspitzen- oder Spracherkennung erkannt hat. Es gibt drei Zustände, die jeweils mit einer Farbe gekennzeichnet werden: i) im Ruhezustand sind die Knöpfe blau, ii) wenn sie per Finger oder per Sprache selektiert werden sind sie hellgrün und iii) wenn sie dann gedrückt worden sind, sind sie gelb. Es gibt zwei Arten von Menüpunkten: Die sogenannten „Togglebutton" sind Menüpunkte, die, wenn sie gedrückt werden, ein einmaliges Ereignis auslösen und dann in den Ursprungszustand zurück gehen. Die sogenannten „Switchbuttons" bleiben nach dem Drücken in dem ausgelösten Zustand und zeigen somit einen Systemzustand an, während eine bestimmte Funktion anhaltend ausgeführt wird.

7.4.2 Verfahren zur Analyse der Menüoperation per Fingerbewegung

Für die Erkennung der Selektion und des Drückens einer Taste wird als erster Schritt wie bei der Zeigegestenerkennung die Hautfarbe segmentiert. Anschließend werden die zusammenhängenden Regionen analysiert. Nur der Bereich um das eingeblendete Menü im Display oben rechts wird dabei berücksichtigt. Anschließend wird die Übereinstimmung der erkannten Region mit einer binären Maske, wie in Abb. 7.4, rechts, berechnet. Für das Drücken einer Taste darf die Differenz zwischen einer Maske M^b mit dem Zentrum auf dem Tastenmittelpunkt einer Taste b und der segmentierten Region S über eine gewisse Anzahl an Frames n_F einen Schwellwert λ nicht übersteigen, so dass gilt:

$$|\frac{1}{n_F}\sum_{t=1}^{n_F}\sum_{x,y} M^b_{x,y} - S_{x,y}(t)| < \lambda. \tag{7.2}$$

n_F ist abhängig von der Verarbeitungsgeschwindigkeit des Systems und liegt bei einer realen Zeit von ca. 1,5 s. Erst dann wird die Taste als selektiert visualisiert. Ab diesem Zeitpunkt wird die Trajektorie der segmentierten Region verfolgt. Bewegt sich diese horizontal um mehr als eine halbe Tastenbreite nach innen gerichtet, wird das Drücken der jeweiligen Taste erkannt.

7.4.3 Evaluierung

In dieser Evaluation soll die Effizienz der Menükontrolle mit Hilfe der Fingerspitzenerkennung unter zwei Gesichtspunkten überprüft werden. Zum einen sollte analysiert werden, wie robust das Verfahren auch bei unterschiedlichen Beleuchtungssituationen funktioniert und wie groß dabei die Anzahl der einzelnen Menüitems, welche eingeblendet werden, sein kann, ohne die Performanz erheblich zu senken. Aus diesen Ergebnissen sollte dann die Anzahl der zu verwendenden Menüpunkte für das Menüdesign abgeleitet werden. Zum anderen sollte analysiert werden, wie komfortabel die Bedienung des Menüs mit dem Finger ist, d.h. wie schnell sich der Benutzer an das Verfahren gewöhnt

und sich somit ein gewisser Gewöhnungseffekt auf die Bedienungsgeschwindigkeit und -genauigkeit auswirkt.

Für die Evaluation wird ein Proband aufgefordert, eine bestimmte Taste zu drücken. Die Nummer der Taste wird zufallsgeneriert. Eine erfolgreiche Bedienung der genannten Taste wird gezählt, wenn der Proband innerhalb von 5 s die Taste auswählt und drückt, also das System diese beiden Aktionen richtig erkannt hat. Das Experiment wurde in dem Büroszenario durchgeführt, so dass der Hintergrund sehr heterogen war. Der erste Teil der Evaluation ist unter den folgenden variierenden Parametern durchgeführt worden:

1. Um die räumliche Auflösung der Fingerspitzenerkennung zu analysieren, wurde die Anzahl der Tasten zwischen 3 und 11 variiert. Die Tasten wurden äquidistant verteilt im rechten Bildrand dargestellt, so dass der Bereich für jede Taste in der vertikalen Ausdehnung zwischen einem Drittel und einem Elftel des Bildes betrug.

2. Um die Robustheit der Hautfarbensegmentierung gegenüber verschiedenen Beleuchtungssituationen zu testen, wurden vier verschiedene Beleuchtungsbedingungen verwendet: natürliches Tageslicht, künstliche Raumbeleuchtung und eine künstliche Lichtquelle in Form eines Strahlers, welcher einmal von hinten und einmal von vorne die Szene erhellt.

Für den ersten Teil des Experimentes bekamen die Probanden eine kurze Zeit zur Eingewöhnung, indem die ersten 20 Items nicht ausgewertet wurden. Anschließend wurden 160 Items mit jeweils 40 unter gleicher Beleuchtungsbedingung ausgewertet. Die Ergebnisse wurden über fünf Probanden und die verschiedenen Beleuchtungsbedingungen gemittelt. Tabelle 7.1 zeigt, dass die Fingerspitzenerkennung mit bis zu sieben einzelnen Menüpunkten effizient und robust funktioniert. Eine größere Anzahl von bis zu neun Punkten wäre möglich, jedoch verlängert dies die Bedienungszeit, so dass aus den Ergebnissen abgeleitet werden kann, dass für das Design des Menüs bis zu sieben Menüpunkte zweckmäßig sind. Darüber hinaus sollte das Menü durch Untermenüs strukturiert werden.

Dass nicht nur die Beleuchtungsbedingungen und der heterogene Hintergrund eine Rolle für die Performanz der Fingerspitzenerkennung spielen, sondern vor allem die Anpassungsfähigkeit des Benutzers an die „Systemeigenarten", sollte in einem zweiten Experiment überprüft werden. Dazu wurde bestimmt, wie schnell sich ein Proband an die Bedienung des Systems gewöhnt. Als Maß für die Gewöhnung sollte die Geschwindigkeit, mit der der Proband erfolgreich einen Menüpunkt auswählt und drückt, dienen, welche im Laufe einer Anpassung an das System zunehmen würde. Das Testszenario entspricht dem ersten Experiment, jedoch mit gleichen Beleuchtungsvariationen und der konstanten Zahl von sieben Menüpunkten. Für das Experiment wurden fünf komplett unerfahrene Probanden wie im vorigen Experiment aufgefordert, einen bestimmten Menüpunkt per Fingerspitze auszuwählen. Ein Versuchslauf bestand aus 15 Items. Die Abbildung 7.4.3 (links), zeigt, dass im Schnitt schon bei dem zweiten Versuchsdurchlauf die Anzahl der Treffer auf über 12 pro 15 Items steigt. Bereits beim dritten Durchlauf liegt die Quote bei fast 95 %. Nach dem fünften Durchlauf erzielten nahezu alle Probanden eine 100%ige

Tabelle 7.1: Trefferraten für die Menukontrollevaluierung.

Number of buttons	3	5	7	9	11
Matches	98.6	97.3	90.6	74.6	57.3

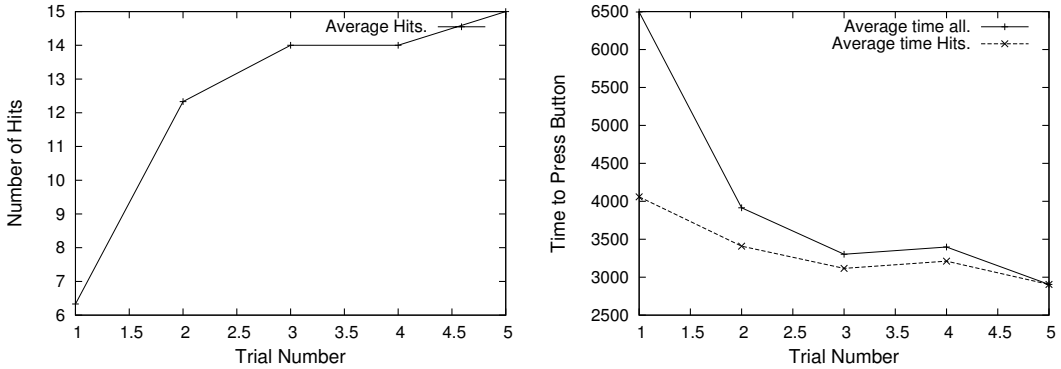


Abbildung 7.5: Links: Durchschnittliche Trefferquote der zweiten Menükontrollevaluierung. Rechts: Durchschnittliche Zeiten in *ms*, die eine Versuchsperson benötigt, um bei der zweiten Menukontrollevaluierungd einen Menüpunkt auszuwählen. Der obere Graf zeigt die Zeit einschließlich der Fehler, der untere Graf repräsentiert die durchschnittliche Zeit der korrekten Auswahl eines Menüpunktes.

Trefferquote. Die Abbildung rechts zeigt die durchschnittliche Zeit, die benötigt wird, um einen Menüpunkt zu selektieren und dann zu drücken. Der obere Graph in der Abbildung gibt die durchschnittliche Geschwindigkeitszunahme für alle Versuche wieder, der untere nur die erfolgreichen. Bereits bei dem dritten Versuchsdurchlauf benötigten die Probanden im Durchschnitt unter 3,5 s. Nach dem fünften Durchlauf wurden für das erfolgreiche Bedienen des Menüs unter 3 s benötigt. Das Experiment zeigt, dass die erfolgreiche Bedienung somit in erster Linie von der Anpassungsfähigkeit des Benutzers abhängt. Durch das Feedback des Systems, dass also die erfolgreiche Selektion und das Drücken eines Menüpunktes sowohl visuell als auch akustisch untermalt wird, kann der Benutzer sein Verhalten schnell an die Verarbeitungscharakteristik des Systems anpassen. Der Mensch erhöht auch hier durch die Integration in die Verarbeitungsschleife die Performanz eines künstlichen computergestützten Systems.

Kapitel 8

Spracherkennung

Für die Spracherkennung zur Steuerung des Systems wurde das von Gernod A. Fink entwickelte ESMERALDA verwendet (Fink, 1999). ESMERALDA steht für *Environment for Statistical Model Estimation and Recognition on Arbitrary Linear Data Arrays* und ist eine integrierte Umgebung für die Entwicklung von Spracherkennungssystemen. Die akustische Modellierung erfolgt durch Hidden-Markov-Modelle auf den durch Vektorquantisierung aus dem Signal extrahierten Merkmalen. Die Sprachmodellierung basiert auf n-Grammen. Beide sind im Bereich der Spracherkennung etablierte Verfahren (vgl. Ephraim und Merhav (2002); Huang u. a. (2001)). Dem Nachteil, dass ein solches System eine große Trainingsmenge benötigt, um Sprache generell zu verstehen, wird hier begegnet, indem für die Bedienung des Systems sowohl der Wortschatz als auch die Grammatik stark eingeschränkt werden kann. Das deklarative grammatikalische Wissen wird in Form einer Grammatik, die von einem LR(1)-Parser (Aho u. a., 1986) geparsed wird, formuliert.

ESMERALDA wurde bereits mit Hilfe der sprecherunabhängigen spontanen Spracherkennungsaufgabe der VERBMOBIL-Umgebung evaluiert und erwies sich dabei als einer der besten Erkenner in diesem Benchmark-Test (Fink, 1999). Aufgrund des robusten, sprecherunabhängingen Klassifikationsverhaltens und der einfachen Einschränkung durch eine kontextfreie Grammatik wurde ESMERALDA bereits in verschiedenen Objekterkennungssystemen und in der Robotik verwendet (McGuire u. a., 2002; Wachsmuth u. a., 2000). Für die Integration in das hier verwendete System konnte eine hohe Robustheit erreicht werden, da der Wortschatz lediglich auf die zu verwendenden Wörter eingeschränkt werden konnte und die Bedienung des Menüs, welches im nächsten Kapitel detailliert erläutert wird, nur einer einfachen und knapp formulierten Grammatik bedurfte, welche in Abb 8 gezeigt wird. Die Nichtterminalsymbole der Grammatik sind durch das vorausgehende $-Zeichen gekennzeichnet. Dem Startsymbol geht ein doppeltes $-Zeichen voraus.

```
$$S:                    $Command | $Response | $End ;

$Command:               $SelectButton | $PressButton | $UnselectButton | $ShowMe
                        | $Label;

$SelectButton:          w"ahle $Name | w"ahle Knopf $Name | w"ahle Knopf $Num |
                        Knopf $Num ausw"ahlen;

$PressButton:           dr"ucke $Name | dr"ucke Knopf $Num | dr"ucke Knopf $Name
                        | dr"ucke ihn;

$Num:                   eins | zwei | zwo | drei | vier | f"unf | sechs | sieben
                        | acht | neun ;

$Name:                  abbrechen | aufnehmen | beenden | Einstellungen |
                        Finde_Objekte | Fingerzeiger aus | Fokuspunkte |
                        Hautfarbe | Lernen | Menu | Objekterkennung |
                        Spracherkennung aus | Strahlzeiger | "ubernehmen |
                        Verwerfen | Zur"ucksetzen ;

$UnselectButton:        Auswahl l"oschen ;

$ShowMe:                zeige mir $Art $Object | zeige mir Knoten $Num $Num ;

$Label:                 das ist $Art $Object | das ist Objekt $Num | das ist das
                        Objekt Nummer $Num ;

$Object:                Tasse | Stift | Glas | Handy| Cafe | Anspitzer |
                        Radiergummi | Tesaroller | Schl"ussel | Geldb"orse |
                        Tacker | Smint ;

$Art:                   die | das | den | eine | ein ;

$Response:              $Zustimmung |     $Ablehnung ;

$Zustimmung:            ja | okay | gut ;

$Ablehnung:             nein | nee | schlecht ;

$End:                   stop | halt | Ende ;
```

Abbildung 8.1: Vereinfachte reguläre Grammatik des Prototypen. $-Zeichen markieren die Nichtterminalsymbole.

Teil III

Interaktives online Lernen

Ein Merkmal kognitiver Systeme ist es, mit der Umgebung zu interagieren und auf veränderte Umweltbedingungen reagieren zu können. Die meisten technischen Systeme, welche eine gewisse künstliche Intelligenz aufweisen und eine Lernfähigkeit besitzen, erlernen ihr Wissen in einer offline-Trainingsphase. Im Hinblick auf kognitive Eigenschaften ist dies unbefriedigend, da während der Laufzeit dieser Systeme kein neues Wissen erlangt werden kann und solch ein System somit nur in engen Grenzen auf sich ändernde Bedingungen reagieren kann. Im Gegensatz dazu ermöglicht das vorgestellte System, während der Laufzeit auf neue Reize in der Umgebung zu reagieren, indem es - von dem menschlichen Experten geleitet - neues Objektwissen erlernen kann. Dieses interaktive Erlernen neuen Wissens, speziell Objektwissens, wird in diesem Teil vorgestellt.

Um einem mobilen System zur Laufzeit Objektwissen zu vermitteln, spielt die Interaktion zwischen Mensch und Maschine eine wichtige Rolle. Nachdem im vorigen Teil einige wichtige Basiskomponenten für die Kommunikation vorgestellt wurden, soll nun gezeigt werden, wie diese in den komplexen Funktionsabläufen für die Umsetzung der erforderlichen Funktionalität einer Brille mit Gedächtnis eingesetzt werden. Die verschiedenen Funktionen werden dabei von einer Vielzahl von einzelnen Modulen und durch mehrfache wechselseitige Kommunikation bewerkstelligt. Diese Abläufe machen eine Organisation auf der Systemseite notwendig. Dazu wurde als zentrale Schaltstelle des künstlichen Systems ein Controller entwickelt, welcher vom Benutzer durch Kommunikation gesteuert, die teils synchrone und teils iterative Bearbeitung der Einzelaufgaben durch Aktivierung von Perzeptionskanälen, Verarbeitungsprozessen und geeigneter Kommunikation ermöglicht. Die dem Benutzer zur Verfügung stehenden Systemfunktionen sind dabei in Form eines Menüs strukturiert. Die beiden Hauptfunktionen des Systems, das iterative Labeln von Bilddaten und das Lernen neuer Objekte zur Laufzeit, stellen neben der Abfrage aktueller Objektaufenthaltsorte zum Wiederfinden verlorener Gegenstände den Kern des Menüs dar. Dieser Teil der Arbeit wird anhand des hierarchisch strukturierten Menüs vorgestellt. Als erstes wird die Architektur und die Kontrolle des Systems erklärt. Im Anschluss daran wird der hierarchische Aufbau des Menüs erläutert, welcher die folgenden Kapitel gliedert. Die Darstellung der Umsetzung der einzelnen Systemfunktionen mit den dazugehörigen Untermenüs bildet den Hauptteil. Hier wird detailliert beschrieben, wie durch Mensch-Maschine-Interaktion die Systemanforderungen umgesetzt wurden. An den entsprechenden Stellen werden die gewählten Verfahren in den Teilkapiteln evaluiert.

Kapitel 9

Systemsteuerung

9.1 Systemarchitektur und Kontrolle

Wie in Abb. 9.1 gezeigt, besteht das entwickelte System aus verschiedenen, teils unabhängig voneinander laufenden Modulen, die in einer flachen Architektur organisiert sind. An dieser Stelle soll nur ein Überblick über den Aufbau gegeben werden. Die einzelnen Abläufe und Funktionen werden anschließend in den entsprechenden Kapiteln im Detail erklärt.

Das System besteht im Wesentlichen aus drei Basistypen von Modulen:

- Eingabemodule für die Verarbeitung von Bild und Sprache;
- Ausgabemodule für die Darstellung des Bildes einschließlich der künstlichen Erweiterungen und des Menüs und die akustischen Signale als Feedback für den Benutzer;
- Kontrollmodul als zentrale Komponente zur Überwachung der anderen Module.

Ein vierter Typ wird einzig durch das SOM-Modul repräsentiert, welches ausschließlich indirekt über das Kontrollmodul mit den Eingabe- und den Ausgabemodulen in Verbindung steht. Die einzelnen Eingabemodule arbeiten unabhängig voneinander und liefern einen fortlaufenden Strom an Verarbeitungsergebnissen. Die Hautfarbensegmentierung und die Merkmalskarten werden direkt auf dem RGB-Bild berechnet. Die Kortikale Karte basiert hingegen auf den Ergebnissen der Merkmalskarten und liefert zusammen mit dem Originalbild die Eingabe für das Objekterkennungsmodul. Zusätzlich wird die Kortikale Karte zusammen mit den Ergebnissen der Hautfarbenerkennung für die Zeigegestenerkennung verwendet.

Parallel zur Bildverarbeitung wird das Audiosignal des mobilen Mikrophons vom Spracherkennungsmodul verarbeitet.

Die zentrale Komponente des Systems ist der Controller, der als Zustandsmaschine realisiert ist. Die einzelnen Zustände korrespondieren zur aktuell zu bearbeitenden Aufgabe, wie z.B. der Aufnahme von Bilddaten oder der Strukturierung von Bilddaten mithilfe der SOM, und werden durch das Einblenden des entsprechenden Menüs visualisiert. Abhängig vom Zustand wertet der Controller die jeweils relevanten Daten der Eingabemodule aus, schaltet gegebenenfalls zwischen den Zuständen hin- und her und sendet je nach Bedarf Daten entweder zum SOM-Modul oder zu den Ausgabemodulen. Diese

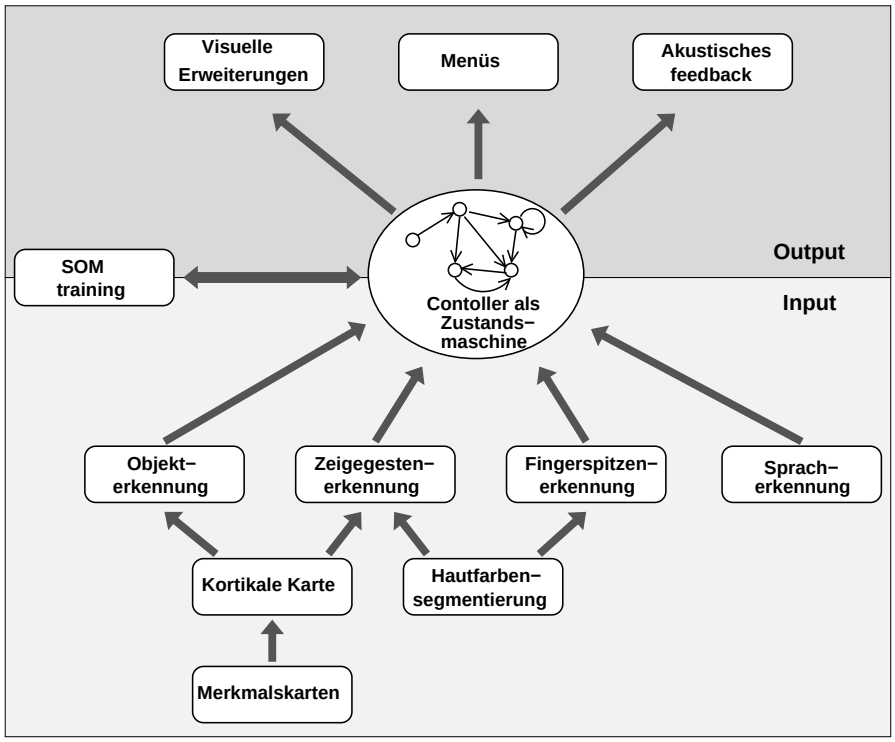


Abbildung 9.1: Systemarchitektur: Eingabemodule verarbeiten, vom als Zustandsmaschine realisierten Kontrollmodul gesteuert, parallel Daten. Systemzustände entsprechen den variierenden Modi und Aufgaben. Das Kontrollmodul liefert die Informationen für die visuellen und akustischen Ausgabemodule. Das SOM-Modul erhält seine Eingaben und liefert seine Ausgaben über den Controller.

liefern dem Benutzer, über das Kontrollmodul gesteuert, die angeforderten Ausgaben. Das AR-Modul überlagert dem im Display dargestellten Kamerabildstrom verschiedene Systeminformationen. Je nach Zustand und Aufgabe können dies die segmentierte Hautfarbenregion, errechnete Fokuspunkte, erkannte Objektlabel oder die Visualisierung der SOM sein.

Das Menümodul stellt die aktuellen Menüpunkte einerseits als Visualisierung des aktuellen Zustandes dar und zeigt andererseits dem Benutzer die verfügbaren Funktionen. Neben der visuellen Darstellung liefert das akustische Signal-Modul verschiedene Geräusche, um auf Verarbeitungsergebnisse hinzuweisen, dies kann z.B. ein akustisches Signal sein, welches erklingt, wenn das Spracherkennungsmodul ein Kommando erkannt hat oder ein Menüpunkt erfolgreich ausgewählt wurde etc.

Um Rechnerkapazitäten freizugeben, aktiviert und deaktiviert das Kontrollmodul zur Zeit nicht verwendete Module.

9.2 Design des Menüs

Im Folgenden wird ein kurzer Überblick über die verschiedenen Funktionalitäten der Demoversionen des Systems gegeben und das hierarchisch aufgebaute Menü mit den jeweilig dazugehörenden Funktionen vorgestellt. Diese Strukturierung wird in den folgenden Kapiteln aufgegriffen und die einzelnen Punkte des Hauptmenüs werden an entsprechender Stelle im Detail vorgestellt. Abbildung 9.2 zeigt den baumförmigen Aufbau des Menüs. Das Hauptmenü, welches Kern dieses Abschnitts ist, ist in der zweiten Spalte dargestellt. Die jeweiligen Untermenüs aus der dritten und vierten Spalte werden bei der Beschreibung der einzelnen Funktionen in den jeweiligen Kapiteln erklärt.

Alle Menüs werden im Display auf der rechten Seite in Form von einzelnen Menüpunkten eingeblendet, so dass sie mit der rechten Hand bedienbar sind. Die einzelnen Menüpunkte werden als rechteckige Tasten mit dem entsprechenden Label der Funktion (meist in abgekürzter Form) eingeblendet. Da es sich bei VAMPIRE um ein internationales EU-Projekt handelt, sind die Menüpunkte Abkürzungen der Funktionen in englischer Sprache.

Im Grundzustand befindet man sich außerhalb des Hauptmenüs, und nur die Taste *Menu* ist oben rechts im Bild zu sehen, so dass die Verdeckung im Bild möglichst gering ist. Drückt man per Sprachkommando oder per Fingergeste den *Menu*-Knopf, gelangt man in das Hauptmenü.

Für das Verlassen des Hauptmenüs oder eines Untermenüs ist jeweils der unterste Menüpunkt eine *Exit*- respektive eine *Cancel*-Taste. Das Hauptmenü ist bei den beiden beschriebenen Versionen der Demos leicht unterschiedlich. In der ersten Version der Online-Lern-Demo ist der erste Menüpunkt *Focus Point*. Ist dieser Knopf aktiv, wird das Zentrum der interessanten Bildregionen, die das Aufmerksamkeitsmodul gefunden hat, mit einem Kreuz markiert. Die Ausdehnung der dazugehörigen ROI wird mit einem eingrenzenden Rechteck angezeigt. Dieser Menüpunkt wird in der Hauptdemo des hier vorgestellten Systems aufgrund der vorher ermittelten sinnvollen Anzahl von sieben Menüpunkten durch den Punkt *SOM* ersetzt und der Menüpunkt *Focus Point* wird in das *Learn Obj*-Untermenü verschoben. Durch Auswahl des Menüpunktes *SOM* gelangt man in das SOM-Menü, welches das iterative Labeln von Bilddaten steuert und in Kap. 10 detailliert beschrieben wird.

Der zweite Hauptmenüpunkt trägt die Bezeichnung *Object Rec* und gehört wie der Menüpunkt *Focus Points* zu der Kategorie der „Switchbuttons". Ist diese Taste gedrückt, ist sie solange aktiv, bis sie erneut gedrückt wurde. Es werden dabei die erkannten Objektlabels rechts neben den dazugehörigen Fokuspunkten eingeblendet. Diese Label werden pro Frame ausgewertet und aktualisiert. In Kap. 11 wird die Objekterkennung mit einem Rechenzeit minimierenden Verfahren vorgestellt. Der dritte Hauptmenüpunkt *Pointing* aktiviert die Zeigegestenerkennung. Sobald nun ein zeigender Finger im Bild erkannt wird, visualisiert ein gelblicher Kegel von der Fingerspitze ausgehend die Richtung der erkannten Zeigegeste. Die Intensität des gelben Kegels nimmt dem Aufmerksamkeitsmodul entsprechend nach außen gaussförmig ab. Zusätzlich wird eine erkannte Zeigegeste mit einem anhaltenden surrenden Geräusch akustisch untermalt. Der nächste

Menüpunkt *Learn Obj* führt in das Untermenü zum interaktiven Objektlernen, welches in Kap. 12 beschrieben wird. Mit Hilfe dieses Menüs können dem System zur Laufzeit neue Objekte durch Präsentation von verschiedenen Objektansichten beigebracht werden.

Der Hauptmenüpunkt *Retrieve Obj* führt in den Abfragemodus. In diesem Zustand wartet das System auf die Frage nach einem Objekt. Hier wird die eigentliche Anforderung an die Brille mit Gedächtnis erfüllt, indem das System die Antwort auf die Frage „Wo ist mein Schlüssel?" gibt. Ist das nachgefragte Objekt im Bild, wird dieses durch einen roten Rahmen hervorgehoben. Ist das gesuchte Objekt nicht im aktuellen Kamerabild zu sehen, wird die letzte Ansicht dieses Objektes verkleinert eingeblendet. Diese Ansicht sollte dem Benutzer genügen, das Objekt wiederzufinden. Die entsprechenden Abläufe werden in Kap. 13 beschrieben.

Der sechste Hauptmenüpunkt *Settings* führt in das Untermenü für die Systemeinstellungen. Dieses Menü dient dazu, im laufenden System Parameter an wechselnde Umweltbedingungen anzupassen, einzelne Funktionen bei fehlerhafter Funktion auszuschalten oder deren Auswertung zu unterdrücken oder aber bestimmte Datenmengen zu sichern, sei es die Bilddatenbank oder aber ein trainierter Klassifikator (Details siehe Kap. 14).

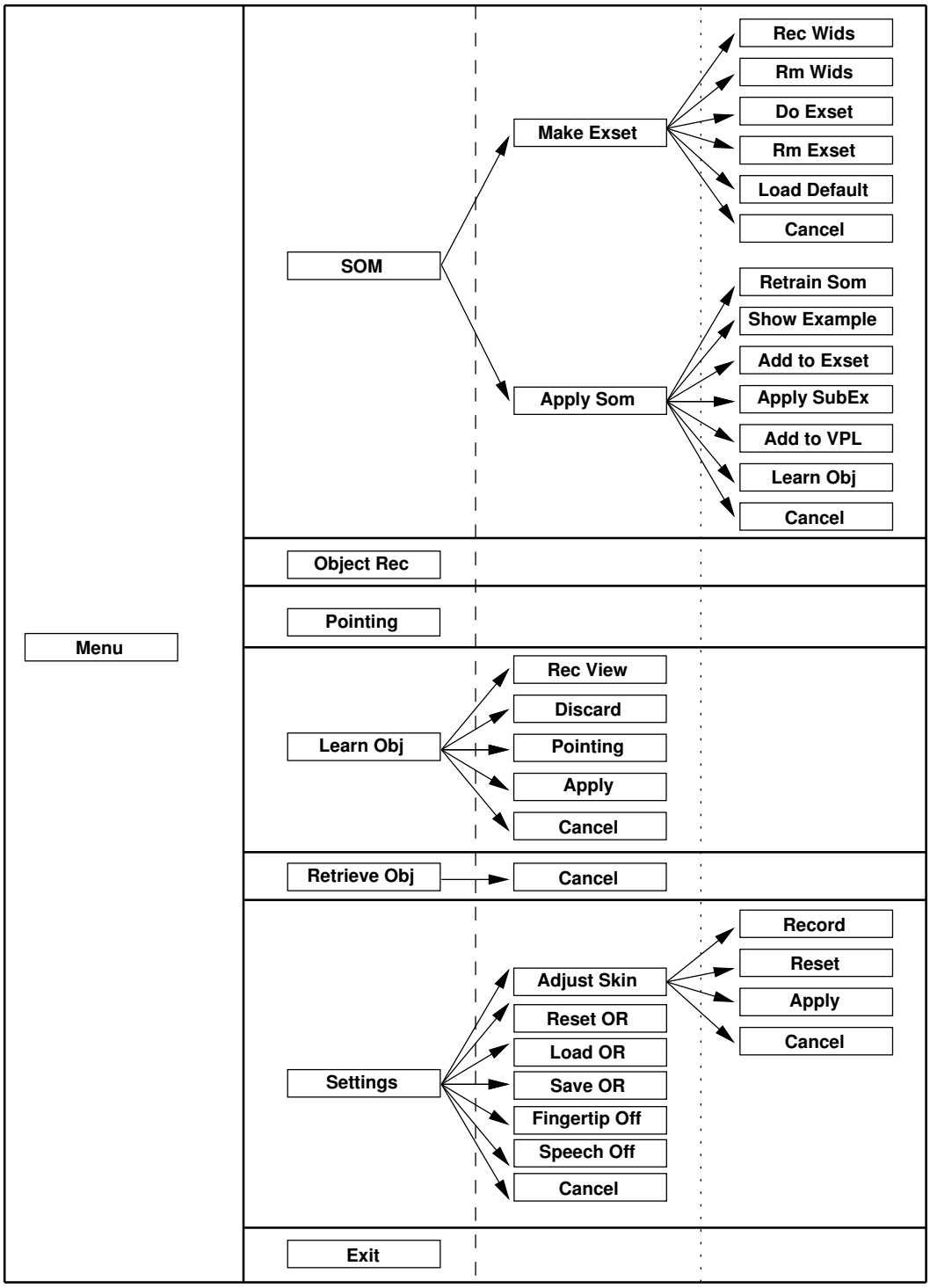


Abbildung 9.2: Design des Menüs. Die zweite Spalte zeigt das Hauptmenü. In der dritten und vierten Spalte wird der hierarchische Aufbau der Untermenüs präsentiert.

Kapitel 10

Lernen von Objekten in der Umgebung durch iteratives Labeln

10.1 Motivation

Für die Antwort auf die Frage nach dem Aufenthaltsort eines bestimmten Objektes benötigt ein System einen Objekterkenner. Bereits in den vorigen Kapiteln wurde der hier verwendete ansichtsbasierte Objekterkenner vorgestellt. Der Vorteil der ansichtsbasierten Objekterkennung liegt darin, dass das Wissen über Objekte komfortabel aus den Ansichten erlangt werden kann und keine aufwändige 2- oder 3-D-Modellierung notwendig ist (vgl. Koenderink und van Doorn (1979); Murase und Nayar (1995); Mel (1997)). Im Gegensatz zu Laborbedingungen oder den Bedingungen in industriellen Produktionsstraßen sind die Herausforderungen, in einer natürlichen Umgebung Objekte zu erkennen, erheblich größer. Objekterkennung in einer natürlichen Umgebung auf Basis von Ansichten muss einen weiten Bereich in Bezug auf verschiedene Bedingungen abdecken. Die Objekte sollten aus unterschiedlichen Blickwinkeln, aus unterschiedlicher Entfernung und unabhängig von der Beschaffenheit des Hintergrundes erkannt werden. Ein weiterer wichtiger Aspekt in einer natürlichen Umgebung ist, dass das Licht im Laufe des Tages erheblich variiert. Dies bezieht sich sowohl auf die Intensität als auch auf die spektrale Zusammensetzung, welche in einem Szenario, wie einem Büro, durch die Verwendung verschiedener Lichtquellen hervorgerufen wird. Für die ansichtsbasierte Objekterkennung bedeutet dies, dass eine große Anzahl an verschiedenen Ansichten von Objekten unter verschiedensten Bedingen zum Trainieren, also für den Lernprozess, vorliegen muss, um trotz schwankender Bedingungen robust eine hohe Klassifikationsperformanz zu gewährleisten. Diese Ansichten müssen dem überwachten Lernverfahren eines Klassifikators entsprechend richtig gelabelt sein. Somit stellt sich die Frage, wie man (wie in Abb. 10.1 visualisiert) einen gelabelten Datensatz aus einer natürlichen Umgebung erhält.

Die Aufgabe besteht darin, einerseits eine ausreichend große Zahl von Ansichten zu erzeugen und diese andererseits zu labeln. Unter Laborbedingungen können Objektansichten verhältnismäßig leicht z.B. durch die Verwendung von Roboter gesteuerten Aufnahmen mit Hilfe eines Drehtellers erzeugt werden, wie bei der Erstellung der etabliertesten Objektdatenbank COIL (Murase und Nayar, 1995; Nene u. a., 1996b). Bei der Verwendung eines mobilen Systems in einer natürlichen Umgebung eignen sich solche künstlich erzeugten Datenbanken zum Trainieren des Klassifikators nicht. Die Ansichten

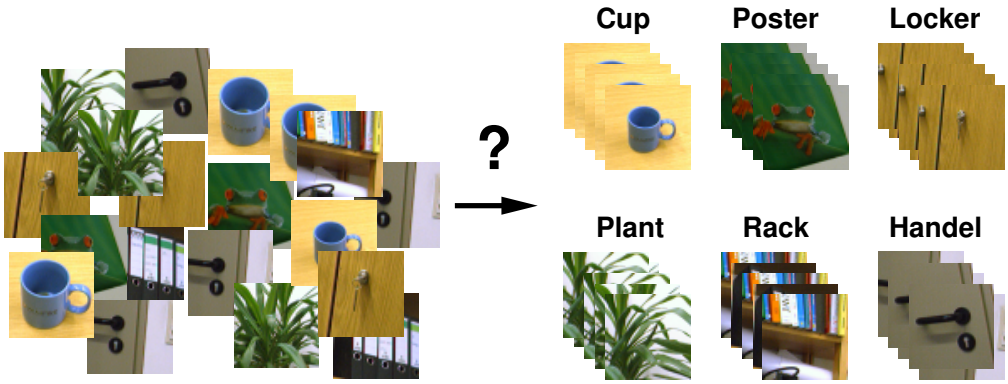


Abbildung 10.1: Wie erzeugt man komfortabel gelabelte Bilddaten für die Trainingsphase eines ansichtsbasierten Objekterkenners?

der Objekte müssen der Perspektive des mobilen Systems und des Benutzers entsprechen und die heterogene Umgebung mit abdecken. Daher ist es unabdingbar, die Ansichten aus der Benutzerperspektive in der Umgebung des Einsatzzwecks zu erlangen und dann in „Handarbeit" zu labeln.

Eine Variante ist es, kleinere Objekte direkt dem System in verschiedenen Ansichten zu präsentieren. Diese Möglichkeit wird in Kap. 12 vorgestellt. Dieses Verfahren benötigt jedoch pro Objekt einen hohen Aufwand und ist daher nur auf wenige Objekte beschränkt. Außerdem können auf diese Weise nur Objekte unter bestimmten Voraussetzungen komfortabel präsentiert werden. Insbesondere größere Objekte oder auch kombinierte Objekte, wie z.B. ein typisches Bücherregal, lassen sich auf diese Art nicht sinnvoll aufnehmen.

Vom kognitiven Standpunkt aus betrachtet, deckt dieses Verfahren ebenfalls nicht die Spannweite ab, mit der Kinder Objekte erlernen. Kinder erlernen die meisten Objekte nicht in einer einmaligen Phase, indem man ihnen Objekte zur Ansicht in die Hand gibt und den Namen nennt. Vielmehr erkunden sie ihre Umgebung und bilden sich Objektkonzepte durch Beobachtung und den Umgang und die Interaktion mit den Objekten, oft noch bevor sie den Namen des Objektes erlernen.

Diese Überlegungen führten dazu, für das System ein Modul zu entwickeln, welches unüberwacht Objektansichten sammelt und aufgrund von Bildmerkmalen gruppiert, um so allein auf Basis der Ansichten eine Vorstrukturierung der aufgenommen Bilddaten in eine Art Objektkategorien zu erreichen. Das bereits beschriebene Aufmerksamkeitsmodul ermöglicht aufgrund der kontextfreien Merkmalskarten bereits eine Lokalisation von potentiell interessanten Bildregionen. Um diese ROIs sollen nun Bildausschnitte gesammelt werden, während sich der Benutzer in seiner Umgebung frei bewegt. Diese Ausschnitte, welche zu einem möglichst großen Teil Ansichten von Objekten beinhalten, sollen ohne größeren Arbeitsaufwand des Benutzers vorstrukturiert werden. Ziel ist es, auf Basis dieser Vorstrukturierung mit erheblich geringerem Aufwand die gesammelten Bilddaten vom Benutzer komfortabel zu labeln.

Für diese Vorstrukturierung von Bilddaten eignen sich sehr gut die von Kohonen vor-

geschlagenen Selbstorganisierenden Karten (Kohonen, 1995). Dieses unüberwachte Lernverfahren bietet sich für diese Aufgabe an, da es die Projektion von hochdimensionalen und hochgradig nicht linearen Datenverteilungen auf nur zwei Dimensionen ermöglicht und dabei die lokalen Nachbarschaftsbeziehungen aufgrund von Ähnlichkeiten der Bildmerkmale erhält und auf ein leicht zu visualisierendes zweidimensionales Gitter abbildet (vgl. Martinetz und Schulten (1994); Villmann u. a. (2003)). SOMs werden bereits vielfach für unüberwachtes Clustern von hochdimensionalen Daten und für deren Visualisierungen durch Projektion auf zwei Dimensionen eingesetzt, wie z.B. in Somervuo und Kohonen (1999) oder Tokutaka u. a. (1999). Eine mit diesem System vergleichbare Anwendung ist das PICSOM-System, welches Bilder durch baumförmige SOMs strukturiert und dadurch einen Anwender bei der Suche nach Bilddaten unterstützt (Laaksonen u. a., 2002). Einen Überblick über die vielfältigen Anwendungen von Selbstorganisierenden Karten geben Oja u. a. (1999).

Bereits für die Zeigegestenerkennung (Kap. 7.2) ergab sich das Problem, geeignete gelabelte Bilddaten für das Training des Klassifikatorts zu bekommen. In einem ersten Vorversuch wurde ein Verfahren entwickelt, diese Bilddaten, die aufgrund der Domäne stärker eingeschränkt sind, unter Zuhilfenahme von Selbstorganisierenden Karten mit dem Werkzeug VALT zu labeln (Heidemann u. a., 2004b). VALT steht für *Visualisation And Labelling Toolkit* und wurde von Saalbach (2001) vorgestellt. Das Verfahren wird im folgenden Abschnitt beschrieben. Aufgrund der einfacheren Datenbasis und der Möglichkeit, das Training verteilt auf mehreren Rechnern offline durchzuführen, war hier die Verwendung der Rohbilddaten möglich.

Für das Iterative Labeln von Objektbilddaten mit einem mobilen System eignet sich diese Datenrepräsentation nicht. Um mit den großen Datenmengen auf einem mobilen System vernünftig umgehen zu können, bedarf es einer erheblichen Dimensionsreduktion. Da die Vorstrukturierung nur so gut sein kann, wie die Datenrepräsentation es zulässt, wird eine Dimensionsreduktion benötigt, welche die charakteristischen Bildeigenschaften und die Datenvarianz erhält und hochgradig diskriminativ abbildet. Nur wenn diese Bedingungen erfüllt sind, kann die Verwendung der SOM zu sinnvollen Clustern führen und das Labeln von Bilddaten ermöglichen.

Im Kontext von Datenkompressionen stellten Manjunath u. a. (2001) und Sikora (2001) die MPEG-7 Merkmale vor, welche effizient die Mehrheit der Bildcharakteristika erhalten. Die einzelnen MPEG-7-Deskriptoren bilden verschiedene Eigenschaften von Bilddaten ab. Dabei reicht die Spanne von inhaltsbasierten Merkmalen wie Farbe und Kanten bis zu semantischen Beschreibungen von Inhalten. In der Bildverarbeitung werden in erster Linie die visuellen Deskriptoren verwendet, welche kontextfrei auf Basis der reinen Bilddaten errechnet werden. Um die Varianz der Bilddaten in Bezug auf verschiedene Aspekte zu erhalten, eignet sich insbesondere eine Kombination von verschiedenen Typen von Merkmalsdetektoren, wie bereits bei Deselaers u. a. (2004) gezeigt wurde. Für das iterative Labeln sollten die ausgezeichneten Eigenschaften der MPEG-7 Diskriptoren verwendet werden. Welche Kombinationen von MPEG-7-Merkmalen sinnvoll sind, wurde auf Basis der statistischen Analysen der Merkmale von Eidenberger (2003) und Eidenberger (2004) entschieden. Eidenberger verwendete die statistischen Verfahren: Mittel-

wert und Varianz der Deskriptorelemente, ihre Verteilung, hierarchische und topologische Clusteranalyse und Faktorenanalyse. Er zeigte, dass die Gesamtheit der Merkmale hochgradig redundant ist und somit eine geeignete Kombination einzelner Merkmale zu einer erheblichen Dimensionsreduktion führt, ohne dabei die diskriminativen Eigenschaften zu verlieren. Die verwendeten Merkmale werden in Kap. 10.5 beschrieben. Für das mobile System bildet eine Kombination dieser Merkmale die Datenbasis, auf der mit Hilfe der SOM die Daten geclustert und anschließend im Display visualisiert werden. Mit dieser Vor-Kategorisierung ist es dem Benutzer möglich, einzelne Bildmengen, welche ein bestimmtes Objekt darstellen, mit einem Label zu versehenen und andere ungeeignete Bildausschnitte (zum Beispiel mit Hintergrundstrukturen) zu verwerfen.

10.2 Vorversuch für das Labeln von Zeigegesten

Für die Gewinnung der Datenbasis für den Klassifikator der Zeigegestenerkennung wurde ein Verfahren entwickelt, welches einerseits die Aufnahme der Bilddaten automatisiert und anderseits mit einem halbautomatischen Verfahren mit geringem Aufwand ein Labeln dieser Daten ermöglicht.

Für die Erkennung von Zeigegesten wird das weite Spektrum von menschlichen Gesten nur auf zeigende Hände in verschiedene Richtungen eingeschränkt. Alle anderen Gesten sollen nicht ausgewertet werden. Um Bildausschnitte einer zeigenden Hand zu erlangen, wurde das in Kap. 7.2 beschriebene Verfahren angewendet. Dabei sitzt der Benutzer in seiner gewohnten Umgebung am Schreibtisch und zeigt auf vor im stehende Objekte. Mit Hilfe der Hautfarbenerkennung und der darauf folgenden Erkennung von zusammenhängenden Regionen liefert das System Bildausschnitte der Hand. Der Benutzer wurde aufgefordert, nicht nur in verschiedene Richtungen zu zeigen, sondern auch andere Handgesten zu machen, damit für das Klassifikatortraining eine ausreichend große Bildmenge für die Rückweisungsklasse vorhanden war. Wie bereits in Kap.7.2.1 beschrieben, war das Klassifikationsergebnis einerseits, ob es sich um eine zeigende Hand handelte, und andererseits, in welche Richtung die Hand zeigt. Die Aufgabe bestand somit darin, die Bilddaten so zu labeln, dass Bilder von Händen, die in die gleiche Richtung zeigen, ein gemeinsames Label erhalten, so dass Bilder auf Basis der Zeigerichtung gruppiert werden und von nicht zeigenden Händen unterschieden werden. Aufgrund der Kameraperspektive aus dem Blickwinkel des Benutzers schränkte sich die Bildmenge automatisch auf Bilder ein, die im unteren Bildrand grob in die Bildvertikale zeigten. Durch die Feedback-gesteuerte Anpassung des Benutzers an das Systemverhalten war dabei eine möglichst präzise Erkennung nicht notwendig. Somit wurde eine Einteilung in fünf verschiedene Zeigerichtungen gewählt. Für diese Einteilung der Bilddaten wurde das Werkzeug VALT verwendet. VALT ist eine grafische Benutzeroberfläche, mit der man SOMs auf Daten trainieren und auf verschiedene Art und Weise visualisieren kann. Dabei kann man unter anderem die Dimension der SOM selbst bestimmen und verschiedene Ansichten für die einzelnen Knoten wählen. Dies kann entweder die Visualisierung der berechneten Prototypvektoren der SOM oder der dem jeweiligen Prototypen am nächsten liegenden Trainingsbeispiele sein. Die Daten, die auf einen bestimmten Knoten

abgebildet werden, können dann in einem Schritt mit der Benutzeroberfläche komfortabel gelabelt werden. Abb.10.2 zeigt den gesamten Vorgang für das Labeln der Bilddaten.

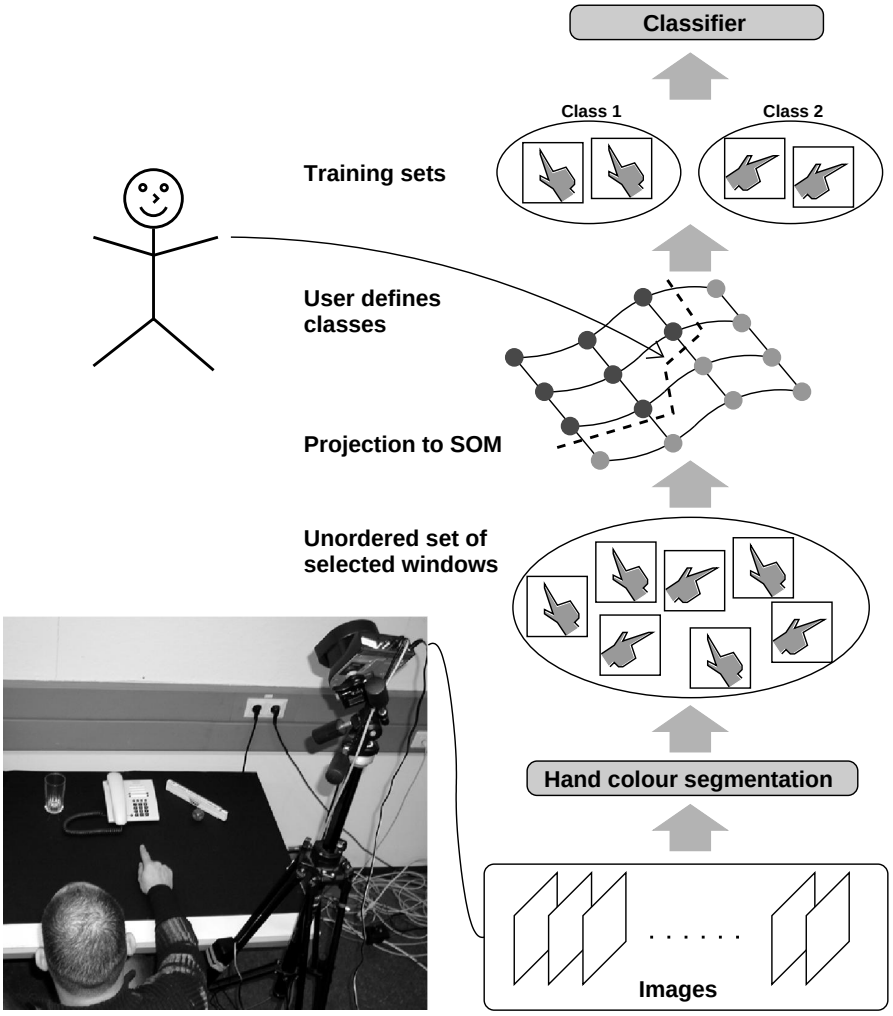


Abbildung 10.2: Aufnahme von gelabelten Bilddaten. Eine Kamera nimmt Sequenzen einer zeigenden Hand auf. Segmentierung von hautfarbenen zusammenhängenden Bildbereichen führt zu einem ungeordneten Satz an Bildausschnitten einer zeigenden Hand. Dieser Datensatz wird auf einer eindimensionalen SOM projiziert, visualisiert und mittels einer interaktiven Benutzeroberfläche in Gruppen mit Bildausschnitten mit annähernd gleicher Zeigerichtung sortiert und benannt. Diese Teilmengen dienen dem Klassifikator als Trainingsmenge (Heidemann u. a. (2004b)).

10.2.1 Partitionieren großer Bilddatensätze mit Hilfe der SOM

Für das Labeln der Bilddaten für die Zeigegestenerkennung und auch für das später beschriebene iterative Labeln von Objektbilddaten wurde eine Standard-SOM, wie sie von Kohonen (1995) vorgeschlagen wurde, verwendet. Die Selbstorganisierenden Karten ermöglichen eine topologieerhaltende Abbildung hochdimenionaler Daten auf ein niedrigdimensionales Gitter. Dadurch können die Daten unter Einbehaltung der Entfernungsbeziehungen im hochdimensionalen Raum visualisiert werden (Vesanto, 1999).

Die SOM besteht aus einem Satz von Prototyp- oder Gewichtsvektoren $\mathbf{w}_i \in \mathbb{R}^n$, welche mit den Knoten eines regulären niedrigdimensionalen Gitter korrespondieren. Nach der Initialisierung mit Zufallswerten werden die Prototypen iterativ mit der folgenden Lernregel trainiert:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) \cdot h_{c(\mathbf{x}),i}(t) \cdot (\mathbf{x}(t) - \mathbf{w}_i(t)), \tag{10.1}$$

mit t als Index der einzelnen Iterationsschritte. $c(x)$ steht für den Prototypvektoren mit der geringsten euklidischen Distanz zu dem zufällig ausgewählten Trainingsdatensatz (hier z.B. einem Bildausschnitt der Hand) $\mathbf{x}(t) \in \mathbb{R}^n$. Die Adaption wird durch die Nachbarschaftsfunktion $h_{c(\mathbf{x}),i}(t)$ und die Lernratenfunktion $\alpha(t)$ kontrolliert. Beide Funktionen fallen monoton mit der Zeit t. Normalerweise wird eine Gaußsche Nachbarschaftsfunktion verwendet:

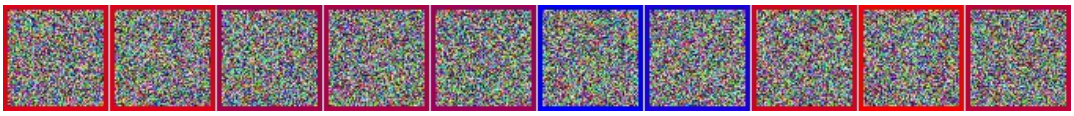
$$h_{c(\mathbf{x}),i}(t) = \exp\left(-\frac{||\mathbf{r}_i - \mathbf{r}_{c(\mathbf{x})}||^2}{2\sigma^2(t)}\right), \tag{10.2}$$

mit $\mathbf{r}_i$ und $\mathbf{r}_{c(x)}$ für die korrespondierenden Gitterknoten. Die Standardabweichung der Nachbarschaftsfunktion wird durch eine monoton fallende Funktion $\sigma(t)$ variiert. Im Gegensatz zu anderen Vektorquantisierungsalgorithmen erzwingt die Nachbarschaftsfunktion die Erhaltung der Topologie. Daher sind die SOMs besonders gut für die Partitionierung und Visualisierung von Bilddaten für die beiden vorgestellten Labelverfahren geeignet.

10.2.2 Komfortable Gewinnung der Datenbasis zur Zeigegestenerkennung

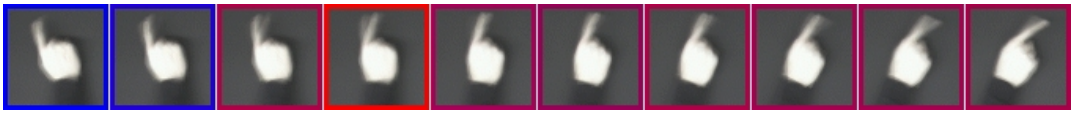
Um den Klassifikator benutzerunabhängig zu trainieren, wurden für die Gewinnung der Datenbasis zum Trainieren des Klassifikators Zeigegesten und andere Handposturen von drei Personen aufgenommen. Diese sollten auf zufällig auf den Tisch verteilte Objekte zeigen und für die Rückweisungsklasse beliebige andere Gesten durchführen. Dabei wurde 3000 Bildauschnitte der Hand aufgenommen. Die Beleuchtungsbedingungen wurden nicht variiert, so dass die Hautfarbensegmentierung zuverlässig war. Da die aufgenommenen Handgesten in einer zweidimensionalen Ebene liegen und in eine Richtung zeigen, wurde eine eindimensionale SOM verwendet.

Wie Abb. 10.3 zeigt, ermöglicht eine 10×1-SOM eine ausreichende Präzision der Zeigerichtung. In der Abbildung werden beide Visualisierungsmöglichkeiten von VALT gezeigt.



(a) Gewichtsvektoren einer zufallsinitialisierten 10 × 1-SOM

(b) Trainingsbeispiele mit der größten Übereinstimmung mit den zufällig initialisierten Prototypen der SOM



(c) Gewichtsvektoren der SOM nach 10000 Trainingsschritten

(d) Trainingsbeispiele mit der größten Übereinstimmung mit den Prototypen der trainierten SOM

Abbildung 10.3: Benutzung einer 10 × 1-SOM zur Strukturierung von Bilddaten:a) und b) vor dem Training mit zufällig initialisierten Gewichten.c) und d) nach 10000 Trainingsschritten repräsentieren benachbarte Knoten ähnliche Zeigerichtungen. a) und c) zeigen die Gewichtsvektoren, b) und d) die Trainingsbeispiele mit der größten Übereinstimmung mit dem Gewicht jedes Knotens.

Die oberen zwei Reihen zeigen den zufallsgenerierten Anfangszustand. Abb. 10.3 c) zeigt die zu den Knoten korrespondierenden Prototypen und d) die dem Prototypen nächsten Traingsbeispiele nach dem erfolgten Training. Am Anfang erkennt man noch keinerlei Ordnung bei den abgebildeten Traingsbeispielen. Nach 10.000 Trainingsschritten mit einer exponentiell abnehmenden Lernschrittweite von 0.5 zu 0.01 sind die Prototypen und die dazugehörigen *best-match*-Beispiele auf Basis der Zeigerichtung angeordnet. Mit VALT können dann die auf den Prototypen abgebildeten Trainingsbeispiele einer Zeigerichtung sehr komfortabel zugeordnet werden. Pro Person benötigt dadurch die Auf-

nahme der Daten nur ca. 10 Minuten. Das Training der SOM dauerte weniger als eine Minute auf einem Standard-PC und konvergiert mit hoher Zuverlässigkeit zu einer SOM-Konfiguration, wie in Abb. 10.3. Das Zuordnen der Klassen durch eine Zeigerichtung über die Prototypvektoren dauerte weitere fünf Minuten.

Die Ergebnisse zeigen, dass das Verfahren zur Erlangung gelabelter Bilddaten für diese Aufgabe sehr gut geeignet ist und im Vergleich zum Labeln von Bilddaten per Hand viel Zeit spart. Dies war die Motivation, für die Gewinnung von gelabelten Objektbilddaten auf dem mobilen System ebenfalls SOMs als Hilfsmittel zur Vorstrukturierung der Daten zu verwenden.

10.3 Iteratives Labeln mit dem mobilen System

Das vorher vorgestellte Verfahren basierte auf einer einfach zu erlangenden Bildmenge, und der Prozess des Labelns wurde mit einer grafischen Benutzeroberfläche an einem normalen PC mit Tastatur und Maus absolviert. Im Gegensatz dazu ist die Portierung einer vergleichbaren Funktionsweise auf ein mobiles System deutlich aufwändiger. Sowohl die Aufnahme der Bilddaten als auch der Prozess des halbautomatischen Labelns unterscheiden sich wesentlich. Die begrenzte Rechnerkapazität bedingt dabei die Notwendigkeit, die Daten unter Erhaltung der Varianz zu komprimieren. Im Folgenden wird anhand der Struktur des Untermenüs für das Iterative Labeln die Realisierung der einzelnen Funktionen vorgestellt. Sowohl die Art der Interaktion als auch die Verwendung der Merkmalskombinationen auf Basis der MPEG-7 Deskriptoren wird zunächst beschrieben und anschließend evaluiert.

Aufbau des Untermenüs zum iterativen Labeln

In Kap. 9 wurde bereits das Hauptmenü des Systems vorgestellt. An dieser Stelle soll die Funktionsweise des iterativen Labelns detailliert anhand des SOM-Untermenüs beschrieben werden, welches separat in Abb. 10.4 gezeigt wird. Das SOM-Menü besteht aus zwei Menüpunkten, die wiederum jeweils in ein Untermenü führen. Für die Bildaufnahme zur Erstellung der Datenbasis wählt der Benutzer den Menüpunkt *Make Exset*, für die Vorstrukturierung der Daten mit Hilfe einer SOM muss *Apply SOM* gewählt werden.

10.4 Bildaufnahme

Durch *Make ExSet* (engl. für *example set*) gelangt man in das Aufnahme-Untermenü. Die Aufnahme von Bildausschnitten wird über das in Kap. 5 vorgestellte Aufmerksamkeitsmodul kontrolliert. Dieses liefert Fokuspunkte als Zentren auffälliger Bildregionen, die vermeintlich Objekte bzw. Objektteile enthalten (Abb. 10.5). Befindet man sich in diesem Menü, werden die stabilen Fokuspunkte mit einem Kreuz visualisiert. Dies ermöglicht dem Benutzer eine erste Einschätzung, ob aktuell ermittelte Bildausschnitte eine sinnvolle Datenbasis für den Objekterkenner darstellen und somit für die Gewinnung der Trainingsmenge geeignet sind. Nur verlässlich vom Aufmerksamkeitsmodul ermittelte

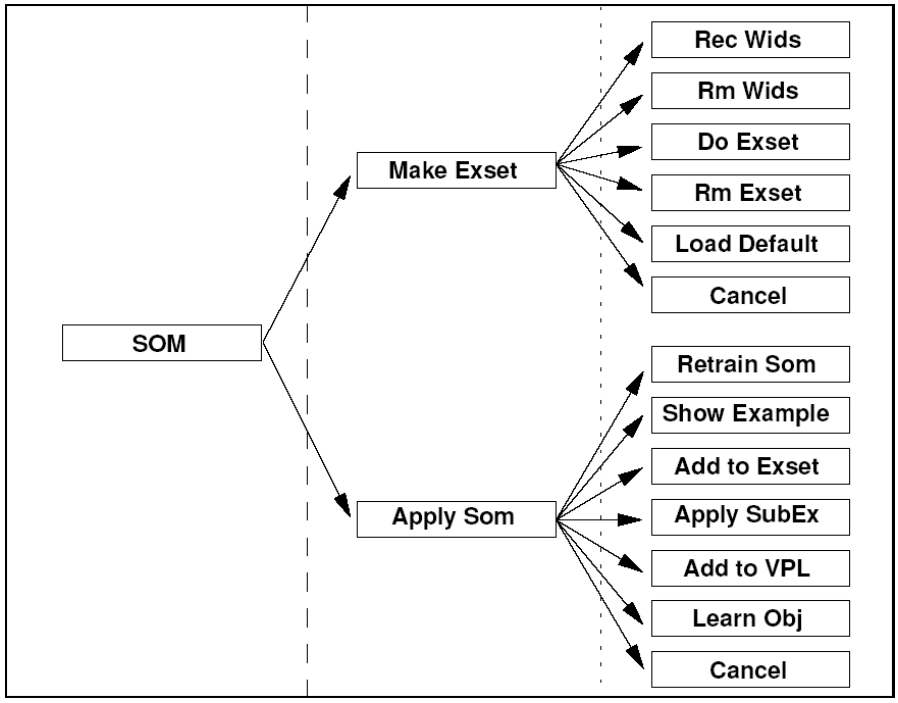


Abbildung 10.4: SomMenu

Bildausschnitte eignen sich für das Training des Klassifikators, da dieser auf der gleichen Grundlage im laufenden System die Bildausschnitte zur Klassifikation geliefert bekommt.

Bei der Aufnahme von Bildausschnitten müssen zwei Faktoren berücksichtigt werden:

1. Fokuspunkte, welche durch schnell wechselnde Bedingungen, wie z.B. durch die Bewegung des Benutzers, hervorgerufen werden und nur in einzelnen Frames ermittelt wurden, eignen sich nicht. Entweder handelt es sich nicht um stabile Fokuspunkte, oder aber der Bildausschnitt ist durch die Bewegung verschwommen.

2. Bewegt sich der Benutzer nicht, sind die Fokuspunkte stabil. Wenn das System um die Fokuspunkte Bildausschnitte aufnimmt, führt das zu anhaltenden Aufnahmen von annähernd gleichen Ansichten von Objekten, wodurch die Trainingsdatenmenge durch eine geringe Varianz ungeeignet wird.

Um die Störungen durch die Bewegung des Benutzers zu umgehen, wurde ein Algorithmus entwickelt, welcher die Bewegung aufgrund der Veränderung der Fokuspunkte erkennt und die Bildaufnahme in diesem Fall unterdrückt. Der Algorithmus berechnet den Ort von über die Zeit korrespondierenden Fokuspunkten FP im Verlauf der letzten Frames. Ein Fokuspunkt F_i^t korrespondiert mit F_j^{t-1}, wenn gilt: $F_j^{t-1} = min_k d(F_i^t, F_k^{t-1}) < t_1$, mit der Distanz in der Bildebene $d(\cdot,\cdot)$ und einem Schwellwert t_1 für die maximal für die Korrespondenz zulässige Distanz. Um durch Störungen hervorgerufene instabile FPs nicht zu berücksichtigen, gehen in die Bewegungsberechnung nur FPs ein, die über

mindestens drei Frames stabil im Sinne der Korrespondenz sind. Da die Bildqualität leidet, wenn die Bewegung zu stark ist, wird die Bildaufnahme gestoppt. Die Bewegung wird als zu stark angenommen, wenn der Schwellwert für die Stärke der Bewegung t_2 wie folgt überschritten wird:

$$\frac{1}{n}\sum_{j=1}^{n}|\frac{1}{\Delta t}\sum_{i=1}^{\Delta t}F_j^t - F_j^{t-i}| < t_2, \tag{10.3}$$

wobei n die Anzahl der ermittelten stabilen Fokuspunkte F_j ist. Wenn die Bewegung abnimmt, werden die Bildausschnitte **einmalig** aufgenommen. Dadurch wird der 2. Faktor berücksichtigt und Aufnahmen gleicher Ansichten von Objekten in statischen Szenen vermieden. Somit werden jeweils die Bilddaten für das Training des Klassifikators nach jeder Bewegung aufgenommen, und erst nach Abschluss der nächsten Bewegung werden erneut Bildausschnitte abgespeichert.

Diese Aufnahmefunktion wird aktiviert, indem *Rec Wids* gedrückt wird. Solange dieser Menüpunkte aktiv ist, wird aufgenommen. Wird die Taste erneut gedrückt, wird die Aufnahme beendet. Falls die Aufnahme fehlerhaft war, z.B. weil die Taste versehentlich aktiviert wurde, können die soeben aufgenommenen Bilddaten mit *Rm Wids* wieder gelöscht werden. Um einen Datensatz für das Labeln zusammenzustellen, können mehrfach Sequenzen aufgezeichnet und ggf. wieder gelöscht werden. Um diese Daten zu labeln müssen die gespeicherten Daten in einen Datensatz für das Training der SOM überführt werden. Dies geschieht, indem der Menüpunkt *Do Exset* gewählt wird. Dazu werden die ausgewählten Bildausschnitte mit den dazugehörigen MPEG-7-Merkmalsvektoren in einen Trainingsdatensatz überführt. Auch hier gibt es die Möglichkeit, die Auswahl wieder mit *Rm Exset* zu löschen. Für Demonstrationszwecke wird bei der Auswahl von *Load Default* ein Beispieldatensatz geladen. Über *Cancel* verlässt man dieses Untermenü und gelangt in das SOM-Menü.

10.5 MPEG-7 – visuelle Bildmerkmale

Für die Datenkompression unter Einbehaltung der Varianz wurden Merkmale des MPEG-7 Standard verwendet. Dieser Standard definiert Deskriptoren, welche Eigenschaften aus medialen Inhalten extrahieren. Entwickelt wurde MPEG-7 im Gegensatz zu den vorherigen MPEG-Standards nicht als Kompressionsstandard, vielmehr standardisiert MPEG-7 die Merkmalsbeschreibung und nicht die Methodik zur Erlangung der Deskriptoren. Unter anderem wurden Deskriptoren für die Beschreibung visueller Medien zur Bildanalyse und -komprimierung entwickelt. Die grundlegenden Deskriporen sind: *Color Layout, Color Structure, Dominant Color, Scalable Color, Edge Histogram, Homogenous Texture, Texture Browsing, Region–based Shape, Contour–based Shape, Camera Motion, Paramtric Motion and Motion Activity.* Die verschiedenen Deskriptoren haben sich bereits in der inhaltsbasierten Bildersuche etabliert(Manjunath u. a., 2001).

Eine Studie von Eidenberger zeigte, dass viele der extrahierten Eigenschaften in verschiedenen Deskriptoren redundant auftreten (Eidenberger, 2003, 2004). Er zeigte, dass

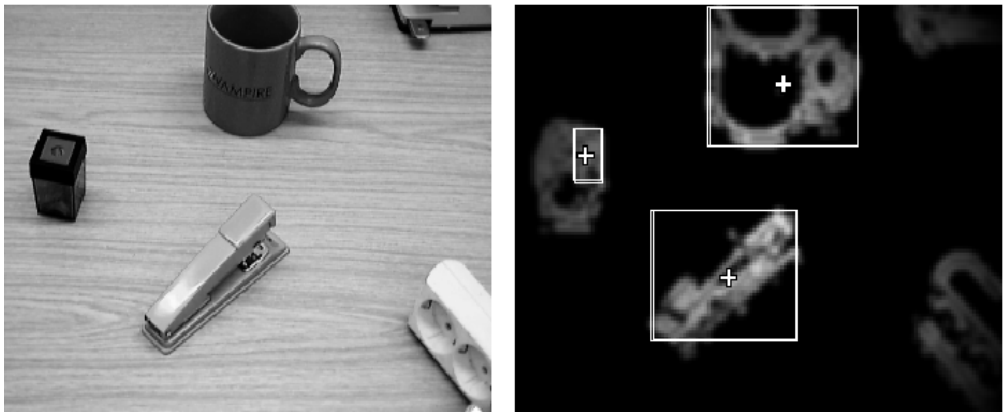


Abbildung 10.5: Als Basis für das iterative Labeln dienen die vom Aufmerksamkeitsmodul gelieferten Bildausschnitte um ermittelte Fokuspunkte. Links: Bildausschnitt der Kamera. Rechts: Ergebnis des Aufmerksamkeitsmoduls. Die weißen Ramen umgeben die errechneten ROIs. Das Kreuz markiert den Schwerpunkt der ermittelten Region als Fokuspunkt, um den Bildausschnitte einer bestimmten Größe abgespeichert werden.

eine Kombination weniger Deskriptoren ausreicht, den Großteil der Varianz in den Daten abzubilden. Basierend auf dieser Analyse wird in der vorliegenden Arbeit eine Kombination aus drei zum Teil abgewandelten Bildmerkmalen verwendet, um trotz der eingeschränkten Rechnerresourcen des mobilen Systems so viele diskriminative Bildeigenschaften wie möglich zu erhalten.

1. Der Kantenhistogramm-Deskriptor (im Original *Edge Histogram*) repräsentiert die Verteilung der Kanten. Er deckt einen großen Bereich der Datenvarianz bei allen Arten von Bilddaten ab.

2. Der *Color Layout*-Deskriptor extrahiert die räumliche Verteilung der Farbe.

3. Der *Scalable Color*-Deskriptor entspricht einer Repräsentation eines HSV-Farbhistogramms und hat sehr gute deskriminative Eigenschaften auf dem Corel-Datensatz gezeigt und eignet sich insbesondere für künstliche Objekte.

Im Folgenden werden die Deskriptoren beschrieben. Detaillierte Beschreibungen finden sich in Manjunath u. a. (2001).

Die Vollständigkeitsanalyse in der Arbeit von Eidenberger offenbarte, dass der Kantenhistogramm-Deskriptor und der *Scalable Color*-Deskriptor Lücken in ihren Abbildungsspektren haben und empfiehlt vor allem für den Kantenhistogramm-Deskriptor eine feinere Auflösung des Operators. In dieser Arbeit wurde er durch eine Änderung der Originalversion optimiert.

Number of Blocks	128	256	512	1024	2048
Edge Length of Block	12	8	6	4	2

Tabelle 10.1: Blockgrößen und korrespondierende Kantenlängen der einzelnen Blöcke für die Berechnung des Kantenhistogramms auf dem Testdatensatz COIL 20.

10.5.1 Kanten Histogramm-Deskriptor

Die räumliche Verteilung von Bildkanten wird mit dem Kantenhistogramm-Deskriptor erfasst. Zuerst werden dafür die Bildausschnitte in 4×4 Unterabschnitte eingeteilt. Der Standard MPEG-7 Deskriptor berechnet für jedes dieser Teilbilder ein Kantenhistogramm. Dazu werden die Kanten in fünf prototypische Richtungen *horizontal, vertikal*, 45°, 135° und *ungerichtet* unterteilt. Die Auflösung des Histogramms wird durch die wählbare Anzahl an Blöcken festgelegt, in die jedes Teilbild zur Kantenberechnung aufgeteilt wird. Für die Berechnung werden diese Blöcke als 2×2 Pixel Bilder betrachtet. Die Grauwerte dieser Vier-Pixel-Bilder resultieren aus den Mittelwerten der jeweils zu den vier Pixeln korrespondierenden Grauwerten. Die Kantenstärke jedes Blockes wird durch die Faltung mit den 2×2-Filtern für jede Richtung berechnet (für Details siehe Manjunath u. a. (2001)).

Im Standard MPEG-7 Deskriptor wird nur die dominante Kantenausrichtung jedes Blockes betrachtet. Wenn dieser einen bestimmten Schwellwert überschreitet, wird das als Eintrag zu dem korrespondierenden Histogrammwert gezählt. Das Verhalten dieses Deskriptors hängt von der gewählten Anzahl der Blöcke ab. Tab. 10.1 gibt die Anzahl der Bildpixel pro Block abhängig von diesem Parameter für Bilder der Größe 128×128 (vorherrschendes Bildformat für die weitere Evaluation) wieder. Da im Standard-Deskriptor nur der vorherrschende Kantentyp zu dem Histogramm beiträgt und die Ergebnisse der anderen Filteroperationen für die anderen Kantenrichtungen nicht weiter betrachtet werden, wurde nachfolgend ein abgewandelter Deskriptor verwendet. Um die Information der anderen Faltungen nicht zu verwerfen, werden hier alle fünf Ergebnisse als kontinuierliche Werte für die lokale Kantenstärke betrachtet, so dass ein kontinuierlicher Wert für alle fünf Richtungen pro Teilbild erhalten bleibt.

In beiden Fällen resultieren aus den 16 Teilbildern mit jeweils fünf Richtungseinträgen 80 Werte. Zusätzlich wird die Performanz dieses Deskriptors erhöht, indem für jede Richtung ein globaler, gemittelter Wert (wie von Park u. a. (2000) vorgeschlagen) an den Merkmalsvektor angehängt wird, so dass dieses Merkmal als 85-dimensionaler Vektor repräsentiert wird. Die Performanz für beide Deskriptoren in Abhängigkeit zur Anzahl der Blöcke wird in Abschnitt 10.7.2 verglichen.

10.5.2 Color Layout-Deskriptor

Das auffälligste visuelle Merkmal ist die Farbe. Um die räumliche Farbverteilung von Bildern zu berücksichtigen, wurde der kompakte *Color Layout*-Deskriptor verwendet, welcher die repräsentativen Farben auf einem Gitter bestimmt und anschließend mit

einer diskreten Cosinus-Transformation die Koeffizienten berechnet.

Dazu werden die RGB-Werte zuerst in den YCrCb-Farbraum transformiert. Anschließend wird das Bild in 64 Blöcke auf einem 8×8-Gitter unterteilt. Für jeden Block wird der mittlere Farbwert errechnet. Hierdurch wird eine gewisse Skalierungsinvarianz erreicht. Auf diesem 8×8-Gitter wird nun die diskrete Cosinus-Transformation berechnet. Für den hier verwendeten, kombinierten Merkmalsvektor werden die niedrigfrequenten Koeffizienten durch das sogenannte Zig-Zack-Scanning selektiert.

Wie von Manjunath u. a. (2001) vorgeschlagen, werden hier sechs Y-, drei Cr- und drei Cb-Koeffizienten verwendet. Neben der Farbe zeigte Eidenberger (2004), dass sich der *Color Layout*-Deskriptor ebenfalls sehr gut für die Beschreibung globaler Umrisse eignet. Er zeigte, dass eine Komponente des Color Layout Deskriptors bereits den Großteil des gesamten *Region Based Shape*-Deskriptors widerspiegelt.

10.5.3 Scalable Color-Deskriptor

Der *Scalable Color*-Deskriptor extrahiert globale Farbmerkmale. Er basiert auf einem Histogramm im HSV-Farbraum und einer hocheffizienten anschließenden Haartransformation, um das Histogramm skalierbar zu machen.

Im Folgenden werden nur die wesentlichen Schritte dargestellt (Details siehe Manjunath u. a. (2001)). Im ersten Schritt werden die RGB-Bildausschnitte in den *HSV*-Farbraum transformiert (Akronym für *Hue Saturation Value*). Der *HSV*-Farbraum wird in 256 Elemente quantisiert, welche die Häufigkeiten der Farben darstellen und aus 16 Stufen für H, 4 Stufen für S und 4 Stufen für V bestehen. Die Quantisierung erfolgt nicht linear zu einem 11 bit-Wert (Manjunath u. a., 2002). Höhere Signifikanz für kleinere Werte höherer Wahrscheinlichkeit wird durch eine nichtlineare Quantisierung zu 4-bit Werten vor der Haartransformation erreicht. Schließlich werden durch die Haartransformation die Hochpasskoeffizienten durch eine Differenzoperation der benachbarten Bins und der Tiefpasskoeffizienten durch die Summationsoperation benachbarter Bins bestimmt.

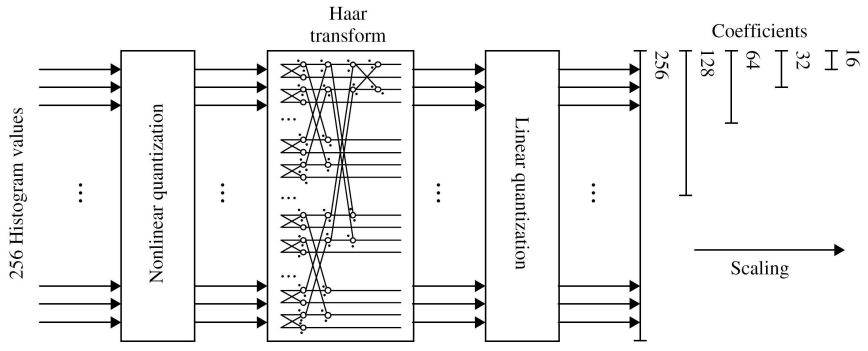


Abbildung 10.6: Ablauf der Extraktion des Scalable Color Deskriptors aus Manjunath u. a. (2002)

Betrachtet man das Histogramm als 16×16 Matrix, wird die Halbierung der Hi-

stogrammeinträge erreicht, indem Paare benachbarter Histogrammzellen aufsummiert werden. Wird dieser Schritt iterativ durchgeführt, erhält man so aus der Haartransformation Histogramme mit 128, 64, 32, ... Elementen. Die Skalierung wird somit durch die Anzahl der Iterationen dieses Schrittes bestimmt. Abbildung 10.5.3 zeigt schematisch die Generierung des Deskriptors.

Die Hochpasskoeffizienten repräsentieren die Informationen in einer feineren Auflösung. Da der Erwartungswert hoch ist, dass benachbarte Farben nur leicht variieren, ist die Differenz der benachbarten Elemente niedrig, so dass sie auf 8-bit Werte durch lineare Quantisierung abgebildet werden können. Der Merkmalsvektor beinhaltet diese Integer-Werte der gewünschten Anzahl an Quantisierungsstufen. In dieser Arbeit wird dieses Merkmal durch eine Quantisierung auf 64 Stufen repräsentiert.

10.6 Ablauf des iterativen Labelns

Wenn eine größere Menge Bildausschnitte über das *Make Exset*-Untermenü, wie in Kap. 10.4 beschrieben, aufgenommen wurde, erfolgt automatisch bei der Überführung in das *Exset* die Extraktion der MPEG-7 Bildmerkmale für jeden Ausschnitt. Die einzelnen Bildausschnitte, welche in einer Büroumgebung aufgenommen werden, können sehr unterschiedliche Strukturen und Farbverläufe enthalten, wie z.B. Bildausschnitte von Teilen von Bücherregalen oder Postern an der Wand, von Tischecken oder aber von einzelnen Objekte auf dem Schreibtisch vor dem Benutzer. Um alle irrelevanten Bildausschnitte zu verwerfen, wird in einem ersten Schritt mit Hilfe einer SOM eine Vorselektion der Bildausschnitte gemacht, bevor dann durch das iterative Labeln die Bildausschnitte für die spätere Klassifikation Label erhalten. Das Labeln der Daten erfolgt iterativ durch die Projektion der Daten auf SOMs, ihrer Visualisierung und dem anschließenden Zuordnen der Objektnamen. Der Merkmalsvektor für das Training der SOM besteht aus der Zusammensetzung der einzelnen Merkmalsvektoren der MPEG-7-Deskriptoren, wie in Kap. 10.5, beschrieben. Dabei werden die einzelnen Komponenten für Farbe und für die Kanten unterschiedlich gewichtet, so dass der Benutzer das Training der SOM auf die Domäne abstimmen kann. Der Merkmalsvektor $\mathbf{x}$ ist somit eine Konkatenation der gewichteten Merkmalskomponenten:

$$\mathbf{x}^{\star} = (w_e * x_1, \ldots, w_e * x_{n_e}, w_c * x_{n_e+1}, \ldots, w_c * x_n), \tag{10.4}$$

mit einem gemeinsamen Gewicht w_e für den n_e-dimensionalen Kantenhistogramm-Merkmalsvektor und einem Gewicht w_c für den $(n - n_e)$-dimensionalen Block der Farbmerkmale. Für den ersten Schritt werden die Gewichte w_e und w_c so berechnet, dass die einzelnen Blöcke über die Varianz normiert sind.

Wählt man aus dem SOM-Menü den Punkt *Apply SOM*, wird eine zweidimensionale Standard-SOM, wie in Kap. 10.2.1 beschrieben, auf den Vektoren $\mathbf{x}^{\star}$ für jeden Bildausschnitt trainiert. Für die Visualisierung der SOM müssen nicht nur die Merkmalsvektoren, sondern auch die dazu korrespondierenden originalen Bildausschnitte abgespeichert werden. Die SOM wird anschließend automatisch im Display visualisiert, indem für jeden Knoten der am besten passende Bildausschnitt dargestellt wird. Im rechten Teil des

Displays erscheint das entsprechende Untermenü für das iterative Labeln. Abbildung 10.7, oben, zeigt ein Beispiel für eine eingeblendete SOM auf Basis von Bildausschnitten des Büroszenarios. Die maximal mögliche Knotenanzahl ist durch die Darstellung des Displays begrenzt. Bei einer maximalen Knotenanzahl von 8×8 lassen sich die Bildausschnitte im Display noch gut erkennen. Die Indizierung der Knoten beginnt unten links bei dem Knoten (0,0) und endet bei der (7,7).

Um zu sehen, wie gut die Vorstrukturierung der Bildausschnitte mit Hilfe der SOM funktioniert hat und zur nachfolgenden Bearbeitung beim Labeln, kann man sich die einzelnen Bildausschnitte pro Knoten visualisieren lassen, in dem der Menüpunkt *Show Examples* gewählt wird. In der Abb. 10.7, unten, sieht man die auf den ausgewählten Knoten (7,7) abgebildeten Bildausschnitte. Man erkennt, dass auf diesem Knoten relativ kompakte Objekte, welche auf dem holzfarbenen Schreibtisch liegen, projiziert worden sind. Die Auswahl des Knotens kann entweder mit einem Sprachkommando wie „Zeige mir Knoten 7 7" getroffen werden oder dadurch, dass die jeweilige Zeile und Spalte per Fingerspitze gewählt wird. Die Art der Selektion mit der Fingerspitze erfolgt wie bei der Menüführung.

Je nachdem wie die Bildausschnitte, deren korrespondierende Merkmalsvektoren auf einen Knoten abgebildet worden sind, aussehen, hat der Benutzer drei Möglichkeiten:

(1) Der Benutzer verwirft Bildausschnitte, wenn sie irrelevant oder aufgrund von schlechter Aufnahmequalität verschwommen und daher nicht zur Klassifikation geeignet sind.

(2) Wenn alle Bildausschnitte eines Knotens das gleiche Objekt, möglichst aus verschiedenen Ansichten oder unter unterschiedlichen Bedingungen, enthalten, kann der Benutzer den Bildausschnitten unmittelbar eine Bezeichnung zuweisen und als Teil des Trainingsdatensatzes für den Klassifikator abspeichern.

(3) Wenn, wie in der Abb. 10.7, unten, verschiedene Objekte auf einen Knoten projiziert wurden, kann der Benutzer diesen Bildausschnitten eine Gruppen-ID zuweisen. Wenn die benachbarten Knoten eine Mischung derselben Objekte enthalten, bekommen sie dieselbe Gruppen-ID.

Der Labelprozess für die Bildausschnitte aus (1) und (2) ist somit abgeschlossen. Für die restlichen Bildausschnitte des 3. Falls kann erneut eine SOM trainiert werden, so dass der Prozess des Labelns in einer Schleife wiederholt wird. Werden bei großen Datenmengen für die Bildausschnitte verschiedener Knoten Gruppen-IDs vergeben, kann jeweils pro Gruppe eine SOM auf den Daten trainiert werden. Dieser iterative Prozess wird solange durchgeführt, bis alle Objekte gelabelt sind. Die folgenden Evaluationskapitel werden zeigen, dass meist 2-3 Schritte genügen, bis eine vollständige Trennung der Objektbilder und somit eine korrekte Zuweisung der Namen erfolgt. Die Effizienz kann dabei gesteigert werden, indem der Benutzer aufgrund der Bildausschnitte entscheidet, ob entweder die Farb- oder Kantenmerkmale diskriminativer sind, und diese hebt der Benutzer durch Veränderung der relativen Merkmalsgewichte w_e/w_c hervor. Anschließend kann auf dem gelabelten Datensatz der Klassifikator trainiert werden.

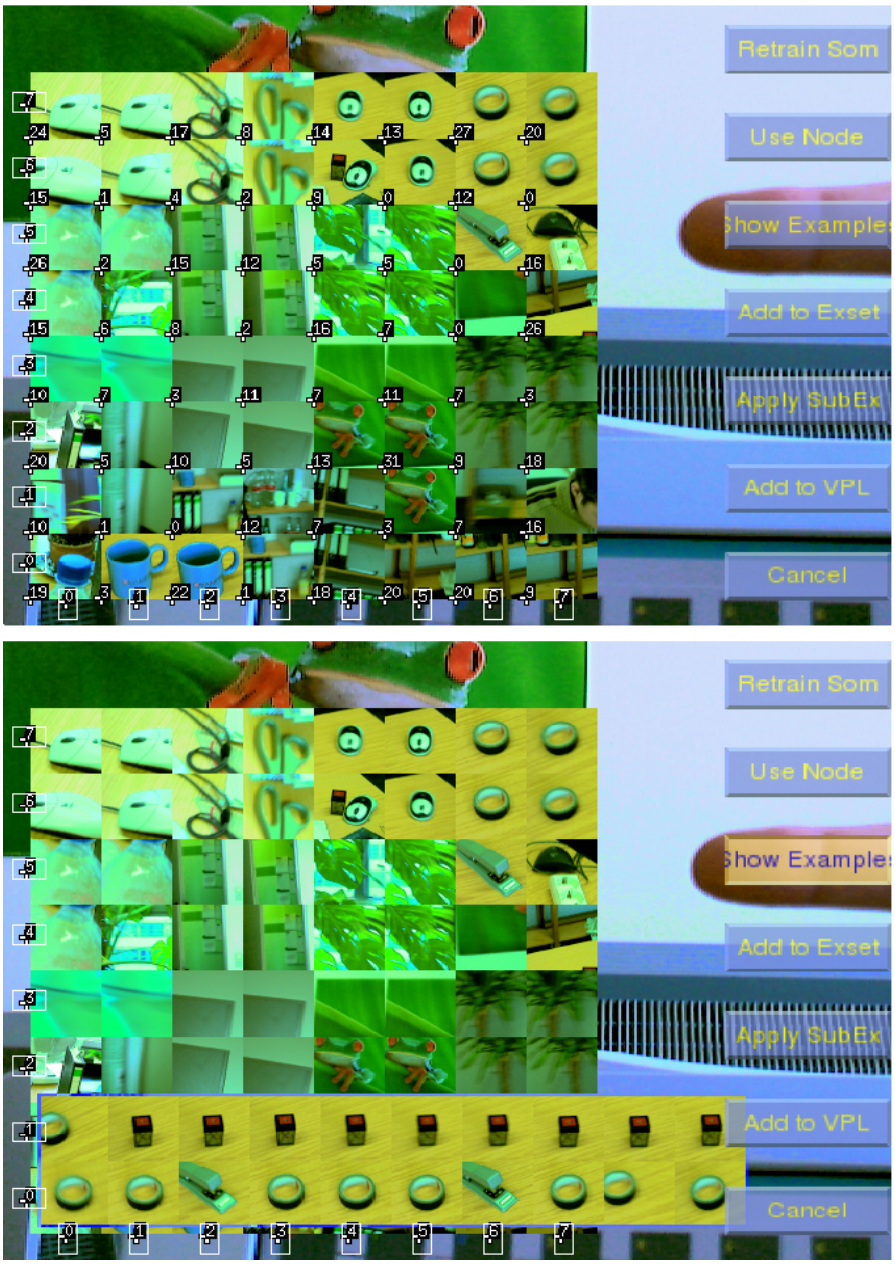


Abbildung 10.7: Oben: Eingeblendete SOM, welche mit 15000 Schritten und gleicher Gewichtung der Kanten- und Farbmerkmale trainiert wurde. Die abgebildeten Bildausschnitte korrespondieren zu den Merkmalsvektoren, die den Prototypen der Knoten am nächsten sind. Die Zahlen unten links in den Bildausschnitten geben die Anzahl der projizierten Vektoren pro Knoten an. Im Hintergrund sieht man das aktuelle Kamerabild mit einem Poster eines Frosches, welches ebenfalls auf einem Bildausschnitt auf der SOM zu sehen ist. Der Benutzer wählt den Menüpunkt *Show Example* mit der Fingerspitze. Unten: Die auf den ausgewählten Knoten (7,7) abgebildeten Bildausschnitte werden eingeblendet.

10.7 Evaluierung auf einem Standarddatensatz

Für die Evaluation der Performanz des Labelprozesses und zur Parametrisierung des Systems wurde eine Standardbilddatenbank verwendet, um reproduzierbare Ergebnisse zu erzielen. Dazu wurden zwei Datensätze auf Basis der COIL-Datenbank (Akronyum für *Columbia University Object Image Library* erstellt (Nene u. a., 1996a). Die COIL 100-Datenbank ist eine etablierte Bilddatenbank, welche bereits für eine Vielzahl von Evaluationen verwendet wurde. Sie besteht aus Bildern von 100 Objekten, welche mit Hilfe eines Drehtellers aus 72 verschiedenen Ansichten aufgenommen wurden. Aus dieser Datenbank wurden die folgenden zwei Testdatensätze erstellt:

> **„Orig":** Eine Teilmenge von 20 Objekten der COIL-100 Datenbank mit jeweils 72 Ansichten pro Objekt. Der Datensatz besteht somit aus 1440 RGB-Bildern im Format 128×128 Pixeln (Beispiele dafür in Abb. 10.10).
>
> **„Distort":** Die Bilder des Datensatzes **Orig** wurden manipuliert, indem die Objekte zufallsgeneriert im Bereich von +/- 5 Pixel in x- und y-Richtung translatiert und um +/- 10% skaliert wurden.

Als erstes wurden die Klassifikationseigenschaften der MPEG-7 Deskriptoren getestet. Anschließend wurden mit Hilfe der SOM die diskriminativen Eigenschaften einer abgeänderten Version des Kantenhistogramm-Deskriptors im Vergleich zum Original analysiert. Im letzten Teil wird die Performanz des gesamten Labelprozesses evaluiert.

10.7.1 Performanz der Merkmale

Die diskriminativen Eigenschaften der MPEG-7-Deskriptoren bzw. von einer Kombination dieser Merkmale zur Klassifikation wurden auf den beiden vorgestellten Datensätzen **Orig** und **Distort** evaluiert. Auf diesen Datensätzen wurden die MPEG-7 Merkmale berechnet, so dass jedes Bild auf einen n_e-dimensionalen Vektor für den *Edge histogram*-Deskriptor, einen n_c-dimensionalen Vektor für den *Color Layout*-Deskriptor und einen n_s-dimensionalen Vektor für den *Scalable Color*-Deskriptor abgebildet wurde. Als einfaches, rekonstruierbares Maß für die diskriminativen Eigenschaften wurde die Klassifikationsrate eines *Nearest neighbor*-Klassifikators ermittelt. Die Datenbasis für den Klassifikator bestand (i) aus den Originalbildern der Testdatensätze, (ii) jeweils aus den zu (i) korrespondierenden drei Merkmalsvektoren und (iii) aus einer Kombination der drei Deskriptoren in Form eines zusammengesetzten $n_e + n_c + n_s$-dimensionalen Merkmalsvektors. Die Klassifikationsrate für die einzelnen Datensätze wurde in Abhängigkeit von der Anzahl der Trainingsansichten bestimmt. Abbildung 10.7.1, links, zeigt die ermittelten Ergebnisse für den **Orig**-Datensatz, rechts für den **Distort**-Datensatz. Die exakten Werte für die ersten zehn Trainingsansichten werden noch einmal in Tabelle 10.2 präsentiert.

Die Ergebnisse zeigen, dass auf den Datensätzen besonders die Farbmerkmale sehr diskriminativ sind und trotz der erheblichen Dimensionsreduktion höhere Klassifikationsraten als die Originalbilder ermöglichen. Der Kantenhistogramm-Deskriptor erreicht

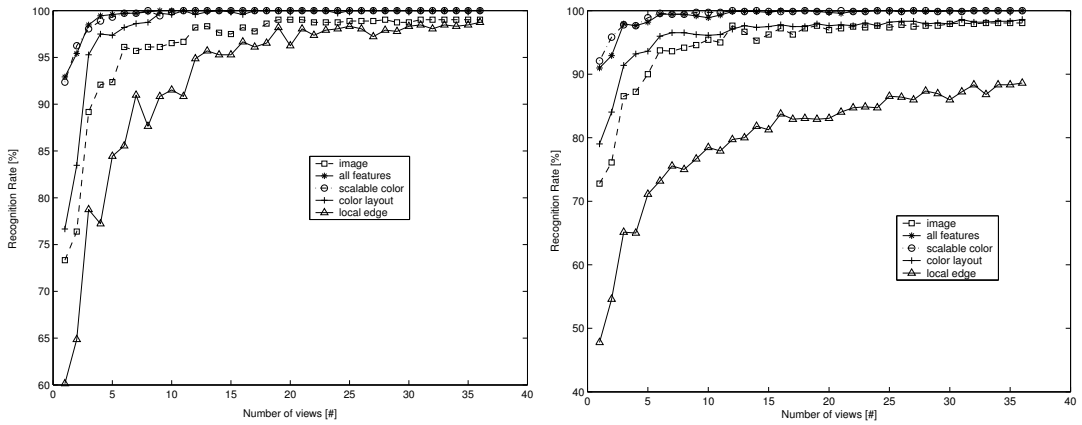


Abbildung 10.8: Verlauf der Klassifikationsrate eines *Nearest Neighbor*-Klassifikator auf den Datensätzen (links) **Orig** und (rechts) **Distort** in Abhängigkeit von der Anzahl der Trainingsansichten.

deutlich niedrigere Raten. Dies liegt vermutlich an der einfachen Datenbasis, da sich die Objekte meist klar in der Farbe unterscheiden und vor schwarzem Hintergrund aufgenommen wurden. Die besten Klassifikationsraten erzielt die Kombination der drei MPEG-7 Deskriptoren. Bereits bei drei Trainingsansichten ermöglicht die Kombination eine Klassifikationsrate von über 98% auf dem **Orig**-Datensatz und ein ähnlich gutes Ergebnis auf dem durch Translation und Skalierung veränderten Datensatz. Bereits auf diesen einfachen Datensätzen konnte sich die Kombination der Merkmale als äußerst effektiv erweisen. Diese Ergebnisse können nur als Indiz für die Diskriminativität auf Bildern in natürlicher Umgebung verstanden werden. Hier wird vermutlich der Kantenhistogramm-Deskriptor eine deutlich größere Rolle spielen. Auf Basis dieser Ergebnisse und der Ergebnisse von Deselaers u. a. (2004) kann in jedem Fall bestätigt werden, dass eine Kombination von MPEG-7 Merkmalen sehr gut geeignet ist, die Varianz von Bilddaten und damit die Diskriminativität trotz erheblicher Datenkomprimierung zu erhalten.

10.7.2 Anpassung des Kantenhistogrammdeskriptors

Bereits aus der Beschreibung des Kantenhistogramm-Deskriptors aus Kap. 10.5.1 geht hervor, dass die Anzahl der Blöcke bei der Berechnung des Deskriptors ein entscheidender Parameter ist. Das ursprüngliche Entwicklungsziel der visuellen MPEG-7-Deskriptoren war der Einsatz für das sogenannte *image retrieval*, also der inhaltsbasierten Suche in Bilddatenbanken. Während in dieser Domäne vor allem mit großen Bildern gearbeitet wird, sollen hier die diskriminativen Eigenschaften von kleinen Bildausschnitten extrahiert werden. Daher sollte das Verhalten des Deskriptors an diese Domäne angepasst werden. Die Performanz des originalen Deskriptors auf hochauflösenden großen Bildern ist ausgezeichnet, jedoch werden die Informationen über Kantenrichtungen, welche nicht die dominierenden Kanten in einem Block sind, nicht für die Histogrammeinträge

	Bild		Kombination		Scalable Color		Color Layout		Local Edge	
#	orig	distort	orig	distort	orig	distort	orig	distort	orig	distort
1	73.3	72.7	92.9	90.9	92.3	92.0	76.6	79.0	60.1	47.7
2	76.3	76.1	95.4	92.9	96.2	95.8	83.4	84.0	64.8	54.5
3	89.1	86.5	98.4	97.9	98.0	97.7	95.2	91.3	78.7	65.1
4	92.0	87.2	99.4	97.6	98.8	97.6	97.5	93.1	77.2	65.0
5	92.3	90.0	99.5	98.1	99.3	98.8	97.3	93.6	84.4	71.1
6	96.1	93.7	99.7	99.4	99.7	99.5	98.1	95.9	85.5	73.1
7	95.6	93.6	99.7	99.4	99.7	99.4	98.6	96.5	90.9	75.5
8	96.1	94.1	99.8	99.4	100.0	99.4	98.7	96.5	87.6	75.0
9	96.1	94.5	100.0	99.1	99.4	99.7	99.7	96.2	90.8	76.6
10	96.5	95.4	100.0	98.8	99.8	99.7	99.5	96.1	91.5	78.4

Tabelle 10.2: Klassifikationsergebnisse eines *Nearest neighbor*-Klassifikators auf verschiedenen COIL-Datensätzen in Abhängigkeit von der Anzahl der Trainingsansichten (aus Bekel u. a. (2005b)).

berücksichtigt. Für die kleinen Bildausschnitte wurde der Deskriptor so abgeändert, dass die Informationen der nicht dominanten Kantenrichtungen mit in die Berechnung eingehen.

Die diskriminativen Eigenschaften der abgeänderten Version wurden mit Hilfe einer SOM mit dem Original verglichen; dabei wurde die optimale Parametrisierung bestimmt. Diese Evaluation erfolgte auf den beiden Datensätzen **Orig** und **Distort** der COIL-Datenbank, auf welchen jeweils die entsprechenden Merkmalsvektoren mit variierenden Kantenlängen der Blöcke für die beiden Deskriptoren berechnet wurden (vgl. Tab. 10.1). Auf diesen Merkmalsvektoren wurde eine SOM mit 8×8 Knoten in 15000 Schritten mit exponentiellem Abfall der Schrittweite α von 0.9 bis 0.1 und σ von 2.0 bis 1.0 trainiert (aus der Gl. 10.2).

Diese Parametrisierung wurde ebenfalls für die weitere Evaluation und für das mobile System beibehalten, da sie geringe Fehlerraten und ein gleichmäßiges „Hineinlegen" des SOM-Gitters im Merkmalsraum erzielt.

Alle Objekte, welche auf einen SOM-Knoten abgebildet wurden, erhielten die gleiche Bezeichnung. Die Bezeichnung für jeden SOM-Knoten ist der Name des Objektes, welches am häufigsten auf diesen Knoten projiziert wurde. Alle anderen Objekte wurden als Fehler gezählt. Die Abb. 10.9 zeigt die gemittelten Fehlerraten für 10 Trainingsläufe auf beiden Datensätzen und für fünf verschiedene Blockgrößen. Bei dem originalen Kantenhistogramm-Deskriptor sinkt die Fehlerrate mit abnehmender Blockgröße. Der modifizierte Deskriptor erreicht insgesamt deutlich geringere Fehlerraten. Bei der Blockgröße von 4×4 Pixeln erreicht er für beide Datensätze ein klares Minimum. Die modifizierte Version erhält somit die diskriminativen Eigenschaften bei kleinen Bildausschnitten besser als das Original. Daher wurde der modifizierte Deskriptor mit dieser Parametrisierung für das mobile System und in der folgenden Evaluation verwendet.

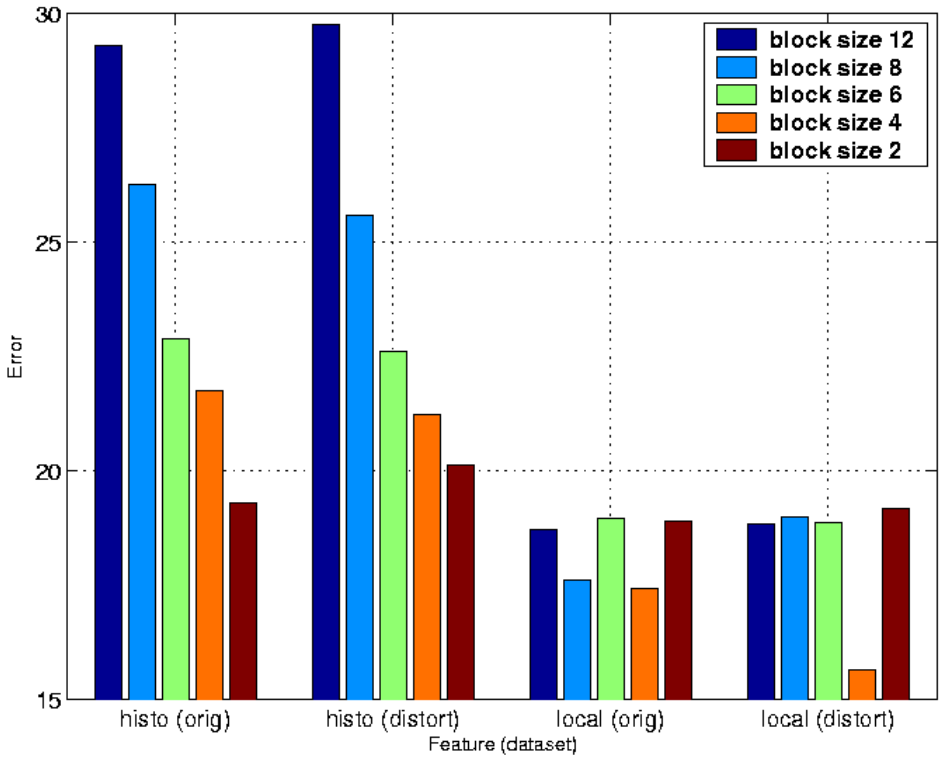


Abbildung 10.9: Fehlerraten des Standard *Edge Histogramm*-Deskriptors (*histo*) im Vergleich mit der geänderten Version (*local*) für die Dantensätze **Orig** und **Distort** der COIL-Datenbank (aus Bekel u. a. (2005a)).

10.7.3 Evaluation der System-Performanz

Um zu beurteilen, ob das vorgeschlagene Verfahren für ein mobiles System geeignet ist, müssen verschiedene Aspekte betrachtet werden. Zum einen sind die Rechnerressourcen begrenzt, so dass für eine komfortable Verwendung des Systems im laufenden Betrieb der Rechenaufwand gering sein muss. Durch die Verwendung der Merkmalskombination aus den Deskriptoren kann diese Bedingung erreicht werden. Das Training einer 8×8–SOM mit 15000 Schritten benötigt auf dem Datensatz **Orig** mit 1440 RGB-Bildern der Dimension 128×128 Pixel im Mittel lediglich 5,3 sec auf einem Standard Pentium IV 2.5 GHz Prozessor.

Ein weiterer zu evaluierender Aspekt ist der Aufwand, den der Benutzer für das Labeln der Daten investieren muss. Als Maß für den Aufwand wurde die Anzahl der Knoten verwendet, die der Benutzer für das Labeln eines Datensatz „anfassen" muss, also die Anzahl der Knoten, auf denen mehr als nur ein Objekt abgebildet wurde, so dass auf diesen Daten nachtrainiert werden muss. Ein weiteres wichtiges Maß ist die Fehlerrate der ersten SOM, welche wie in Kap. 10.7.2 bestimmt wurde. Die Fehlerrate, also die Anzahl der Objekte, welche nicht die Mehrheit pro Knoten darstellen, ist ein Anhaltspunkt

Abbildung 10.10: Projektion der COIL-Testdatensätze auf eine SOM.

für die Beurteilung des weiteren Aufwandes. Knoten mit einer Fehlerrate von nur 1% führen mit hoher Wahrscheinlichkeit schon im nächsten Iterationsschritt zu einer perfekten Trennung. Im Gegensatz dazu sind bei einer Fehlerrate von 49% wahrscheinlich häufiger mehrere Schritte nötig.

Tabelle 10.3 zeigt die Ergebnisse für eine 8×8-SOM: In der ersten Spalte stehen die verwendeten Deskriptoren, *Kante* steht für den *Edge Histogram*-Deskriptor, *Farbe* für den zusammengesetzten Merkmalsvektor aus dem *Scalable Color*- und dem *Color Layout*-Deskriptor und *Kombi* für die Kombination aller drei verwendeten Deskriptoren. Die zweite Spalte gibt den Datensatz an, und zwar entweder den unveränderten COIL-Datensatz **Orig** oder den durch Translation und Skalierung manipulierten **Distort**-Datensatz. Die Ergebnisse nach einmaligem Training sind unter *Erstes Training* aufgelistet, unter *Nachtrainieren* stehen die Ergebnisse nach dem zweiten Iterationsschritt mit jeweils pro Gruppe/Knoten einmaligem Nachtrainieren auf den noch nicht getrennten Bildausschnitten. Die Spalten *Treffer* geben die Anzahl der Knoten an, auf

		Erstes Training			Nachtrainieren	
Merkmal	Datensatz	Treffer	Fehler	Fehlerknoten	Fehler	Treffer
Kombi	Orig	33.5	**1.083**	3.1	**0.069**	13.2
Kombi	Distort	33.9	**1.861**	3.9	**0.083**	19.7
Farbe	Orig	34.0	**2.013**	4.3	**0.333**	18.3
Farbe	Distort	31.7	**2.569**	4.5	**0.243**	19.9
Kante	Orig	56.4	**16.736**	36.8	–	–
Kante	Distort	57.2	**16.125**	39.0	–	–

Tabelle 10.3: Fehlerraten und Anzahl von Fehlerknoten des SOM-Trainings (Details in Abschnitt 10.7.3.)

denen Objekte projiziert wurden, *Fehler* gibt den prozentualen Anteil der Objekte an, welche nicht in der Überzahl pro Knoten sind, und *Fehlerknoten* gibt die Anzahl der Knoten wieder, auf denen mehr als ein Objekt abgebildet wurde. Bereits nach dem ersten Training beträgt die Fehlerquote für den schwierigeren Datensatz unter Verwendung der vorgeschlagenen Merkmalskombination weniger als 2 % und für den Originaldatensatz nur wenig über 1 %. Dabei spielt die Farbe auf dem COIL-Datensatz eine entscheidende Rolle, da der Fehler auf den Farbmerkmalen nur geringfügig höher ist; die Kantenmerkmale kommen hingegen auf Werte von gut 16 %.

Der Aufwand für den Benutzer ist dabei selbst auf einem mobilen System handhabbar, da bei der Verwendung der Merkmalskombination im Durchschnitt je nach Datensatz lediglich auf 3-4 Knoten unterschiedliche Objekte projiziert wurden und somit nur diese das Training einer zweiten SOM in einem weiteren Iterationsschritt notwendig machen. Nach einmaligem Nachtrainieren ist der Fehler bereits im Mittel auf unter 0,1 % gesunken, da sich die bisher auf einen oder wenige Knoten verteilten verschiedenen Objekte bei der zweiten SOM auf 13-19 Knoten verteilen und dabei fast immer eine vollständige Trennung der Daten erfolgt.

Die Abbildung 10.10 zeigt die klare Strukturierung der SOM in Bereiche von jeweils sich ähnelnden Objekten. Die Strukturierung erfolgt sowohl durch die Farbe als auch durch die Form der Objekte. Im oberen linken Bereich der SOM finden sich rote Objekte, direkt darunter braune, wobei für beide Farben im linken Teilbereich kompakte, eher runde Objekte abgebildet sind und im rechten eher längliche. Somit erkennt man, dass die Kombination der Merkmale zu einer sinnvollen Strukturierung führt.

Die Abbildung 10.11 zeigt im Gegensatz dazu eine Auswahl von Objekten, die oft gemeinsam im ersten Iterationsschritt auf einzelne Knoten projiziert wurden. Die Zahlen zeigen die Häufigkeiten in den entsprechenden Konfusionsmatrizen einmal bei der Verwendung der Merkmalskombination aus allen drei Deskriptoren und einmal nur aus den beiden Farb-Deskriptoren an. Die Verwechslungshäufigkeit der ersten beiden Objekte, die Spielzeugkatze und die Tasse, unter Verwendung der Farbmerkmale zeigt, dass hier die diskriminativen Eigenschaften vor allem durch den Kantenhistogramm-Deskriptor kodiert werden. Wenn die gesamte Merkmalskombination verwendet wird, werden diese

zwei Objekte nicht mehr verwechselt. Die beiden Holzbauklötze erweisen sich als die schwierigsten Objekte, da sie sich nicht durch die Farbe unterscheiden und die Form bei beiden je nach Ansicht stark variiert und sich viele dieser Perspektiven einander ähneln, so dass für solche Objekte eine Trennung immer aufwändig bleibt. Dennoch zeigt die rechte Spalte von Tab. 10.3, dass auch diese Objekte bereits im zweiten Schritt komplett getrennt und gelabelt werden können.

Insgesamt kann aus den Ergebnissen geschlossen werden, dass sich die Kombination der Merkmale sehr gut zur Vorstrukturierung von Bilddaten eignet und der Aufwand für das Labeln von Bilddaten mit dem beschriebenen System dabei so gering ist, dass es selbst auf einem mobilen System handhabbar ist. Besonders im zweiten Schritt können die Objekte durch die Integration des Benutzers als Experten in der Interaktionsschleife noch effektiver getrennt werden, indem der Benutzer auf Grund seiner Einschätzung die entsprechenden Gewichte für die Kanten- oder Farbmerkmale erhöht.

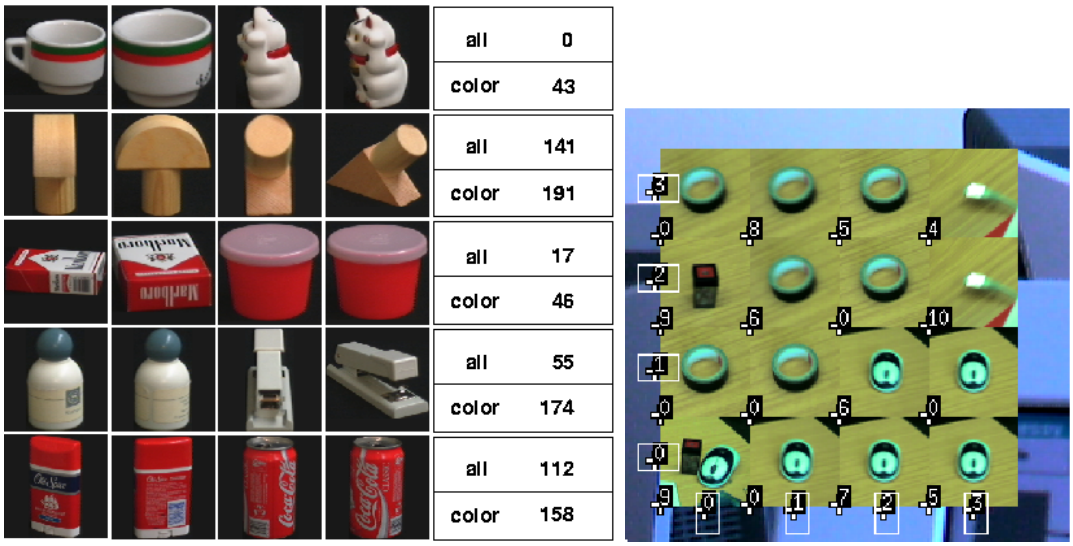


Abbildung 10.11: Links: Objektpaare, welche häufiger gemeinsam auf einzelne SOM-Knoten projiziert wurden. Die Zahlen geben die Werte aus der Konfusionsmatrix an, einmal für die beiden Farbmerkmale *color* und einmal für die Kombination aller drei Deskriptoren *all*. Rechts: Büroobjekte der Knoten mit gleicher ID aus Abb. 10.7, welche bereits im zweiten Nachtrainieren getrennt werden können (siehe 10.7.4).

10.7.4 Beispiele aus einer Büroumgebung

Um die Funktionalität des Systems in einer realen Umgebung zu demonstrieren, wurde im Büroszenario ein Datensatz mit dem in 10.4 beschriebenen Verfahren aufgenommen. Dazu bewegt sich der Benutzer im Büro für einen Zeitraum von ca. fünf Minuten möglichst natürlich und schaut sich dabei in dieser Umgebung um. Nach jeder Bewegung werden dann aus dem ersten Kamerabild in Ruheposition die Bildausschnitte, welche das

Aufmerksamkeitsmodul liefert, aufgenommen. Dabei wurden 670 Bildausschnitte aus 220 Frames aufgenommen, welche den Bewegungsfilter passiert haben.

Die Bildausschnitte, die dabei vom System abgespeichert wurden, beinhalten Objekte, wie Teile des Schreibtisches, Lampen, Pflanzen, Teile der im Büro hängenden Poster etc. Die Bildausschnitte, auf einer SOM nach dem ersten Training projiziert, welche dem Kamerabild überlagert wird, zeigt Abb. 10.7. Das Ergebnis ist eine klar strukturierte SOM auf welcher helle Bildausschnitte oben links und dunkle unten rechts abgebildet sind. Bei Objekten, welche vor dem Benutzer auf dem Schreibtisch liegen, dominiert die Schreibtischfarbe, welche zu einer Region oben rechts führt, in der kompakte Objekte auf dem Schreibtisch abgebildet werden. Die grünen Ausschnitte des Poster befinden sich rechts von der Mitte in direkter Nachbarschaft von grünen Pflanzen. Objekte mit größeren Blauanteilen, wie die VAMPIRE-Projekttasse sind unten links abgebildet. Diese Organisation der Daten auf die SOM zeigt, dass sich die Datenrepräsentation durch die Kombination der MPEG-7-Merkmale auch in dieser heterogenen Umgebung als Datenbasis eignet, um mit Hilfe der SOM eine Vorkategorisierung der Daten zu erreichen. Dadurch ist es möglich, dass in nur wenigen Arbeitsschritten auch hier die Daten gelabelt werden können. Abb. 10.7, unten, zeigt mit einem blauen Rahmen hervorgehoben die Bildausschnitte, welche auf einem einzelnen Knoten (7,7) projiziert wurden. Die benachbarten Knoten zeigen eine ähnliche Kombination von kompakten Objekten welche auf dem Schreibtisch liegen und somit für den Benutzer wichtige Objekte darstellen, die potentiell verlegt werden können. Diese Knoten erhielten die gleiche Gruppen-ID und auf diesen Ausschnitten wurde gemeinsam im zweiten Iterationsschritt eine 4×4-SOM trainiert. Aufgrund der einheitlichen Hintergrundfarbe des Schreibtisches und der Objektfarbe, hat der Benutzer die Kantenmerkmale als diskriminativer eingeschätzt und das entsprechende Gewicht erhöht. Abb. 10.11 zeigt das Ergebnis. Bereits im zweiten Schritt wurden alle Objekte separiert, indem sie auf einzelne Knoten projiziert wurden und so gelabelt werden konnten.

Kapitel 11

Objekterkennung

Das Kernstück einer Brille mit Gedächtnis ist der Objekterkenner. Der verwendete Objekterkenner wurde bereits in Kap. 6 vorgestellt. Die Performanz des ansichtsbasierten Objekterkenners in einem mobilen System in heterogener Umgebung ist immer abhängig von den aktuellen Bedingungen im Vergleich zu den Bedingungen, bei denen der Klassifikator trainiert wurde. Im vorigen Kapitel wurde bereits gezeigt, wie das System benutzergesteuert zur Laufzeit neue Trainingsdaten aufnehmen und labeln kann. Dieses Verfahren ist immer dann notwendig, wenn die Objekte nicht mehr richtig erkannt werden, sei es durch einen Wechsel der Bedingungen in der aktuellen Umgebung oder z.B. durch einen Raumwechsel.

Der Benutzer als Teil der Verarbeitsschleife (*„human in the loop"*) nimmt dabei die Rolle des Experten an, welcher die aktuelle Klassifikationsperformanz ohne Weiteres beurteilen kann. Dazu benötigt er jedoch das Feedback vom System, wo und als was Objekte im laufenden Bildstrom erkannt wurden. Im Hauptmenü kann der Benutzer den zweiten Hauptmenüpunkt mit der Bezeichnung *Object Rec* wählen und erhält so das Feedback über die Objekterkennungsergebnisse des VPL.

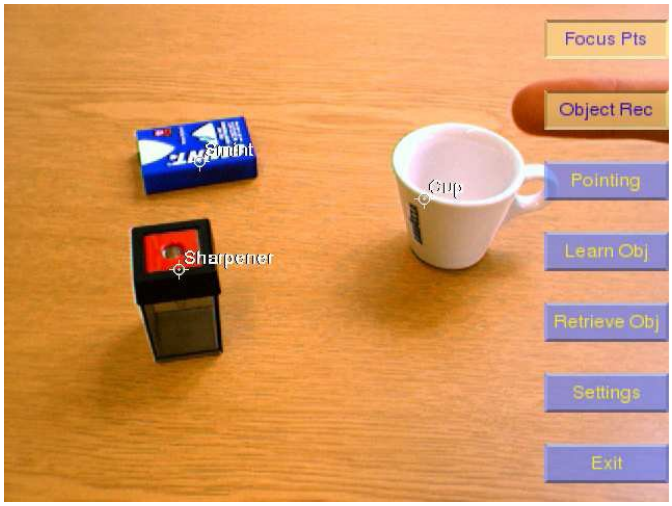


Abbildung 11.1: Eingeblendete Fokuspunkte und Objektlabels.

Der Objekterkenner wertet in jedem Frame eine quadratische Bildregion um die Fokuspunkte, die vom Aufmerksamkeitssystem geliefert werden, mit einer Dimension von $n \times n$-Pixeln aus. (Aufgrund der Auflösung der Kamera wurde hier $n = 41$ gewählt.) Solange dieser Menüpunkt aktiv ist, werden die erkannten Objektlabels, wie in Abb. 11.1 gezeigt, rechts neben den dazugehörigen Fokuspunkten eingeblendet. Diese Label werden pro Frame ausgewertet und aktualisiert. Die Klassifikation pro Frame erfordert auf den Rohbildern einen erheblichen Rechenaufwand, der insbesondere von der Dimension des Klassifikators abhängt. Um aufgrund der begrenzten Rechnerressour-

cen eines mobilen Systems den Rechenaufwand möglichst gering zu halten und um trotzdem eine Klassifikationsperformanz auf hohem Niveau zu erhalten, wurde ein Verfahren entwickelt, welches Berechnungsergebnisse des Aufmerksamkeitssystems nutzt, um die Klassifikation zu beschleunigen. Dieses Verfahren wird im folgenden Abschnitt im Detail erläutert.

Reduktion der visuellen Komplexität

In einer vom Menschen erschaffenen Umgebung, wie z.B. einem Büro, gibt es allein aufgrund dessen, dass der Mensch mit Objekten hantiert, eine Vielzahl von Objekten wie z.B. Stifte, welche eine längliche Ausdehnung aufweisen. Das visuelle Erkennungssystem des Menschen ist weitestgehend rotationsinvariant. D.h. wenn der Mensch einen Stift auf einem Tisch liegend sieht, ist es für ihn verhältnismäßig egal, ob der Stift senkrecht oder waagerecht liegt. Um zu überprüfen, ob zwei auf dem Tisch liegende Stifte gleich sind, rotiert der Mensch die beiden Bilder im Kopf und legt sie übereinander. Für einen ansichtsbasierten Objekterkenner stellt dies jedoch ein Problem dar. Für ihn sind sich zwei waagerecht liegende längliche Objekte, wie z.B. ein Stift und eine Heftklammer, deutlich ähnlicher als zwei Stifte, welche einmal horizontal und einmal vertikal auf dem Tisch liegen, da die meisten ansichtsbasierten Objekterkenner auf der Distanzberechnung im Pixelraum oder auf ortsgetreuen Merkmalen basieren. Bei dem VPL-Klassifikator führt die Verarbeitung der ersten zwei Schritte, also die Partitionierung durch die Vektorquantisierung und die anschließende lokale PCA, zu Prototypvektoren, die mit *rezeptiven Feldern* verglichen werden können, mit denen die Bildausschnitte während des Klassifikationsprozesses gematched werden. Diese rezeptiven Felder sind für längliche Objekte kaum selektiv. Zur Verdeutlichung dieses Aspektes wurde ein VPL mit den Parametern $N_V = 5, N_P = 8$ und $N_L = 20$, im Folgenden als 5-8-20-VPL bezeichnet, mit jeweils 20 Ansichten von fünf verschiedenen länglichen Objekten trainiert. Abbildung 11.2, links, zeigt die ersten acht Hauptkomponenten von den fünf Prototypvektoren. Diese Prototypvektoren differenzieren sich nicht im Hinblick auf die verschiedenen Objekte, sondern sind für einzelne Rotationsrichtungen spezifisch. D.h., die Ausrichtung der länglichen Objekte spielt eine wesentlich größere Rolle als das Aussehen der Objekte selbst. Um auf solch einer Datenbasis die länglichen Objekte gut klassifizieren zu können, müsste der VPL entsprechend hoch dimensioniert sein, was die Rechenzeit entsprechend stark beeinflusst.

In dem hier vorgestellten System eignet sich ein einfacher Trick, um dies zu vermeiden. Da die Objekterkennung in diesem System Bildausschnitte vom Aufmerksamkeitssystem geliefert bekommt und dieses bereits die Ausdehnung der Objekte als ROIs berechnet hat, kann die Ausrichtung dieser ROIs für eine bessere Objekterkennung verwendet werden. Dazu werden Objekte, bei denen das Verhältnis der vertikalen zur horizontalen Ausdehnung einen bestimmten Schwellwert θ übersteigt (als sinnvoll erwiesen sich Werte von $1,5 < \theta < 2$), als längliche Objekte betrachtet. Diese Bildausschnitte werden, bevor sie dem VPL als Eingabe dienen, in Richtung ihrer Hauptachse gedreht. Dies führt zu sehr viel ähnlicheren Eingaben der gleichen länglichen Objekte, wie man in Abb.11.3

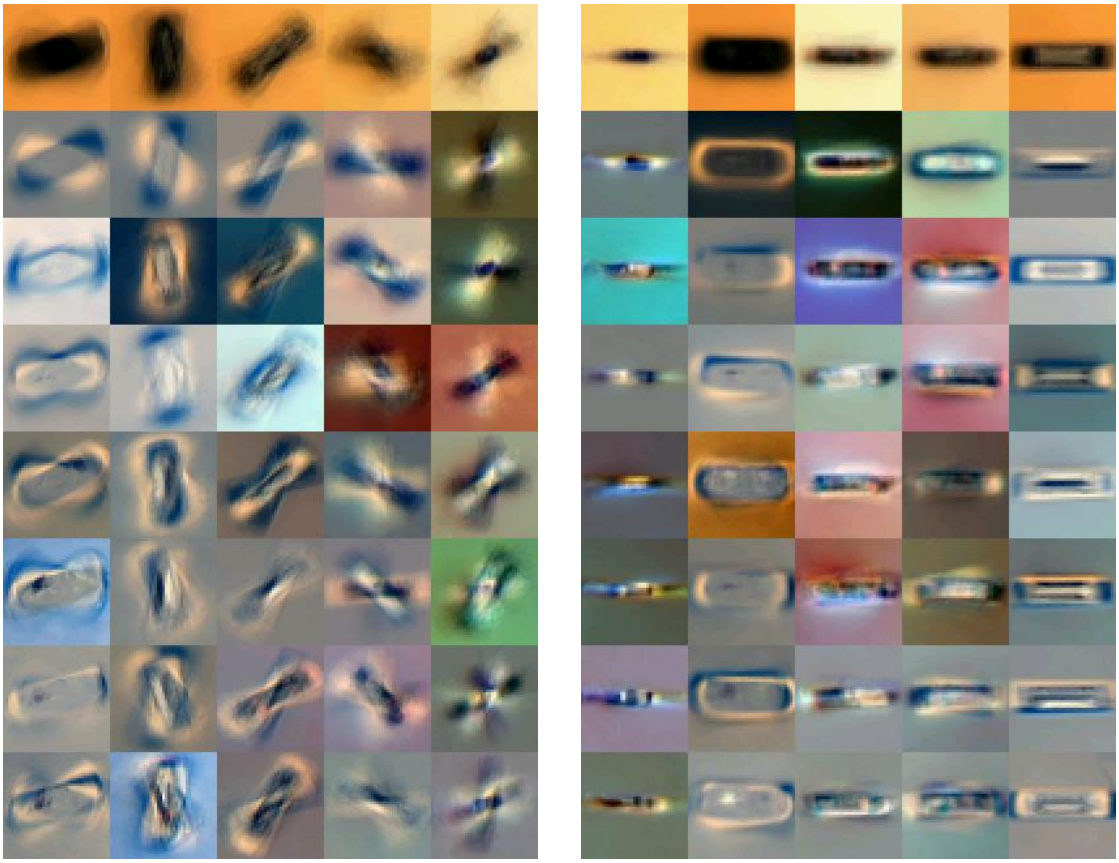


Abbildung 11.2: „Rezeptive Felder" zweier VPL's. Links: trainiert ohne vorherige Rotation der Objekte. Rechts: Rotations-2D-Alignment durch die Rotation des Bildausschnittes in Richtung der Hauptachse des Entropyblobs. Jede Spalte zeigt die ersten acht Hauptkomponenten für jeweils einen der fünf Prototypvektoren des Vektorquantisierungsschrittes.

sieht. In diesem einfachen Datensatz erkennt man, dass sich die Ansichten der einzelnen länglichen Objekte im Wesentlichen nur noch durch die Skalierung unterscheiden. Das Ergebnis dieser Rotation führt daher zu sehr viel selektiver wirkenden, rezeptiven Feldern des Klassifikators, wie die Abb. 11.2, rechts, zeigt. Man erkennt, dass so bereits die Vektorquantisierung des VPLs zu einer Trennung der Objekte führt. Die vorherige Ausrichtung von länglichen Objekte entlang der Hauptachse führt somit bei gleicher Dimensionierung der ersten zwei Stufen des VPL, also der Anzahl der Prototypvektoren und der Hauptkomponenten, zu deutlich verbesserten Klassifikationsergebnissen. Tabelle 11.1 zeigt die Klassifikationsfehler auf einem Testdatensatz aus jeweils 20 Ansichten von den 12 in Abb. 11.3 gezeigten Objekten. Schon bei diesem kleinen Datensatz zeigt sich die Steigerung der Performanz durch die Rotation deutlich, und der Klassifikationsfehler sinkt bei einzelnen Objekten um das Zehnfache. Die Ausnutzung der Information des Aufmerksamkeitssystems ermöglicht somit unter Erhaltung der Klassifikationsperfor-

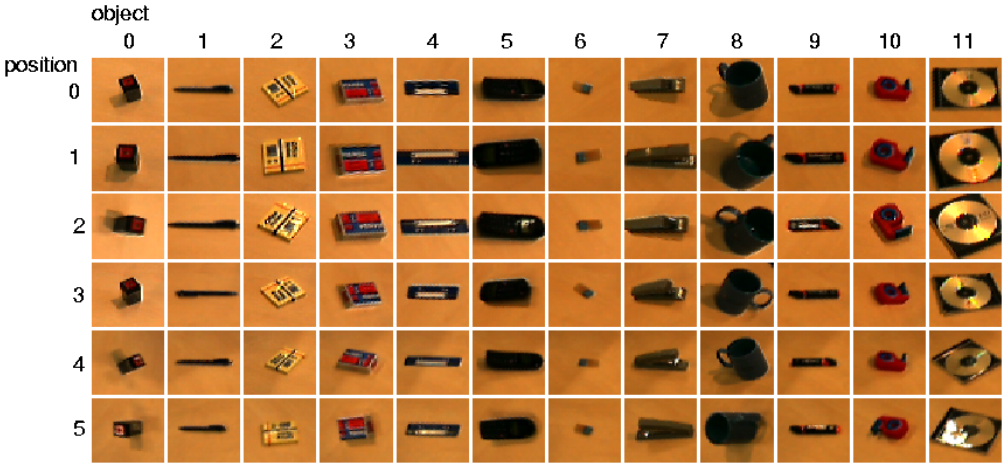


Abbildung 11.3: Verschiedene Objektansichten von Objekten auf einem Schreibtisch. Die Variation der länglichen Objekte ist auffällig gering, da sie in die gleiche Ausrichtung rotiert wurden.

Methode	Stift	Heftklammer	Telefon	Tacker	Textmarker
Unrotierte Objekte	2,534	3,926	0,647	3,571	7,004
Rotierte Objekte	0,0	0,617	0,0	0,396	0,772

Tabelle 11.1: Klassifikationsfehler mit und ohne Rotation der Objekte in Richtung der Hauptachse der ROIs.

manz die Verwendung eines deutlich niedriger dimensionierten Klassifikators und somit eine deutliche Reduktion der Rechenzeit.

Kapitel 12

Objektlernen durch Präsentation von Ansichten

Bereits in Kap. 10 wurden die zwei Wege beschrieben, auf denen ein Kind Objekte in seiner Umgebung erlernt. Der eine Weg ist das Bilden von Objektkonzepten durch das Hantieren bzw. durch das Sammeln von Erfahrungen mit den Objekten. Der zweite Weg soll hier kurz anhand eines Beispiels betrachtet werden, ohne dabei zu sehr in die Tiefe der neurologischen Forschung zu gehen: Beobachtet ein Kind zum ersten Mal die Mutter beim Zusammenheften von einzelnen Dokumenten mit Hilfe eines gewöhnlichen blauen Tackers, wird es sicherlich neugierig fragen, was dies ist. Für gewöhnlich gibt die Mutter daraufhin dem Kind das Objekt und nennt dabei den Namen des Objektes. Das Kind betrachtet das Objekt von verschiedenen Seiten und hat somit genau dieses blaue Objekt als Tacker kennengelernt. Wenige Ansichten genügen dem Kind, um das Objekt zu erlernen. Begleitet das Kind die Mutter zum ersten Mal ins Büro, lernt es dabei eine Vielzahl neuer Objekte kennen. Dabei wird es voraussichtlich die meisten Objekte direkt nach dem Erlernen richtig benennen können. Von langer Dauer ist dies allerdings noch nicht und so werden sicherlich nicht alle Objekte beim nächsten Besuch richtig erkannt. Bei dieser Art des Lernens werden Objekte sehr schnell in das Kurzzeitgedächtnis bzw. das Arbeitsgedächtnis überführt, welches direkt zur Verfügung steht. Die Überführung in das Langzeitgedächtnis ist zeitaufwändiger. Dies geschieht beim Menschen vor allem in einer Phase, in der die meisten Perzeptionskanäle keinen weiteren Input liefern, also in der Ruhephase im Schlaf. Dieses grundlegende Konzept aus einer Kombination aus schnellem, aber begrenztem Lernen und einer länger dauernden Überführung in ein Gedächtnis mit größerer Kapazität und Robustheit war Grundlage für die Entwicklung des Online-Lernens des vorgestellten Systems.

Im Gegensatz zu Systemen, in denen in einer offline-Trainingsphase eine Menge von gelabelten Objektansichten für das Training verwendet werden, welche unter Verwendung von Drehtellern und Roboterarmen automatisiert in künstlicher Umgebung erzeugt wurde, soll das vorgestellte System neu auftretende Objekte in der natürlichen Umgebung erlernen können. Eine automatisierte Erstellung von Trainingsdaten ist zwar sehr komfortabel, eignet sich allerdings nur für begrenzte Domänen im Hinblick auf die Anzahl der Objekte, der Objektgrößen, der Beleuchtungsbedingungen und der Kameraperspektiven. In dem mobilen System in einer natürlichen Umgebung eignet sich das Verfahren kaum. Die Umgebung zeichnet sich zudem durch eine Vielzahl von Objekten aus, welche das Erlernen aller Objekte zu einer nahezu unlösbaren Aufgabe macht. Daneben benötigt

ein mobiles System in einer natürlichen Umgebung eine Flexibilität gegenüber komplett neu auftretenden Objekten.

Um diese Flexibilität zu Erlangen und um ohne lange Trainingsphasen neue Objekte direkt für die Arbeit mit dem System zur Verfügung zu haben, wurde ein Verfahren entwickelt, mit dem zur Laufzeit neue Objekte durch einfache Präsentation erlernt werden können. Dazu werden dem System einzelne Objekte vom Benutzer präsentiert, so dass das System diese Objekte auf Basis weniger Ansichten vergleichbar mit dem Beispiel des Kindes erlernen kann. Dieses Wissen wird in einem ersten Schritt in eine Art Kurzzeitgedächtnis überführt, welches fast unmittelbar zur Verfügung steht. Somit kann zumindest eine begrenzte Anzahl von Objekten direkt erlernt werden, mit denen gearbeitet werden kann.

Wenn einem artifiziellen Objekterkenner auf diese Weise Objekte vermittelt werden, müssen an anderer Stelle Kompromisse eingegangen werden. Aufgrund der Geschwindigkeit des Erlernens kann die Erkennung dieser Objekte nicht besonders robust sein, sondern beschränkt sich auf die aktuelle Situation. Dem menschlichen Vorbild nachempfunden, wird in einer Phase, in dem das System Kapazitäten frei hat, die Erkennung robuster gemacht und somit das Wissen in eine Art Langzeitgedächtnis mit größerer Kapazität überführt.

Neben dem Erlernen neuer Objekte ermöglicht das Verfahren ebenfalls ein schnelles Beheben von Erkennungsfehlern. So kann z.B. die Vielfalt von Objekten mit gleicher Bezeichnung, wie ein roter Tacker mit einer völlig anderen Form, durch Präsentation zur Laufzeit erlernt und dadurch diese Objektkategorie erweitert werden oder aber die Erkennung von anderen Objekte bei veränderten Beleuchtungsbedingungen robuster gemacht werden.

12.1 Funktionsweise der Bilddatenaufnahme

Bei dem entwickelten Prototypen gelangt man über den Hauptmenüpunkt *Lern Obj* in das Untermenü zum Erlernen neuer Objekte durch Präsentation. Die Präsentation von Objekten aus verschiedenen Ansichten bedarf einer umfangreichen Interaktion und Kommunikation zwischen dem menschlichen Experten und dem System in der Rolle des Lernenden. Voraussetzung dafür ist eine Verständigung über das Objekt, welches im aktuellen Bild neu erlernt werden soll bzw. welches falsch erkannt wurde und für welches die Erkennung nun verbessert werden soll. In dem System wurden zwei Möglichkeiten dazu realisiert: Erstens kann dem System per Zeigegeste vermittelt werden, welches Objekt erlernt werden soll.

Eine zweite Vorgehensweise hat sich in der Praxis allerdings als komfortabler erwiesen. Dazu nimmt das System an, dass sich das zu erlernende Objekt im Zentrum des Bildes befindet. Dies wird erreicht, indem innerhalb des Aufmerksamkeitsmoduls eine Manipulatorkarte überlagert wird, welche im Zentrum des Bildes die Aufmerksamkeit hervorhebt und gaussförmig zu allen Seiten abfallen lässt. In beiden Fällen wird das vermeintlich referenzierte Objekt durch einen eingeblendeten roten Rahmen hervorgehoben.

Entspricht das vom System hervorgehobene Objekte nicht dem gewünschten, tritt der

Benutzer auch hier als Experte auf, indem er entweder durch Bewegung des Kopfes und somit Änderung der Kameraperspektive oder aber durch eine Variation der Zeigegeste den Fokus auf das richtige Objekt lenkt.

Die aktuelle Ansicht dieses Objektes kann nun aufgenommen werden. Das Aufmerksamkeitssystem liefert in beiden Fällen das Zentrum des gewünschten Objektes, so dass über die Menüsteuerung ein Bildausschnitt um diesen Fokuspunkt abgespeichert wird. Dazu kann entweder per Fingerspitze oder per Sprachkommando der Menüpunkt *Rec View* gedrückt werden. Als Rückmeldung vom System erklingt bei erfolgreicher Aufnahme das typische Geräusch eines Kameraauslösers, so dass der Benutzer über den Erfolg seiner Anweisung informiert ist. Anschließend kann der Benutzer in gleicher Art und Weise dem System verschiedene Ansichten des Objektes präsentieren und aufnehmen lassen. Je mehr Ansichten dem System präsentiert werden, desto robuster ist die Klassifikation. Die Anzahl der aufgenommenen Ansichten liegt somit im Ermessen des Benutzers. Je nach aktueller Aufgabe und Komplexität des Objektes kann der Benutzer die Anzahl variieren und alle relevanten Ansichten des Objektes aufnehmen. Für kompakte Objekte, welche von allen Seiten aus gleich aussehen, wie z.B. für einen einfarbigen Ball, reichen deutlich weniger Ansichten als z.B. für einen asymmetrischen Bilderrahmen. Sollten bei der Aufnahme Fehler passieren, indem beispielsweise der Fokuspunkt bei der Aufnahme zu einem anderen Objekt springt, können mit *Discard* diese Bildausschnitte wieder verworfen werden.

Im Anschluss an die Bildaufnahme kann durch Wahl des Menüpunktes *Apply* eine Objektbezeichnung per Sprachkommando gegeben werden. Wenn ein Objekt mit gleicher Bezeichnung bereits dem System bekannt ist, werden die neuen Ansichten der bereits bestehenden Wissensrepräsentation hinzugefügt. Nach einer kurzen Trainingszeit von wenigen Sekunden steht nun das neue Objektwissen zur Verfügung, und das System ist wieder betriebsbereit. Unbemerkt vom Benutzer werden im Hintergrund überschüssige Systemressourcen verwendet, um eine robustere Version des Klassifikators zu trainieren. Die beiden Trainingvarianten werden im Detail in den folgenden Kapiteln beschrieben.

12.2 Training des Objekterkenners

12.2.1 Virtuelle Vergrößerung des Bilddatensatzes

Um trotz der wenigen aufgenommenen Ansichten möglichst robuste Erkennungsergebnisse zu erreichen und den Aufwand des Benutzers bei der Bildaufnahme möglichst gering zu halten, wird der online aufgenommene Bilddatensatz vor dem Training des Klassifikators künstlich mit zwei verschiedenen Verfahren vergrößert:

Scherung und Skalierung: Das Erscheinungsbild von Objekten hängt im Wesentlichen von der Kameraperspektive ab. Je nach Blickwinkel erscheint das Objekt in einer kleineren oder größeren Skalierung und Scherung. Durch künstliches Skalieren und Scheren der Bildausschnitte wird der Trainingsdatensatz virtuell um diese Ansichten erweitert.

Translation: Das Aufmerksamkeitsmodul ermittelt nicht immer das genaue Zentrum

eines Objektes, um den Bildausschnitt für den Klassifikator zu positionieren. Um den Klassifikator gegen solche Verschiebungen des Bildausschnitts robust zu machen, werden die aufgenommenen Bildausschnitte künstlich in verschiedene Richtungen translatiert und dem Datensatz hinzugefügt.

12.2.2 Trainingsvarianten

Die Aufteilung des VPL-Klassifikators in verschiedene Verarbeitungsschichten macht es möglich, einerseits für eine Art Kurzzeitgedächtnis nahezu in Echtzeit neue Objekte zu erlernen und andererseits einen robusteren Erkenner durch ein vollständiges Training im Hintergrund zu erstellen. Der VPL besteht aus den drei Schichten für die Vektorquantisierung, die lokale PCA und die LLM-Netze (vgl. Kap. 6). Die ersten beiden Schichten beinhalten eine Vektorquantisierung der vorhandenen Bilddaten im Rohformat und einer anschließenden Hauptkomponentenanalyse auf die in Voronoizellen eingeteilten Gruppen dieser Bilddaten. Diese beiden Schritte sind durch die hohe Dimensionalität des Merkmalsraums und durch den Umfang der Datenbasis sehr rechenintensiv. Die Rechenzeit hängt annähernd linear von der Anzahl der Objekte ab. Der aufwändigste Schritt ist dabei die Berechnung der Hauptkomponenten durch das Sangernetz.

Die Projektionen der Bilddaten auf die Hauptkomponenten dienen anschließend den LLM-Netzen als Eingabevektoren. Diese Vektoren haben eine deutlich niedrigere Dimensionalität. Im Prototypen wurden meist VPLs mit der Dimension 5-8-20 verwendet, also fünf Vektoren für die Vektorquantisierung und acht Hauptkomponenten. Da für jede Zelle der einzelnen Prototypvektoren des Quantisierungsschrittes ein LLM-Netz trainiert wurde, besteht das Training der letzten Stufe aus dem Training von 5 LLM Netzen auf achtdimensionalen Merkmalsvektoren. Dieser Schritt dauert auf einem handelsüblichen Rechner weniger als eine Sekunde. Wenn bereits ein Objekterkenner trainiert wurde und somit bereits eine Merkmalsextraktion durch die V und P-Schicht vorliegt, kann unter der Annahme, dass diese Merkmale auch neue Objekte abdecken, nur die letzte Schicht neu trainiert werden. Diese Trainingsvariante wird im Folgenden als Schnelles Training (**ST**) bezeichnet. Trainiert man für eine Art von Kurzzeitgedächtnis nur diese Schicht neu, ist unmittelbar ein aktualisierter Klassifikator betriebsbereit, mit dem die neuen Objekte zumindest in der aktuellen Umgebung und unter den aktuellen Bedingungen erkannt werden können. Wie beim menschlichen Kurzzeitgedächtnis sind die Objekte auf diese Weise schnell erlernbar, das neu erlangte Wissen weist jedoch sowohl eine begrenzte Kapazität als auch eine geringere Robustheit auf (vgl. Kap. 12.3). Die Mannigfaltigkeit der Objektansichten unter verschiedenen Bedingungen kann so nicht komplett abgedeckt werden, da die Merkmalsextraktion der ersten Schichten des VPL-Klassifikators die erweiterte Domäne nicht optimal abdecken kann.

Eine robustere Klassifikation erreicht der VPL, indem die ersten beiden Schichten die diskriminativen Merkmale der Objekte in dem erweiterten hochdimensionalen Merkmalsraum in Form von rezeptiven Feldern extrahieren. Sowohl die Tesselierung der Daten mit dem *Activity Equalisation Vector Quantisation*-Algorithmus als auch die Hauptkomponentenanalyse durch das iterative Trainieren der verschiedenen Sangernetze sind auf den Rohdaten deutlich rechenaufwändiger als das Training der LLM-Netze. Für ein mobiles

System, welches sofort verfügbar neues Wissen akquirieren soll, eignet sich diese Art des Trainings nicht. Angelehnt an das menschliche Vorbild, kann das Erlernen des robusten Wissens jedoch im Hintergrund geschehen. Dazu werden alle drei Schichten auf der neuen Datenbasis trainiert (im Folgenden als kompletter Trainingsmodus **KT** bezeichnet). Ähnlich wie bei der Überführung von Gedächtnisinhalten in das Langzeitgedächtnis beim schlafenden Menschen kann bei dem künstlichen System in „Ruhephasen" das komplette Training in eigenen Threads im Hintergrund laufen und nicht genutzte Systemressourcen verwenden, ohne die aktuelle Funktion zu beeinflussen.

12.3 Evaluation

Bei der Entwicklung eines Systems mit den beiden vorgestellten Trainingsvarianten ergeben sich folgende Fragestellungen, welche in einer Evaluation überprüft werden sollen:

1. Wie stark hängt die Klassifikationsperformanz und die Trainingszeit von der Dimensionierung des VPLs ab?
2. Wie verhält sich das schnelle Lernen im Vergleich zu der Trainingsvariante des kompletten Lernens?
3. Welche Rolle spielt die Vielfalt der bereits erlernten Objekte und somit die Variabilität der bestehenden rezeptiven Felder auf die Performanz des schnellen Lernens?

Für die Evaluation der beiden Trainingsvarianten wurden Bilddaten von zwölf typischen Büroobjekten wie z.B. einem Tacker, einem Bleistiftanspitzer oder einen Textmarker unter festgelegten Bedingungen aufgenommen. Dazu wurden auf sechs markierten Positionen auf einem Schreibtisch jeweils 10 zufällig ausgewählte Ansichten von jedem Objekt aufgenommen, so dass pro Objekt 60 Ansichten zur Verfügung standen. Der Datensatz bestand somit aus 720 Bildausschnitten der Dimension 61×61 Pixel. Für das Trainieren wurden die 120 Bilder einer Referenzposition im Zentrum des Schreibtisches verwendet. Die restlichen 600 Bildausschnitte dienten als Testdatensatz. Somit stellte die Evaluation hohe Anforderungen an einen Klassifikator, da sich der Blickwinkel zwischen Trainings- und Testdatensatz erheblich unterscheidet, dadurch konnten die Unterschiede der beiden Modi besser aufgezeigt werden.

Abb. 12.1 zeigt die Performanz des VPL-Klassifikators im Verhältnis zur benötigten Trainingszeit im kompletten Trainingsmodus unter Variation der VPL-Dimensionen. Der 3-3-20-VPL bleibt zwar bis zu einer Objektanzahl von sieben Objekten noch unter einer Minute, jedoch sinkt die Erkennungsrate bei dieser Anzahl bereits auf unter 70 Prozent. Bei der Erhöhung der Dimension der einzelnen VPL Schichten werden bereits bei dem 5-5-20-VPL gute Klassifikationsergebnisse auf dem anspruchsvollen Datensatz erzielt. Jedoch steigt die Trainingszeit für das komplette Training schnell auf über zwei Minuten, was dazu führt, dass dies im laufenden System unpraktikabel wird. Die Dimensionierung des VPL steht somit in engem Zusammenhang mit der Klassifikationsperformanz und korreliert mit dieser positiv. Im Gegensatz dazu steigt die Rechenzeit

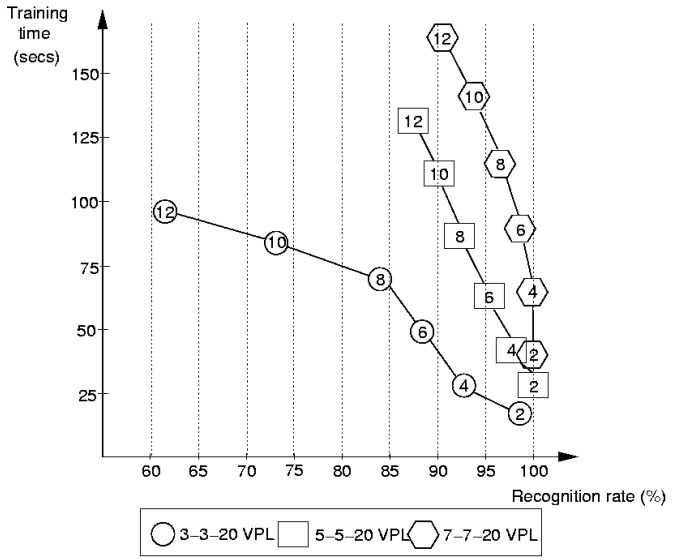


Abbildung 12.1: Klassifikationsrate und Trainingszeit in Abhängigkeit von der unterschiedlichen Dimensionierung der einzelnen Schichten eines VPLs im Hinblick auf eine variiernde Anzahl von Objekten bei komplettem Training. Die Zahlen in den Symbolen stehen für die Anzahl der Objekte (aus Bekel u. a. (2004)).

kontinuierlich einerseits bei Erhöhung der Datenbasis, auf der trainiert wird, und andererseits in Abhängigkeit von der Erhöhung der Dimensionierung des rechenaufwändigen Vektorquantisierungs- und des PCA-Schrittes.

Die Ergebnisse zeigen einerseits, dass selbst beim kompletten Training die Vielzahl von Objekten in einer natürlichen Umgebung erzwingt, dass die Dimension des Klassifikators deutlich gesteigert werden muss, und andererseits, dass dies selbst bei steigender Rechnerkapazität zukünftiger Systeme zu Trainingszeiten führen würde, die für ein online-Lernen zur Laufzeit kaum geeignet sind.

Die Frage, die daraus resultiert, ist, ob eine Kombination von schnellem und kompletten Lernen diesen Nachteil umgehen kann. Dazu wurden auf Basis von VPLs, welche mit einer variierenden Anzahl von Objekten im kompletten Trainingsmodus trainiert wurden, überprüft, wie sich das schnelle Training bei weiter steigender Objektanzahl in Bezug auf Klassifikationsperformanz und Trainingszeit verhält. Um das Verhalten gut dokumentieren zu können, wurde die Größe des VPL konstant mit der geringen Dimension von 3-8-30 gehalten. Abb. 12.2, links, zeigt die Ergebnisse im Hinblick auf die Klassifikationsperformanz. Die einzelnen Linien spiegeln die Performanz des Klassifikators im schnellen Trainingsmodus wider, welcher anfänglich mit 2, 4, 6, 8 und 10 Objekten im kompletten Modus trainiert wurde. Es zeigt sich, dass die Klassifikationsperformanz im schnellen Trainingsmodus im Wesentlichen von der vorher bereits bestehenden Merkmalsextraktion der V- und P-Schicht abhängt. Je größer die Datenbasis, auf der der VPL komplett trainiert wurde, ist, umso besser sind die Klassifikationsergebnisse selbst

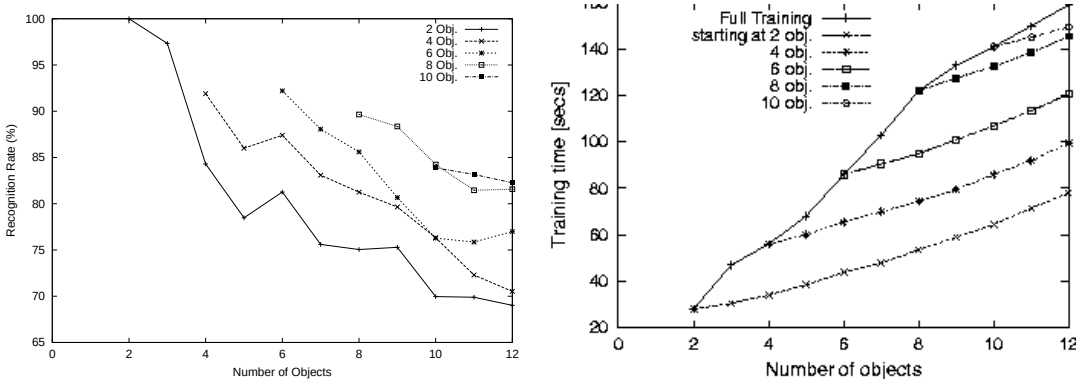


Abbildung 12.2: Links: Verlauf der Klassifikationsrate für den *schnellen Trainingsmodus*: Jede Kurve visualisiert den Abfall der Klassifikationsrate für einen Klassifikator der anfänglich im kompletten Trainingsmodus mit 2, 4, 6, 8 und 10 Objekten trainiert wurde und der dann im schnellen Trainingsmodus weitere Objekte erlernt hat. Die Dimension des verwendeten VPLs ist 3-8-20 (Heidemann u. a. (2007)). Rechts: Trainingszeiten für den *schnellen Trainingsmodus* verglichen mit dem *kompletten Trainingsmodus* (aus Bekel u. a. (2004)).

bei einer steigenden Anzahl von Objekten. Die Linien fallen nicht mehr so stark ab und bleiben bei einer umfangreichen Datenbasis, auf der die beiden Schichten V und P neu trainiert wurden, fast konstant.

In Abb. 12.2, rechts, wird die Trainingszeit im kompletten Modus im Vergleich zum schnellen Training gezeigt. Bereits bei dem niedrig dimensionierten VPL steigt die Trainingszeit im kompletten Training annähernd linear und beansprucht für jedes weitere Objekt im Schnitt ungefähr 16s mehr (durchgezogene Linie). Im Gegensatz dazu bleiben die Kurven im schnellen Trainingsmodus deutlich darunter. Unabhängig von der Größe der Datenbasis steigt die Trainingzeit pro hinzugefügtem Objekt lediglich um ca. 4s.

Daraus kann geschlossen werden, dass das schnelle Training gut geeignet ist, wenn auf Basis eines vielseitig trainierten Objekterkenners wenige neue Objekte erlernt werden sollen. Dies entspricht genau dem eigentlichen Zweck des Verfahrens, der darin besteht, in der aktuellen Situation „auf die Schnelle" eine fehlerhafte Erkennung zu korrigieren bzw. dem System schnell neues Objektwissen zu vermitteln, welches möglichst sofort verfügbar ist.

Kapitel 13

Objektretrieval - Wo ist mein Schlüssel?

Perspektivisch zielt die Entwicklung einer „Brille mit Gedächtnis" auf die Unterstützung von demenzkranken Personen oder Patienten mit limbischer Enzephalitis oder ähnlichem pathologischen Befund ab, bei denen die kognitiven Leistungen besonders im Bereich des Kurzzeitgedächtnisses gestört sind. Personen, die unter solch einer Erkrankung leiden, sind nicht in der Lage, sich an den Aufenthaltsort von Objekten zu erinnern, welche sie erst Minuten zuvor weggelegt haben.

Daraus ergibt sich die Aufgabe einer Brille mit Gedächtnis, welche darin besteht, den Aufenthaltsort von Objekten aus dem Umfeld einer Person in einer für die Person verständlichen Art und Weise mitzuteilen. (Im Informatikkontext wird dafür der englische Begriff „Retrieval" für Wiedergewinnung bzw. Auffinden verwendet.) Dies kann einerseits durch eine Lagebeschreibung in Relation zum aktuellen Aufenthaltsort sein, also in Form von Mitteilungen der Art „Der gesuchte Schlüssel befindet sich rechts von Ihnen" oder aber absolut durch eine Beschreibung des aktuellen Aufenthaltsortes, wie „Der gesuchte Schlüssel befindet sich auf dem Schreibtisch unter dem Buch". Zu der Funktionalität des Gesamtsystems, welches im Teil IV vorgestellt wird, gehört die genaue Orientierung im dreidimensionalen Raum. Die Aufgabe der Selbstlokalisation wird durch Tracking planarer Targets und durch eine Positionsschätzung durch Inertialsensoren gewährleistet (Projektpartner der Technischen Universität Graz). Dadurch kann die relative Beschreibung von Orten erfolgen.

Der hier vorgestellte Prototyp dient als Baustein, welcher später in das Gesamtsystem des Projektes integriert wurde (siehe Kap. 18). Dennoch sollte bereits der Prototyp den Aufenthaltsort verständlich mitteilen können. Anstelle der verbalen Mitteilung über den absoluten Ort eines Objektes wurde die Idee entwickelt, dass der Aufenthaltsort eines Objektes ebenso gut durch ein Bild des Objektes in seiner zuletzt wahrgenommenen Umgebung zum Wiederfinden des Objektes verwendet werden kann. Auf Basis dieser Idee wurde ein Bildergedächtnis entwickelt, welches später ebenfalls im Gesamtsystem als Teil in das VAM (also des **V**isual **A**ctive **M**emorys) integriert wurde.

Funktionsweise des Objektretrievals

Im laufenden System werden in jedem Frame die Objekte im Bild, deren Lage das Aufmerksamkeitssystem ermittelt hat, klassifiziert. Wird nun nach einem bestimmten Gegenstand gefragt, welcher im Bild vorhanden ist, wird dieser durch einen Rahmen

hervorgehoben. In der Regel ist dies jedoch nicht der Fall, sondern es handelt sich um eine Nachfrage eines Objektes, von dem der Aufenthaltsort vergessen wurde und welches somit nicht im aktuellen Bild zu erkennen ist. Unter der Annahme, dass ein Bild der Ansicht des Objektes in seiner zuletzt wahrgenommenen Position als Information für das Wiederfinden eines Objektes genügt, wurde ein Bildergedächtnis entwickelt. Das Gedächtnis besteht aus einer gelabelten Menge von Bildern. Diese Bilder werden im laufenden System folgender Maßen abgespeichert:

1. Das letzte Bild aus dem Bilderstrom wird temporär gespeichert. Dabei werden mit dem gleichen Algorithmus, wie in Kap. 10.4 vorgestellt, immer nur die relevanten Bilder im Anschluss an eine Bewegung (von dem Benutzer oder durch Bewegungen im Bild) abgespeichert.

2. Wenn im aktuellen Frame Objekte nicht mehr erkannt werden, die im letzten Bild noch erkannt wurden, wird das temporär gespeicherte letzte Bild auf ein Viertel der Bildgröße runterskaliert und als Bild der letzten Ansicht dieses Objektes und somit als Information über den aktuellen Aufenthaltsort des Objektes mit dem dazugehörigen Objektlabel abgespeichert.

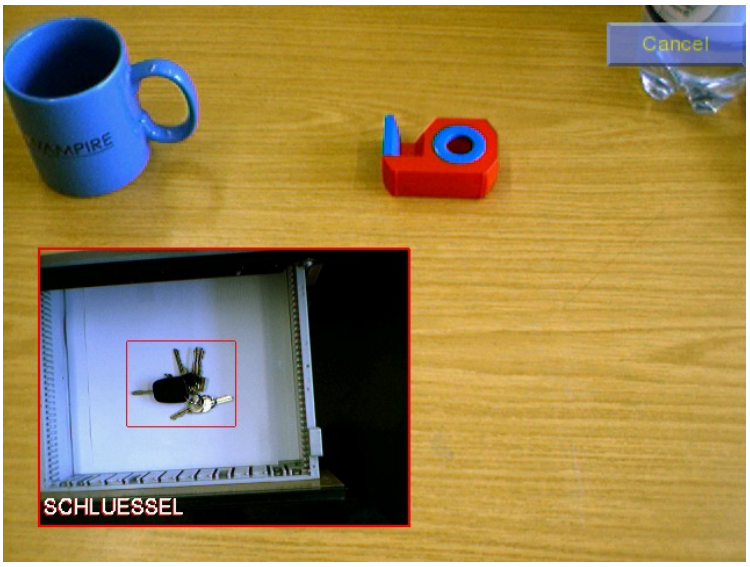


Abbildung 13.1: Die Antwort des Systems auf die Frage nach dem Schlüssel: Eingeblendeter ist der aktuelle Aufenthaltsort des gesuchten Objektes in der vom System zuletzt wahrgenommenen Umgebung, hier der Schreibtischschublade.

Wird nun im *Retrieval*-Modus nach einem Objekt gefragt, wird über das vom Spracherkenner erkannte Objektlabel in dem Gedächtnis das Bild des Objektes in seiner zuletzt wahrgenommenen Position im Bild eingeblendet. Abb. 13.1 zeigt ein Beispiel, in dem der gesuchte Schlüssel gezeigt wird, der zuletzt in der Schublade des Schreibtisches vom System erkannt wurde und somit in dieser Position dem Benutzer in einer kleineren Auflösung in dem aktuellen Bild eingeblendet wird. Das Beispiel zeigt, dass die Ansicht eines Objektes in seiner Umgebung im Allgemeinen reicht, um das Wiederfinden eines Objektes zu ermöglichen.

Kapitel 14

Settings

Der sechste Hauptmenüpunkt *Settings* führt in das Untermenü für die Systemeinstellungen. Abb. 14.1 zeigt noch einmal den Aufbau dieses Untermenüs. Es dient dazu, im laufenden System Parameter an wechselnde Umweltbedingungen anzupassen, einzelne Funktionen bei fehlerhafter Funktion auszuschalten oder deren Auswertung zu unterdrücken, oder aber bestimmte Datenmengen zu sichern. Wechselnde Umweltbedingungen können sich auf alle Erkennungsleistungen des Systems auswirken. Diese Umweltfaktoren können einerseits akustische sein, welche sich auf den Spracherkenner auswirken, wie beispielsweise starke Hintergrundgeräusche, oder aber andererseits die visuelle Wahrnehmung beeinflussen. Sollte die Hintergrundlautstärke so hoch sein, dass es eine fehlerfreie Menüführung durch die Erkennung der Sprachkommandos unmöglich macht, kann das System allein durch die Fingerspitzenerkennung gesteuert durch den Menüpunkt *Speach Off* die Spracherkennung ausgeschaltet werden. Somit kann es nicht mehr zu einer Fehlklassifikation und dadurch verursacht zu unerwünschtem Systemverhalten kommen. Variieren die Beleutungsbedingungen sehr stark und ist eine fehlerfreie Hautfarbenerkennung nicht mehr gewährleistet, kann diese im laufenden System nachjustiert werden, indem in das Untermenü *Adjust Skin*, welches in Kap. 14 beschrieben wird, gewählt wird. Sollte dies nicht zum Erfolg führen oder sollte der Erkenner durch andere Ursachen nicht fehlerfrei arbeiten können, kann durch den Menüpunkt *Fingertip Off* die Erkennung der Fingerspitze und somit die Menüführung über die visuelle Wahrnehmung der Fingerspitze ausgeschaltet werden.

Besonders in der Demoversion war es sinnvoll, flexibel im Hinblick auf Fehleingaben zu bleiben. Wenn versehentlich beim Lernen neuer Objekte Fehler gemacht wurden und der Klassifikator beispielsweise Objekte falsch gelabelt hat oder auf falschen Trainingsdaten trainiert wurde, kann eine stabil funktionierende Version geladen und die aktuelle verworfen werden. Dazu kann einerseits eine Defaultversion durch *Reset OR* oder andererseits eine zu letzt abgespeicherte stabile Version geladen werden, indem *Load OR* gewählt wird. Dazu kann, wenn der Objekterkenner neue Objekte erlernt hat oder aber robuster mit neuen Ansichten trainiert wurde, diese Version des aktuellen VPL mit *Save OR* abgespeichert werden. Anschließend kann durch *Cancel* das *Setting*-Menü wieder verlassen werden.

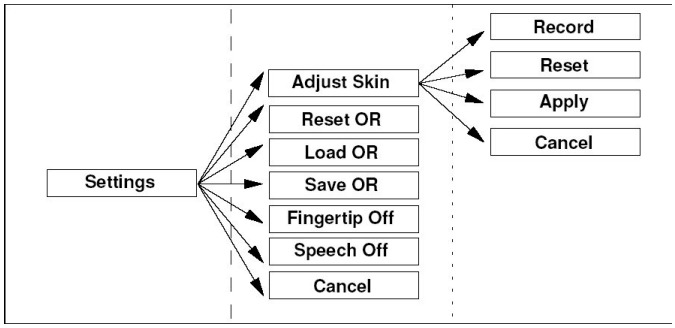


Abbildung 14.1: Das Untermenü für die Anpassung des Systems an geänderte Bedingungen zur Laufzeit (für Details siehe Text).

Anpassung der Hautfarbensegmentierung

Wenn die Hautfarbensegmentierung fehlerhaft ist, kann der Benutzer in kurzer Zeit die Segmentierung an die veränderten Bedingungen adaptieren.

Ist das *Adjust Skin*-Untermenü aktiv, werden die Pixel, die vom Hautfarbenerkenner als Hautfarbe erkannt worden sind, gelb dargestellt, und ein rotes Rechteck erscheint in der Mitte des Bildes, wie in Abb. 14.2. Der Benutzer agiert auch hier als Experte und kann entscheiden, ob er mit dem Erkennungsergebnis zufrieden ist und das System korrekt arbeitet. Ist die Hautfarbenerkennung fehlerhaft und werden die hautfarbenen Pixel im Bild nicht erkannt oder aber zu viele andere Pixel als Hautfarbe erkannt, kann die Hautfarbenerkennung neu eingestellt werden, indem der Hautfarbenerkenner neu trainiert und somit an die aktuellen Bedingungen adaptiert wird. Die Verteilung der Hautfarbe im r-g-Farbraum wird dabei aus Bildauschnitten von der Hand des Benutzers abgeleitet. Dazu hält der Benutzer die Hand vor den rot angezeigten Rahmen im Bildzentrum. Dieser Bildausschnitt wird über den Menüpunkt*Record* aufgenommen, wobei jeweils die Pixel in dem rot angezeigten Rechteck aufgenommen und zum Trainieren des Erkenners benutzt werden. Das Zentrum der Farbverteilung dieser Pixel und die Standardabweichung der Verteilung in r- und g- Richtung werden dabei bestimmt und die Parameter der quadratischen Funktion, welche im Erkenner den Skinlokus eingrenzen, dementsprechend angepasst (vgl. Kap. 7.1). Dazu werden die r-Koordinaten der Scheitelpunkte der beiden quadratischen Funktionen auf die Höhe des Verteilungsschwerpunktes gelegt und die g-Koordinaten um den doppelten Abstand der Standardabweichung vom Schwerpunkte der Verteilung verschoben. Werden versehentlich falsche Bildbereiche aufgenommen, kann mit *Reset* in den vorherigen Zustand mit dem vorher trainierten Klassifikator zurückgegangen werden.

Nach jeder Aufnahme der Pixel im rechteckigen Fenster werden der Klassifikator neu trainiert und das neue Ergebnis direkt angezeigt. Der Benutzer entscheidet, wie viele Bildausschnitte für die Bestimmung der Hautfarbenverteilung nötig sind, damit die Segmentierung robust ist. Ist die Hautfarbensegmentierung korrekt, wird die Parametrisierung des Klassifikators durch Drücken von *Apply* übernommen und in das Hauptmenü

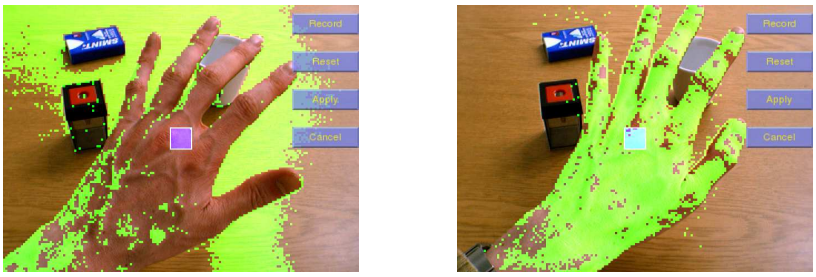


Abbildung 14.2: Hautfarbenadaption: Links: Die Visualisierung der als Hautfarbe erkannten Pixel zeigt, dass die Segmentierung fehlerhaft ist und in erster Linie die Tischfarbe als Hautfarbe erkannt wird. Der Benutzer hält nun die Hand unter das Zielrechteck und nimmt mehrere Bildausschnitte zur Readaptation des Hautfarbenklassifikatrs auf. Rechts: Nach der Adaptation arbeitet die Segmentierung präzise und ermöglicht wieder eine korrekte Interaktion von Mensch und Maschine durch Handgesten.

zurückgegangen.

Durch das Ausnutzen des Einschätzungsvermögens des Benutzers in Kombination mit der einfachen Readaptation durch die Mensch-Maschine-Interaktion empfiehlt sich dieses Verfahren für ein mobiles System, da die rechenaufwändigen Verfahren zur Hautfarbensegmentierung, welche auf Wahrscheinlichkeitsmodellen und Trackingmechanismen beruhen, umgangen werden können. Dieses Beispiel zeigt erneut, dass der Nachteil unpräziser Ergebnisse des maschinellen Systems durch die Verwendung einfacher Verfahren völlig unproblematisch durch geschicktes Ausnutzen des Benutzers als Experten mittels einfacher Mensch-Maschine-Interaktion umgangen werden kann.

Teil IV

Einbettung in das übergeordnete System

Das bisher beschriebene System in Form eines Prototypen einer Brille mit Gedächtnis ist nur ein Teil des übergeordneten Ziels des Projektes VAMPIRE, der Entwicklung eines Büroassistentensystems. Um aus dem hier vorgestellten lernfähigen Objekterkennerprototypen perspektivisch so etwas wie eine „Brille mit Gedächtnis" oder gar ein Assistentensystem zu erstellen, bedarf es weiterer wesentlicher Funktionalitäten.

Im Rahmen des Projektes steht hinter der umgangssprachlichen Beschreibung der „Brille mit Gedächtnis" eine vielschichtige Architektur mit einem visuellen aktiven Gedächtnis (VAM) als zentrale Komponente. Dieses visuelle aktive Gedächtnis geht deutlich über die Möglichkeiten des Bildgedächtnisses im hier beschriebenen Subsystem hinaus. Das Bildergedächtnis, welches zusätzlich eine 3-D-Position der Objekte enthält, befindet sich auf der sensorischen Gedächtnisebene eines hierarchisch strukturierten und in vier Ebenen unterteilten VAM. Die weiteren Schichten sind ein merkmalsbasiertes Gedächtnis, ein episodisches Gedächtnis und ein kategorisches Gedächtnis auf der obersten Hierarchieebene. Ein Überblick über die Gedächtniskomponenten und die dazu notwendigen Fähigkeiten liefert die Abb.14.3. Sowohl die Konzeption des VAM als auch die Realisierung der weiteren Systemkomponenten wurden innerhalb des Projektes von den verschiedenen Projektpartnern bearbeitet.

Die Kernbereiche der einzelnen Projektpartner bestehen aus i) grundlegenden Trackingverfahren, ii) der Annotation von Bildsequenzen, iii) der Integration des Kontextes und iv) der Lernfähigkeit des Systems auf verschiedenen Ebenen.

Die folgende Liste gibt einen Überblick über die beteiligten Projektpartner und der Zuordnung der einzelnen Aufgaben, welche schwerpunktmäßig von den einzelnen Partnern des Projektes VAMPIRE bearbeitet werden:

- *Technische Universität Graz*
  - *Konstruktion und Inbetriebnahme der Hardware*
  - *Selbstlokalisation durch Tracking planarer Targets*
  - *Positionsschätzung durch Inertialsensoren*
- *Angewandte Informatik, Universität Bielefeld*
  - *Systemintegration und -koordination*
  - *Vernetzung des Systems durch XCF*
  - *Konstruktion der Gedächtnisstrukturen und -funktionalitäten*
  - *einfache Handlungserkennung*
- *Universität Nürnberg-Erlangen*
  - *Tracking von Objekten basierend auf Hyperplanes, Farbhistogrammen und Kantenmerkmalen*
- *University of Surrey*
  - *Erkennung zusammenhängender Oberflächen oder auch Mosaiking*

– *Konstruktion eines episodischen Gedächtnisses in Form von Entscheidungsbäumen im Kontext eines Wimbelton-Tennismatches*

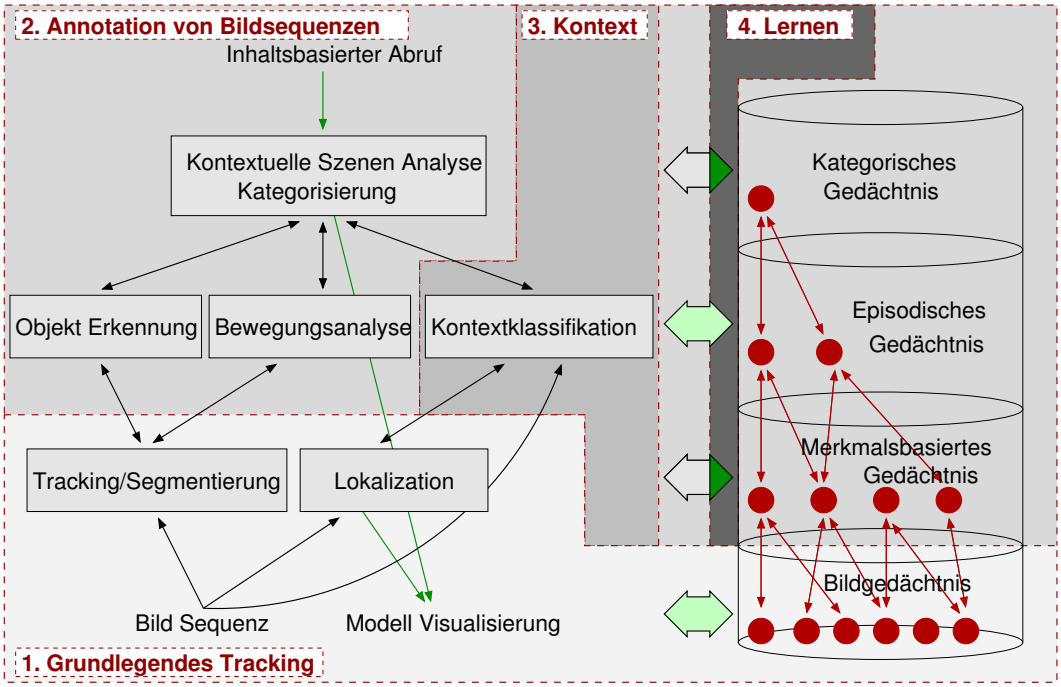


Abbildung 14.3: Aufbau des Gesamtsystems (http://www.vampire-project.org/)

Die einzelnen Komponenten des Gesamtsystems werden in den folgenden Abschnitten näher beschrieben, wenngleich für detaillierte Beschreibungen auf die einzelnen Arbeiten verwiesen wird. Schwerpunkt dieses Teils ist die Vorstellung des Gesamtsystems um die Integration der vorliegenden Arbeit als wichtige Komponente des VAMPIRE-Assistentensystems aufzuzeigen.

Kapitel 15

Überblick über die weiteren Systemkomponenten

15.1 Mosaiking

Die visuelle Perzeption des vorgestellten Systems basiert auf der Aufnahme von einer zeitlichen Abfolge von visuellen zweidimensionalen Informationen. Um ein komplettes räumliches Modell einer dreidimensionalen Szene aus dem vorhandenen Bildstrom zu gewinnen und um dynamische Veränderungen in der Szene besser zu erkennen, wurde von der Arbeitsgruppe aus Surrey um Josef Kittler ein Mosaiking-Verfahren entwickelt. Dieses Verfahren ermöglicht, räumliche Zusammenhänge über sich verändernde Perspektiven und über den begrenzten Bildausschnitt der Kamera hinaus durch eine Modellierung zusammenhängender Flächen zu erreichen. Bei der Interpretation der Daten des beschriebenen Systems geht jeglicher raum-zeitlicher Zusammenhang zwischen den Einzelbildern verloren. Des Weiteren können zwar Teile von großen Objekten erkannt werden, ihre räumliche Relation und die Interpretation als zusammenhängendes Objekt kann das vorliegende System jedoch nicht erkennen. Diese Nachteile kann das Mosaiking-Verfahren umgehen. Die dreidimensionale Welt, welche durch die sich bewegende Kamera aufgenommen wird, soll dabei in eine Repräsentation eines zusammenhängenden 2-D Bildes überführt werden. Diese Transformation ist allerdings grundsätzlich nicht wohldefiniert und somit ohne Verluste bzw. umkehrbar zu erhalten. Eine 2-D Ansicht eines 3-D Objektes kann, wie bereits mehrfach in den vorigen Kapiteln gezeigt, völlig unterschiedlich sein. Die Transformation von 2-D-Koordinaten auf 3-D-Koordinaten ist somit unterrepräsentiert. Das Problem wird in Mosaikingverfahren umgangen, indem die Bewegung der Kameraperspektive eingeschränkt wird, wie es z.B. in der Domäne des Tennisszenarios der Fall ist. In einem mobilen AR-System ist diese Einschränkung jedoch unerwünscht. Eine weitere Möglichkeit, die Wohldefiniertheit der Transformation zu erreichen, kann man durch ein Mosaikingverfahren erreichen, bei dem als Grundlage von einer planaren Oberfläche ausgegangen wird, so dass eine der 3-D-Koordinaten sich nicht verändert und bei den notwendigen Transformationen auf 0 gesetzt werden kann. Somit wird die 3-D-Szene in planare Sub-Szenen zerlegt, welche für die Erkennung in einer von Menschen geschaffenen Umgebung wertvolle Informationen erhält, da alle Arbeitsplattformen, wie z.B. der Schreibtisch, eine (allein schon wegen der Gravitation sinnvolle) zweidimensionale Fläche aufweisen (Gorges u. a., 2004). Die Lokalisation der Objekte im Hinblick auf den Kontext ist somit einfacher einzuordnen und kontextuelle

Gedächtnisinhalte, wie *Die Tasse steht auf dem Tisch*, sind möglich.

15.2 Selbstlokalisation und 3-D Positionsbestimmung

Das in dieser Arbeit beschriebene System verwendet zur Vermittlung über die Gedächtnisinhalte des Aufenthaltsortes von Objekten ein Bildergedächtnis, welches ein Objekt in seiner zuletzt wahrgenommenen Position zeigt. Da diese Bilddaten aufgrund der Bauweise des EVS aus dem Blickwinkel des Benutzers aufgenommen wurden, genügen diese Bilddaten in den meisten Fällen für die Lokalisation der gesuchten Objekte.

Das kontextuelle Gedächtnis, welches ebenfalls Handlungen des Benutzers in einen Kontext einordnen soll und somit gegebenenfalls bei bestimmten Handlungen durch Anweisungen assistieren soll, bedarf einer Repräsentation der dreidimensionalen Szene. Für diese Repräsentation ist einerseits die Bestimmung der Position des Benutzers und andererseits die dreidimensionale Lokalisation der Objekte aus dem zweidimensionalen Kamerabild in einem globalen dreidimensionalen Koordinatensystem notwendig.

Abbildung 15.1: Hardware mit CMOS-Kamera

Zur Bestimmung der Pose des Benutzers, also der Kombination aus Position und Orientierung, wurde ein Verfahren entwickelt, bei dem über das visuelle Verfolgen planarer, sogenannter Käseecken-Targets die sechs Freiheitsgrade für die Position und Orientierung berechnet werden können. Diese kontrastreichen schwarzen Targets auf weißem Hintergrund in Form einer Käseecke weisen sieben Ecken mit bestimmten Orientierungen auf, welche die Berechnung der Pose ermöglichen (M. K. Chandraker und Pinz, 2003). Um auch während der Bewegung des Benutzers in Echtzeit die Freiheitsgrade möglichst präzise zu bestimmen, wird im Gesamtsystem eine CMOS-Kamera verwendet, welche auf dem Helm positioniert ist. Dadurch kann die Pose mit 15 Hz bestimmt werden. Unterstützt wird die Posenerkennung durch das Auswerten von Daten vom am Kopf befindlichen Inertialsensoren.

Für die kontextuelle Einordnung der dreidimensionalen Szene ist neben der Posenbestimmung des Benutzers die Lokalisation der umgebenden Objekte essentiell. Biologisch motiviert, kann die Entfernung zu Objekten aus der Disparität von Stereobildern berechnet werden. Die verwendete Hardware verfügt, wie bereits beschrieben, über Stereokameras. Da aus der Posenberechnung der Blickwinkel des Benutzers bekannt ist, kann durch die Verwendung der Kamerabilder die Objektposition bestimmt werden.

Für die kontextuelle Interpretation einer 3-D-Szene kann weiterhin eine Kombina-

tion der beiden Verfahren des Mosaiking für die Erkennung planarer Ebenen mit der Selbstlokalisation durch das Tracken der planaren Targets ermöglicht werden. Unter der Annahme, dass sich die Objekte in einer künstlichen Umgebung immer an oder auf planaren Flächen befinden, kann bei bekannter Pose mit nur einer Kamera der Aufenthaltsort eines Objektes in Relation zu der bereits ermittelten Fläche bestimmt werden (Hanheide, 2006).

Kapitel 16

Handlungserkennung mit Hilfe verschiedener Trackingverfahren

Erweitert man die Idee der Brille mit Gedächtnis zu einem Assistentensystem, so ist nicht nur das Verständnis über den Aufenthaltsort von Objekten notwendig, sondern das System muss erkennen, was der Benutzer mit den Objekten macht. Einfache Beispiele für Handlungen im Büroszenario sind das Telefonieren, wobei das Objekt *Telefon* vom Tisch aufgenommen und an das Ohr geführt wird oder das Arbeiten am Computer, wobei entweder die Finger die Tastatur bedienen oder die PC-Maus mit der Hand geführt wird. Werden diese Aktionen erkannt, so können sie in das kontextuelle Gedächtnis überführt werden und das Assistentensystem kann darauf reagieren. Ein erweitertes Beispiel für ein Assistentensystem könnte perspektivisch einen Benutzer dabei unterstützen, neu erworbene Möbel zusammen zu bauen. Auch hierfür hantiert der Benutzer mit Objekten, und das System muss für die Unterstützung bei der Tätigkeit erkennen können, was der Träger des mobilen Systems zur Zeit mit den Bauteilen zusammengebaut hat. Dann kann das System entsprechend reagieren und durch Anweisungen ggf. die Handlung korrigieren. An den Beispielen kann erkannt werden, dass es bei der Handlungserkennung nicht um einfache Bewegungserkennung geht, sondern vielmehr um Bewegungen im Kontext mit der Umgebung bzw. den umgebenen Objekten und deren Auswirkungen auf diese.

Im Projekt VAMPIRE werden die Basisinformationen für solch eine Handlungserkennung von der Objekterkennung, der Posenerkennung und der kontextuellen Einordnung der Umgebung durch die Kombination aus den Ergebnissen des Mosaiking und der Objekterkennung geliefert. Diese Informationen dienen als Grundlage für einen trajektorienbasiertes Verfahren zur Handlungserkennung.

Zusätzlich zu diesen Informationen muss die raum-zeitliche Trajektorie der Hand bzw. von Objekten in der Hand erkannt werden. Die Verwendung des mobilen Systems stellt dabei eine große Herausforderung dar, da sich sowohl das Kamerabild als auch das Objekt bzw. die Hand im Bild bewegen. Für diese Aufgabe wurden von dem Projektpartner der Universität Nürnberg-Erlangen zwei echtzeitfähige robuste Trackingverfahren entwickelt, einerseits ein Regionentracking, welches aus dem Bereich des sogenannten Hyperplanetrackings stammt (Gräßl u. a., 2005), und andererseits ein Farbhistogrammbasiertes, sogenanntes Kernel-based Tracking (Bajramovic u. a., 2005).

Von der Arbeitsgruppe der Angewandten Informatik der Universität Bielefeld wurden diese Verfahren verwendet, um einerseits die Trajektorie von Objekten in der Hand und andererseits die Bewegung des Hintergrundes durch die Kamerabewegung zu be-

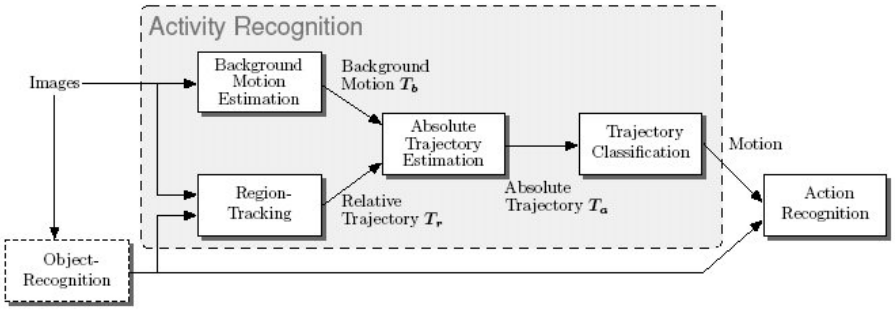


Abbildung 16.1: Handlungserkennung aus einer Kombination von Hintergrund- und Objekttracking mit kontextuellen Informationen des Gedächtnisses (aus Hanheide (2006)).

stimmen. Hieraus wird dann die eigentliche Trajektorie der Handlung bestimmt und unter Verwendung der Informationen des Objekterkenners, der kontextuellen Lagebestimmung der Objekte und des Benutzers im Raum die Handlung erkannt. Den Ablauf der Handlungserkennung zeigt Abb. 16.1.

Kapitel 17

XCF-System

Aufgrund der Herausforderung, die vorgestellten Algorithmen verteilt auf verschiedenen Rechnern laufen zu lassen, eine Kommunikation zwischen diesen zu ermöglichen und dabei echtzeitfähig zu bleiben, wurde ein neues Framework entwickelt. Diese Herausforderungen wurden auf einer XML basierenden Integrationsplattform oder *Middleware*, dem sogenannten XCF (XML enabled Communication Framework), gelöst. Diese von Sebastian Wrede entwickelte Middleware ist ein leicht zu benutzendes Werkzeug, um objektorientiert verteilte Systeme zu konstruieren. XCF bietet ein XML standardisiertes Framework für eine einfach zu benutzende Software in Form einer Integrationsplattform an. Es basiert hauptsächlich auf ZeroC's Internet Communication Engine (ICE) als objektorientierte Middleware und nutzt als Datenbasis die native XML Datenbank von Berkeley (DBXML, (Berkeley, 2004)).

Es bietet die Möglichkeit der Verarbeitung von synchronen Datenströmen, RPC (*remote procedure calls*) und RMIs (engl. für *Remote Method Invocation*).

Für das Speichern der Gedächtnisinhalte des Gesamtsystems wurde XML gewählt, da es sehr flexibel einsetzbar ist und eine einfache abstrakte Konzeptbeschreibung ermöglicht.

Für die Kommunikation über die Gedächtnisinhalte des Systems werden Daten in XML-Syntax binarisiert übermittelt (nach der XOP-Konvention, engl. für *XML-binary Optimized Packaging*). Zusätzlich dazu besteht die Möglichkeit, größere Datenmengen, wie z.B. Bilder, binär zu übertragen. Durch die Spezifikation von XML Schemata ist die Laufzeittypsicherheit garantiert, und die Programmierung von Schnittstellen ist aufgrund der API intuitiv und auch für Middleware-Einsteiger leicht zu erlernen.

Kapitel 18

Integration des vorgestellten Systems in das Gesamtsystem

Der vorgestellte Prototyp diente der Entwicklung verschiedener Teilkomponenten des VAMPIRE Gesamtsystems. Der ursprünglich beabsichtigte Schwerpunkt der Entwicklung eines lernfähigen Objekterkenners wurde dabei ausgeweitet, indem für die interaktive Lernfähigkeit als ersten Schritt die Entwicklung einer möglichst natürlichen Mensch-Maschine-Interaktion in den Vordergrund rückte.

Aus der Entwicklung des vorgestellten Prototypen gingen folgende Komponenten ganz oder zumindest teilweise in das Gesamtsystem ein:

1. die Entwicklung eines semi-transparenten virtuellen Menüs
2. die Entwicklung einer per Sprache und Fingerspitze zu bedienenden Menüführung
3. die Zeigerichtungserkennung und deren Visualisierung
4. die Objekterkennung und das interaktive Objektlernen durch Präsentation von Objektansichten
5. die Objektlokalisation durch die Verwendung eines Bildergedächtnisses
6. die Idee der Systemsteuerung der Input- und Outputkanäle über einen XML-basierten Kontroller

Die koordinierende Zentrale im Gesamtsystem besteht aus dem von M. Hanheide entwickelten VAM (Hanheide, 2006). Ein vom VAM gesteuerter Server übernimmt die Visualisierung des Menüs und der für den Benutzer wichtigen Daten, wie Objektlabels, Zeigerichtung etc. Die Spracherkennung läuft separat auf dem Gesamtsystem und kommuniziert ebenfalls über das VAM.

Somit gehen die Teile des vorgestellten Systems in das Gesamtsystem ein, welche dem VAM die Information über die Zeigegestenerkennung, die Objekterkennung und die Fingerspitzenerkennung für die Menüführung liefern. Dazu läuft auf dem Gesamtsystem auf einem Thread eine NESSY-Instanz, welche die notwendigen Daten auswertet und über XCF dem VAM zur Verfügung stellt.

Für die Kommunikation werden drei XML-Objekte verwendet:

1. Für die Menüführung wird ein XML Objekt erstellt, welches Informationen über den Interaktionsstatus (also ob ein Menüpunkt gewählt oder gedrückt wurde), über die Position des Fingers und über die aktuelle Frame-Nummer liefert.

2. Das Ergebnis der Zeigerichtungserkennung wird in ein XML Objekt verpackt, welches über die Zeigerichtung und die Handposition informiert.

3. Pro erkanntes Objekt wird jeweils ein XML-Objekt gesendet, welches das Label, die x- und y-Koordinaten im Bild, die umgebende Region, die Klassifikationswahrscheinlichkeit, die Nähe zur Hand und die Wahrscheinlichkeit, ob auf das Objekt gezeigt wird, enthält.

Teil V

Schluss

Kapitel 19

Zusammenfassung und Ausblick

19.1 Zusammenfassung

Informatische Forschung beschäftigt sich zumeist mit der Spezialisierung auf einen bestimmten Bereich und der Entwicklung optimierter Verfahren für eine bestimmte Aufgabe. Innerhalb des Projektes VAMPIRE wurde das ehrgeizige Ziel verfolgt, ein umfassendes Gesamtsystem in Form eines computergestützten Assistentensystems zu entwickeln, welches eine Vielzahl an Funktionalitäten bieten soll. Bei dieser Entwicklung wurden ebenfalls die zu lösenden Aufgaben auf verschiedene Arbeitsgruppen aufgeteilt und von den jeweiligen Experten bearbeitet.

In dieser Arbeit lag der Hauptfokus darauf, ein hochgradig integriertes und vor allem lernfähiges Gesamtsystem in Form eines Demonstrators zu konstruieren, welcher bereits eine Vielzahl von Funktionen des späteren Assistentensystems integrieren sollte. Der Schwerpunkt lag hier nicht auf der Spezialisierung in einem bestimmten Bereich, sondern vielmehr darauf, eine Vielzahl von benötigten Verfahren zu finden, für die Aufgabe zu modifizieren und gemeinsam in ein System zu integrieren, um eine Brille mit Gedächtnis überhaupt realisieren zu können.

Technische Basis für die Realisierung einer Brille mit Gedächtnis ist eine *Augmented Reality*-Apparatur, deren Kernstück aus zwei Kameras und einem 3-D-Display besteht und welche für die akustische Kommunikation über ein Mikrophon und über Kopfhörer verfügt. Dies ermöglicht, dass die visuelle Perzeption von Mensch und Maschine identisch ist und Systeminformationen sowohl visuell als auch akustisch mitgeteilt werden können.

Auf dieser Hardwareplattform wurde ein System entwickelt, das eine Vielzahl von Funktionen so integriert, dass es auf natürliche Weise mit dem Benutzer interagieren und Gedächtnislücken des Benutzers in Bezug auf verlegte Gegenstände schließen kann. Das System kann einerseits Objekte erlernen und andererseits dem Benutzer sein erlangtes Wissen mitteilen.

Bei der Entwicklung eines solchen Systems ist die große Herausforderung in der Kombination von Echtzeitfähigkeit und der komplexen Funktionsweise in einem mobilen System zu sehen. Um ein System zu entwickeln, welches zur Laufzeit Wissen durch möglichst natürliche Mensch-Maschine-Kommunikation erlangen und wiedergeben kann, mussten Wege gefunden werden, mit denen bei der Verarbeitung Rechenzeit gespart werden kann. Eine der Kernideen für die Lösung dieses Problems bestand darin, den Benutzer als Experten für die Korrektur bzw. die Steuerung des Systems zu verwenden. In dieser Arbeit wird aufgezeigt, dass erst die Integration des menschlichen Experten in die Verarbei-

tungsprozesse des künstlichen Systems die Verwendung von Verfahren ermöglicht, welche zwar eine geringere Leistungsfähigkeit bieten als vergleichbare komplexere Algorithmen, dafür aber einen sehr geringen Rechenaufwand benötigen und somit überhaupt erst für ein mobiles System in Frage kommen. Die fehlende Robustheit und Fehlerfreiheit der informatischen Verfahren wird durch die enge Interaktion mit dem menschlichen Experten kompensiert.

Um den Benutzer eines solchen Systems als Experten für die Korrektur von Systemungenauigkeiten oder gar -fehlern mit einbeziehen zu können, wurde in dieser Arbeit ein System entwickelt, mit dem über die natürliche Kommunikation durch Sprache und Gestik ein gemeinsames Verständnis von Mensch und Maschine im gegebenen Kontext erlangt werden konnte. Dabei kann der Benutzer sowohl durch Zeigegesten die Aufmerksamkeit des Systems lenken als auch durch natürliche Bewegungen mit dem Finger die Systemfunktionen über ein virtuelles Menü steuern. Alternativ kann die Steuerung des Systems verbal erfolgen.

Für den Fall, dass die visuelle Erkennung der Gestik durch geänderte Lichtverhältnisse oder die akustische Erkennung der Sprache durch Störgeräusche fehlerhaft sind, gewährleistet die Kombination dieser beiden Kommunikationsmöglichkeiten, dass der Benutzer das System über den alternativen Kommunikationsweg an die veränderten Bedingungen anpassen kann, so dass in beiden Fällen die Funktionalität erhalten bleibt. Das System kann dem Benutzer sowohl durch akustische Signale als auch über das Display Informationen, wie beispielsweise über die Systemzustände oder Verarbeitungsergebnisse, mitteilen; auf diese kann der Benutzer bei Bedarf reagieren und damit das System steuern.

Neben der Funktion zur Systemsteuerung ist die natürliche Kommunikation eine weitere wesentliche Voraussetzung für die Entwicklung eines online trainierbaren und somit lernfähigen Objekterkenners. Dem Objekterkenner des entwickelten Demonstrators kann neues Objektwissen einerseits durch Präsentation von Objekten vermittelt werden. Andererseits kann das System selbstständig, von einem Aufmerksamkeitssystem gesteuert, Bilddaten aufnehmen und diese interaktiv durch Kommunikation mit dem Benutzer für den Objekterkenner verfügbar machen.

Die erste Möglichkeit interaktiven Objektlernens besteht darin, dass dem System durch den Benutzer neue Objekte in natürlicher Art und Weise aus mehreren Ansichten präsentiert werden. Das System nimmt dabei die Bilddaten auf und erlernt die Objekte nach Nennung des Namens. Diese Variante eignet sich insbesondere für das Erlernen von solchen Objekten, mit denen der Benutzer hantiert.

Dagegen ermöglicht die zweite Variante, dass auch Objekte aus der Umgebung des Benutzers komfortabel gelernt werden. Dazu werden, von einem Aufmerksamkeitssystem gesteuert, Bilddaten aufgenommen, während sich der Benutzer ganz natürlich in seiner Umgebung bewegt. Diese Bilddaten können anschließend unter Zuhilfenahme Selbstorganisierender Karten strukturiert, im Display visualisiert und anschließend komfortabel gelabelt werden. Die so nach Objekten getrennten Bilddaten werden dem Klassifikator zum Erlernen übertragen. Für die Strukturierung der Daten wurden Bildmerkmale nach dem MPEG-7 Standard in der originalen oder in einer für die Aufgabe optimier-

ten Version verwendet. Diese entsprechen dabei grob zwei unterschiedlichen Kategorien, einerseits Farbmerkmalen und andererseits kanten- oder strukturbasierten Merkmalen. Die Gewichtung der Merkmale kann der Benutzer je nach Charakteristik der Bilddaten variieren. In der Arbeit wird an zwei unterschiedlichen Datensätzen aufgezeigt, dass es die Clusterung der Bilddaten durch die SOM ermöglicht, dass eine große Anzahl an Bildausschnitten in wenigen Schritten nach Objekten getrennt werden und interaktiv vom Benutzer gelabelt werden kann. Dabei erhöht die Möglichkeit, die Gewichtung der Merkmale zu variieren, die Trennungseffizienz der Bilddaten.

Für beide Arten des online-Lernens von Objekten wurde ein kognitiv motiviertes Lernverfahren entwickelt, bei dem einerseits, wie beim menschlichen Kurzzeitgedächtnis, Wissen über eine kleine Menge von Objekten nahezu unmittelbar zur Verfügung stehen kann und andererseits nach längerer Zeit eine Vielzahl von Objekten in einer Art Langzeitgedächtnis gespeichert werden kann.

Diese zwei Lerngeschwindigkeiten werden dabei von der in drei Verarbeitungsebenen unterteilten Architektur des verwendeten VPL-Klassifikators ermöglicht. Dazu wird beim schnellen Erlernen von Objektwissen nur die letzte Schicht neu trainiert, was zu einer geringeren Robustheit führt und wodurch die Erkennungperformanz nur bei einer geringen Anzahl von Objekten zufriedenstellend ist. Zu einem Zeitpunkt, an dem das System nicht ausgelastet ist und somit Ressourcen frei sind, findet das robustere und zeitaufwändigere Erlernen über alle Schichten statt. Wie sich die Performanz der beiden Verfahren bei variierender Objektanzahl im Hinblick auf die Rechenzeit verhält, wurde an Standarddatensätzen ermittelt.

Der Objekterkenner ist die Basis für das Wiederfinden von Objekten. Dazu enthält das System eine Vorstufe des VAMs in Form eines Bildergedächtnisses, in welchem die zuletzt wahrgenommenen Aufenthaltsorte jedes Objektes abgelegt werden und auf Nachfrage vom Benutzer visuell präsentiert werden können. Somit können verlegte Objekte vom Benutzer erfragt werden. Das Bild von dem gesuchten Objekt in seiner zuletzt vom System wahrgenommenen Position wird eingeblendet und ermöglicht so ein leichtes Wiederfinden.

19.2 Schlussfolgerung und Ausblick

Die Entwicklung von verschiedenen Systemen aus dem Bereich des sogenannten *wearable computing* ist zur Zeit eine der größten Herausforderungen in der Informationstechnologie. Die Leistungsfähigkeit der Systeme nimmt stetig zu, wobei die komfortable Bedienbarkeit und die Interaktion mit dem Benutzer immer noch in den Kinderschuhen stecken.

Diese Arbeit zeigt Wege auf, wie moderne Mensch-Maschine-Interaktion aussehen kann und wie man im Bereich des *wearable computing* in Zukunft die Systeme intelligenter machen kann. Bei der Entwicklung dieses Demonstrators konnte aufgezeigt werden, dass es prinzipiell möglich ist, so etwas wie eine „Brille mit Gedächtnis" zu entwickeln. Jedoch benötigt die Umsetzung dieser Funktionalität eine Integration von vielen verschiedenen Verfahren, welche teilweise noch nicht für die Anforderungen in mobilen Systemen zu-

friedenstellend entwickelt sind. Der Demonstrator bietet einige interessante Lösungen, wobei dennoch klar ist, dass es sich um ein System handelt, welches von der Serienreife noch weit entfernt ist.

Die größte Herausforderung ist dabei, dass künstliche Systeme bei weitem nicht die Fähigkeiten des menschlichen Sehsystems erreichen können. Deutlich wird dies bei der sinkenden Performanz des künstlichen visuellen Aufmerksamkeitssytems und des Erkenners bei stark variierenden Bedingungen. Das hier verwendete Aufmerksamkeitssystem verwendet Bildmerkmale, welche bei homogenen Hintergründen und bei nicht zu stark variierender Skalierung sehr gut funktionieren.

Die Wahl geeigneter Merkmale ist immer auch kontextabhängig. Dennoch ermöglichen einige Ansätze, wie die von Lowe 1999 entwickelten SIFT-Feature, eine weitgehende Unabhängigkeit im Hinblick auf variierende Skalierung, Translation und Rotation (Lowe, 1999b, 2004).

Die kontextabhängige Bedeutung verschiedener Bildmerkmale könnte perspektivisch zu einer sinnvollen Weiterentwicklung des SOM-Trainings führen, indem das Verfahren selbstständig aus dem Verhalten des Benutzers erlernt, wann welche Kategorie von Bildmerkmalen höher zu gewichten ist. Ergänzend könnten SOM-Verfahren verwendet werden, die automatisiert die Dimension der SOM mit Hilfe von aufgabenorientiert optimierten Abstandsmaßen bestimmen bzw. selbständig die Knotenanzahl des SOM-Gitters erhöhen.

Ein Nachteil des vorgestellten Verfahrens besteht sicherlich darin, dass der verwendete Klassifikator im Vergleich zu anderen Klassifikatoren eine geringere Performanz hat. Dieser Nachteil wird jedoch durch die Möglichkeit des „Simulierens" eines Kurzzeit- und eines Langzeitgedächtnisses zum Teil kompensiert.

Perspektivisch kann eine deutlich bessere Erkennungsleistung durch die Verwendung hierarchischer Feedforward-Netze erreicht werden. Angestrebt ist die Integration des von Ingo Bax im Rahmen des Projektes entwickelten Objekterkenners (Bax, 2007). Dieser integriert durch die Verwendung einer zusätzlichen Schicht ebenfalls eine robuste Ojektlokalisation. Im Vergleich zu dem hier vorgestellten System zeigt diese Art der Objektlokalisation besonders bei reich strukturierten Hintergründen und teilweiser Verdeckung der Objekte eine hohe Performanz. Das vorgestellte System ist dagegen in erster Linie für ein Szenario gut geeignet, in dem sich Objekte in einer vom Menschen erschaffenen Umgebung vor einheitlichen Hintergründen befinden. Auch gegenüber Rotation und der Variation der Skalierung erweist sich der Erkenner von Bax als sehr robust.

Dass in der vorgestellten Arbeit dennoch ein vom Klassifikator getrenntes Aufmerksamkeitssystem verwendet wird, lässt sich dadurch begründen, dass es zusätzlich für andere Aufgaben, wie der Lokalisation von Objekten für das iterative Labelverfahren, herangezogen werden kann. Diese Modularisierung der einzelnen Verarbeitungsschritte wäre eine Herausforderung für zukünftige Weiterentwicklungen der dem menschlichen Sehsystem nachempfundenen hierarchischen Netze. Dass dieses Feld noch immer offene Forschungsfragen enthält und nur durch die fächerübergreifende Verständigung zwischen Psychologen, Biologen und Informatikern zu neuen Erkenntnissen führen kann, zeigen Frintrop u. a. (2010). Sie versuchen durch ihren Überblick aus verschiedenen Perspekti-

ven auf das Feld der visuellen Aufmerksamkeit zu der Verständigung der verschiedenen Wissenschaftszweige beizutragen.

Eine Möglichkeit, die Klassifikationsleistungen künstlicher Systeme zu erhöhen, indem neurobiologische Kenntnisse zur Optimierung Computer gestützter Systeme verwendet werden, zeigen Bauckhage u. a. (2008) auf. Sie zeigen, dass eine Integration des Konzeptes des *Visual Active Memories* eine wesentliche Voraussetzung für die kognitiven Fähigkeiten künstlicher Systeme darstellt und erst dadurch Objekterkennung generalisierungsfähiger werden kann.

Die Trennung von Kurzzeit- und Langzeitgedächtnis des vorgestellten Systems ist ebenfalls dem biologischen Vorbild nachempfunden. Ein Manko dieses Systems ist es jedoch, dass für das komplette Training des Klassifikators für eine Art Langzeitgedächtnis die Verarbeitungsschritte des Kurzzeitgedächtnisses nicht verwendet werden. Kirstein u. a. (2008) entwickelten die hier vorgestellte Idee der Zweistufigkeit des Lernens weiter, indem sie ein online trainierbares Hierarchisches Feedforward-Objekterkennungsmodell konstruierten, bei dem auf den vom STM erstellten Repräsentationen von Objekten das LTM aufbaut. So kann prinzipiell auch für das insgesamt zeitaufwändigere Trainieren des LTMs Rechenzeit gespart werden.

Neben der Lernfähigkeit von Objektwissen hat diese Arbeit insbesondere in Bezug auf die natürliche Mensch-Maschine-Interaktion neue Wege aufgezeigt. Die hier präsentierte Idee, ein mobiles System zu entwickeln, bei dem die Interaktion zwischen Mensch und Maschine aus einer Kombination von Gestik und Sprache und durch die Verwendung eines AR-Systems in beide Richtungen möglich ist, wurde von Li und Jia (2010) aufgegriffen. Die Autoren verwenden ebenfalls ein mobiles AR-System zum Labeln von Objektdaten durch Kommunikation per Sprache und Gestik. Die Idee des virtuellen Menüs wurde von ihnen fortgeführt und um ein virtuelles *Touchpad* erweitert, welches über Zeigegesten bedienbar ist.

Die Idee der Multimodalität der Interaktion und der Menüführung eines AR-Systems wurde auch von Irawati u. a. (2006) aufgegriffen, die ein System entwickelten, mit dem Möbel virtuell im Raum positioniert werden können.

Die Auswertung von Gesten auf einem mobilen System stellt bei all diesen Forschungsbereichen immer noch eine Herausforderung dar. In der vorliegenden Arbeit wird die Objektreferenzierung durch eine Zeigegeste in 2-D ausgewertet. Es gibt bereits auf den hier präsentierten Ideen basierte Fortentwicklungen, bei denen ebenfalls 2-D Merkmalskarten verwendet wurden und die darüber hinaus durch die Kombination mit Disparitätsbasierten Auswertungen zu einer dreidimensionalen Interpretation der Zeigegeste kommen (Jia u. a., 2007)

Dass insbesondere die Forschung im Bereich der Augmented Reality stetig zunimmt, zeigen Dünser u. a. (2008) in einer Literaturrecherche zum Thema AR in den Datenbanken der führenden Verlage im Bereich der informatischen Forschung. Dünser u. a. (2008) zeigen, dass allein in den Jahren 1992 bis 2007 insgesamt 6071 Veröffentlichungen den Begriff der AR enthielten. Der Schwerpunkt der Recherche lag in der Untersuchung von Veröffentlichungen zum Thema der Benutzerevaluation. Die Studie zeigt, dass sich mit 8% nur ein sehr geringer Anteil der Veröffentlichungen der wichtigen Aufgabe stellte,

Benutzerevaluationen durchzuführen, wie es die vorliegende Arbeit macht (vgl. Heidemann u. a. (2004a)). Doch insbesondere die Bedienbarkeit solcher Systeme ist von hohem Stellenwert, wie auch Koelsch u. a. (2006) aufzeigen.

Natürliche Interaktion mit mobilen Systemen und die Verwendung von kognitiven Fähigkeiten in mobilen Systemen bietet noch ein breites Forschungsfeld für die Zukunft. Hierfür bietet die vorliegende Arbeit Ideen und Impulse zur Weiterentwicklung.

Literaturverzeichnis

[Aho u. a. 1986] Aho, Alfred V. ; Sethi, Ravi ; Ullman, Jeffrey D.: *Compilers: principles, techniques, and tools*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1986. – URL http://portal.acm.org/citation.cfm?id=6448. – ISBN 0201100886

[Anderson 2003a] Anderson, M.L.: Embodied cognition: A field guide. In: *Artificial Intelligence* 149 (2003), Nr. 1, S. 91–130

[Anderson 2003b] Anderson, M.L.: Representations, symbols and embodiment. In: *Artificial Intelligence* 149 (2003), Nr. 1, S. 151–6

[Azuma 1997a] Azuma, R.: Survey of Augmented Reality. In: *Teleoperators and Virtual Environments* 6 (1997), Nr. 4, S. 355–385

[Azuma 1997b] Azuma, Ronald T.: Course notes on Registration and Correcting for Dynamic Error from Course Note: Making Direct Manipulation Work in Virtual Reality. In: *ACM SIGGRAPH 97*. Los Angeles, USA, 1997, S. 3–8

[Backer u. a. 2001] Backer, G. ; Mertsching, B. ; Bollmann, M.: Data- and Model-Driven Gaze Control for an Active-Vision System. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23 (2001), Nr. 12, S. 1415–1429

[Baddeley 2003] Baddeley, Alan: WORKING MEMORY: LOOKING BACK AND LOOKING FORWARD. In: *Nat Rev Neurosci* 4 (2003), October, Nr. 10, S. 829–839. – URL http://dx.doi.org/10.1038/nrn1201

[Bajramovic u. a. 2005] Bajramovic, F. ; Grassl, C. ; Denzler, J.: Efficient Combination of Histograms for Real-Time Tracking Using Mean-Shift and Trust-Region Optimization, 2005, S. 254

[Bauckhage u. a. 2005] Bauckhage, C. ; Hanheide, M. ; Wrede, S. ; Käster, T. ; Pfeiffer, M. ; Sagerer, G.: Vision Systems with the Human in the Loop. In: *EURASIP J. on Applied Signal Processing* 2005 (2005), Nr. 14, S. 2375–2390

[Bauckhage u. a. 2008] Bauckhage, C. ; Wachsmuth, S. ; Hanheide, M. ; Wrede, S. ; Sagerer, G. ; Heidemann, G. ; Ritter, H.: The visual active memory perspective on integrated recognition systems. In: *Image Vision Comput.* 26 (2008), Nr. 1, S. 5–14. – ISSN 0262-8856

[Bax 2007] Bax, I.: *Hierarchical Feed-forward Models for Robust Object Recognition*, Department of Computer Science, Faculty of Technology, Bielefeld University, Dissertation, 2007

[Bax u. a. 2003] Bax, I. ; Bekel, H. ; Heidemann, G.: Recognition of Gestural Object Reference with Auditory Feedback. In: *Proc. Int'l Conf. Neural Networks*. Istanbul, Turkey, 2003, S. 425–432

[Bekel u. a. 2004] Bekel, H. ; Bax, I. ; Heidemann, G. ; Ritter, H.: Adaptive Computer Vision: Online Learning for Object Recognition. In: al., C. E. R. et (Hrsg.): *Proc. DAGM 2004*. Tübingen, Germany : Springer, 2004, S. 447–454

[Bekel u. a. 2005a] Bekel, H. ; Heidemann, G. ; Ritter, H.: Interactive Image Data Labeling Using Self-Organizing Maps in an Augmented Reality Scenario. In: *Neural Networks* 18 (2005), Nr. 5/6, S. 566–574

[Bekel u. a. 2005b] Bekel, H. ; Heidemann, G. ; Ritter, H.: SOM Based Image Data Structuring in an Augmented Reality Scenario. In: *Proc. Int'l Joint Conf. on Neural Networks*. Montréal, Québec, Canada, 2005, S. 3278–3283

[Belongie u. a. 2001] Belongie, Serge ; Malik, Jitendra ; Puzicha, Jan: Matching shapes. In: *In ICCV*, 2001, S. 454–461

[Berkeley 2004] Berkeley: *DBXML Sleepycat Software*. 2004. – http://www.sleepycat.com/products/xml.shtml

[Billinghurst und Poupyrev 2000] Billinghurst, Kato H. Kiyokawa K. Belcher D. ; Poupyrev, I.: Experiments with Face to Face Collaborative AR Interfaces. In: *Virtual Reality Journal* 4 (2000), Nr. 2

[Billinghurst und Kato 2000] Billinghurst, M. ; Kato, H.: Out and About: Real World Teleconferencing. In: *Britisch Telecom Technical Journal (BTTJ)* Millenium Edition (2000)

[Billinghurst und Kato 2001] Billinghurst, M. ; Kato, Kiyokawa K. Belcher D. Poupyrev I.: The MagicBook: A Transitional AR Interface. In: *Computer and Graphics* November (2001), S. 745–753

[Bruce und Morgan 1954] Bruce, V. ; Morgan, M.: Violations of Symmetry and Repetition in Visual Patterns. In: *Psychological Review* 61 (1954), S. 183–193

[Carmichael und Hebert 2002] Carmichael, Owen ; Hebert, Martial: Object recognition by a cascade of edge probes. In: *In British Machine Vision Conference*, 2002, S. 103–112

[Carpenter u. a. 1991] Carpenter, G. A. ; Grossberg, S. ; Rosen, D.: Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. In: *Neural Networks* 4 (1991), Nr. 6, S. 759–771

[Christensen 2003] Christensen, Henrik I.: Cognitive (Vision) Systems. In: *ERCIM News* 53 (2003), April, S. 17–18

[Christmas u. a. 2005] Christmas, W. ; Kostin, A. ; Yan, F. ; Kolonias, I. ; Kittler, J.: A system for the automatic annotation of tennis matches. In: *Fourth International Workshop on Content Based Multimedia Indexing.* Invited Paper (2005)

[Curtis u. a. 1998] Curtis, D. ; Mizell, D. ; Gruenbaum, P. ; Janin, A.: Several Devils in the Details: Making an AR App Work in the Airplane Factory. In: *1rst International Workshop on Augmented Reality (IWAR 98).* San Francisco, USA, 1998

[Deselaers u. a. 2004] Deselaers, Thomas ; Keysers, Daniel ; Ney, Hermann: Features for Image Retrieval: A Quantitative Comparison. In: *DAGM-Symposium*, 2004, S. 228–236

[Duncan 1992] Duncan, Humphreys G.: Beyond the search surface: Visual search and attentional engagement. In: *Experimental Psychology: Human Perception and Performance* 18 (1992), Nr. 2, S. 578–588

[Dünser u. a. 2008] Dünser, Andreas ; Grasset, Raphaël ; Billinghurst, Mark: A survey of evaluation techniques used in augmented reality studies. In: *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses.* New York, NY, USA : ACM, 2008, S. 1–27

[Eidenberger 2003] Eidenberger, H.: How good are visual MPEG–7 features. In: *Proceedings SPIE Visual Communications and Image PRocessing Conference.* Lugano, 2003, S. 5150:476–488

[Eidenberger 2004] Eidenberger, H.: Statistical analysis of content–based MPEG–7 descriptors for image retrieval. In: *ACM Multimedia Systems journal,Springer* 10 (2004), Nr. 2, S. 84–97

[Ephraim und Merhav 2002] Ephraim, Yariv ; Merhav, Neri: Hidden Markov processes. In: *IEEE Trans. Inform. Theory* 48 (2002), S. 1518–1569

[Fink 1999] Fink, G. A.: Developing HMM-based Recognizers with ESMERALDA. In: Matoušek, Václav (Hrsg.) ; Mautner, Pavel (Hrsg.) ; Ocelíková, Jana (Hrsg.) ; Sojka, Petr (Hrsg.): *Lecture Notes in Artificial Intelligence* Bd. 1692. Berlin Heidelberg : Springer, 1999, S. 229–234

[Fislage u. a. 1999] Fislage, M. ; Rae, R. ; Ritter, H.: Using Visual Attention to Recognize Human Pointing Gestures in Assembly Tasks. In: *7th IEEE Int'l Conf. on Computer Vision*, 1999

[Frintrop u. a. 2010] Frintrop, Simone ; Rome, Erich ; Christensen, Henrik I.: Computational visual attention systems and their cognitive foundations: A survey. In: *ACM Trans. Appl. Percept.* 7 (2010), Nr. 1, S. 1–39. – ISSN 1544-3558

[Fukushima 1980] Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. In: *Biol. Cybern.*, 1980, S. 36:193–202

[G. Kurtenbach 1990] G. Kurtenbach, E. A. H.: Gestures in Human-Computer Communication. In: *B. Laurel (Hrsg.): The Art of Human-Computer Interface Design* (1990), S. 309–317

[Gorges u. a. 2004] Gorges, N. ; Hanheide, M. ; Christmas, W. ; Bauckhage, C. ; Sagerer, G. ; Kittler, J.: Mosaics from Arbitrary Stereo Video Sequences. In: *Lecture Notes in Computer Science, vol. 3175, Heidelberg, Germany, Springer-Verlag (DAGM 2004)* (2004)

[Graham 1980] Graham, N.: Spatial frequency channels in vision: Detecting edges without edge detectors. In: *C. S. Harris: Visual Coding and Adaptability* Ed. Hillsdale, NJ: Erlbaum (1980), S. 215–252

[Gräßl u. a. 2005] Grässl, Christoph ; Zinsser, Timo ; Scholz, Ingo ; Niemann, Heinrich: 3-D Object Tracking with the Adaptive Hyperplane Approach Using SIFT Models for Initialization. In: *MVA*, 2005, S. 5–8

[Handmann u. a. 2000] Handmann, U. ; Kalinke, T. ; Tzomakas, C. ; Werner, M. ; Seelen, W. v.: An Image Processing System for Driver Assistance. In: *Image and Vision Computing* 18 (2000), Nr. 5, S. 367–376

[Hanheide 2006] Hanheide, Marc: *A Cognitive Ego-Vision System for Interactive Assistance*, Technische FakultÃ?t – UniversitÃ?t Bielefeld, phdthesis, dec 2006. – 198 S. – URL http://bieson.ub.uni-bielefeld.de/volltexte/2007/1032/

[Harris und Stephens 1988] Harris, C. ; Stephens, M.: A Combined Corner and Edge Detector. In: *Proc. 4th Alvey Vision Conf.*, 1988, S. 147–151

[Hebb 1949] Hebb, Donald O.: *The Organization of Behavior: A Neuropsychological Theory.* New York : Wiley, June 1949. – ISBN 0805843000

[Heidemann 1998a] Heidemann, G.: *Ein flexibel einsetzbares Objekterkennungssystem auf der Basis neuronaler Netze*, Department of Computer Science, Faculty of Technology, Bielefeld University, Dissertation, 1998

[Heidemann 1998b] Heidemann, G.: *Ein flexibel einsetzbares Objekterkennungssystem auf der Basis neuronaler Netze.* Technische Fakultät, Univ. Bielefeld, Dissertation, 1998. – Infix, DISKI 190

[Heidemann 2004a] Heidemann, G.: Combining spatial and colour information for content based image retrieval. In: *Computer Vision and Image Understanding, Special Issue on Colour for Image Indexing and Retrieval* 94 (2004), Nr. 1-3, S. 234–270

[Heidemann 2004b] Heidemann, G.: Focus-of-Attention from Local Color Symmetries. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26 (2004), Nr. 7, S. 817–830

[Heidemann u. a. 2004a] Heidemann, G. ; Bax, I. ; Bekel, H. ; Bauckhage, C. ; Wachsmuth, S. ; Fink, G. ; Pinz, A. ; Ritter, H. ; Sagerer, G.: Multimodal interaction in an augmented reality scenario. In: *Proc. Int'l Conf. Multimodal Interfaces ICMI 2004*, ACM Press, 2004, S. 53–60

[Heidemann u. a. 2005] Heidemann, G. ; Bekel, H. ; Bax, I. ; Ritter, H.: Interactive Online Learning. In: *Pattern Recognition and Image Analysis* 15 (2005), Nr. 1, S. 55–58

[Heidemann u. a. 2007] Heidemann, G. ; Bekel, H. ; Bax, I. ; Ritter, H.: Interactive Online Learning. In: *Pattern Recognition and Image Analysis* 17 (2007), Nr. 1, S. 146–152

[Heidemann u. a. 2004b] Heidemann, G. ; Bekel, H. ; Bax, I. ; Saalbach, A.: Hand Gesture Recognition: Self-Organising Maps as a Graphical User Interface for the Partitioning of Large Training Data Sets. In: Kittler, J. (Hrsg.) ; Petrou, M. (Hrsg.) ; Nixon, M. (Hrsg.): *Proc. ICPR 2004* Bd. 4. Cambridge, UK : IEEE-CS, 2004, S. 487–490

[Heidemann u. a. 2000] Heidemann, G. ; Lücke, D. ; Ritter, H.: A System for Various Visual Classification Tasks Based on Neural Networks. In: al., A. S. et (Hrsg.): *Proc. 15th Int'l Conf. on Pattern Recognition ICPR 2000, Barcelona* Bd. I, IEEE-CS, 2000, S. 9–12

[Heidemann und Ritter 2001] Heidemann, G. ; Ritter, H.: Efficient Vector Quantization Using the WTA-rule with Activity Equalization. In: *Neural Processing Letters* 13 (2001), Nr. 1, S. 17–30

[Heidemann und Ritter 2003] Heidemann, G. ; Ritter, H.: Learning to Recognise Objects and Situations to Control a Robot End-Effector. In: *KI Künstliche Intelligenz, special issue on Vision, Learning, Robotics* 2 (2003), S. 24–29

[Huang u. a. 2001] Huang, Xuedong ; Acero, Alex ; Hon, Hsiao-Wuen: *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, April 2001. – URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0130226165. – ISBN 0130226165

[Hubel und Wiesel 1959] Hubel, D. H. ; Wiesel, T. N.: Receptive fields of single neurons in the cat's striate cortex. In: *Journal of Physiology* 148 (1959), S. 574–591

[Hummel und Biederman 1992] Hummel, J. ; Biederman, I.: Dynamic binding in a neural network for shape recognition. In: *Psych. Rev.* 99 (1992), S. 480–517

[Irawati u. a. 2006] Irawati, Sylvia ; Green, Scott ; Billinghurst, Mark ; Duenser, Andreas ; Ko, Heedong: Move the couch where?": developing an augmented reality multimodal interface. In: *ISMAR '06: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Washington, DC, USA : IEEE Computer Society, 2006, S. 183–186. – ISBN 1-4244-0650-1

[Itti u. a. 1998] Itti, L. ; Koch, C. ; Niebur, E.: A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 20 (1998), Nr. 11, S. 1254–1259

[Jähne 1991] Jähne, B.: *Digital image processing*. Springer, 1991

[Jia u. a. 2007] Jia, Yunde ; Li, Shanqing ; Liu, Yang: Tracking pointing gesture in 3D space for wearable visual interfaces. In: *HCM '07: Proceedings of the international workshop on Human-centered multimedia*. New York, NY, USA : ACM, 2007, S. 23–30. – ISBN 978-1-59593-781-0

[Jolliffe 1986] Jolliffe, I.: *Principal Component Analysis*. New York : Springer Verlag, 1986

[Kalinke und Handmann 1997] Kalinke, T. ; Handmann, U.: Fusion of Texture and Contour Based Methods for Object Recognition. In: *IEEE Conf. on Intelligent Transportation Systems 1997*. Stuttgart, 1997

[Kalinke und von Seelen 1996] Kalinke, T. ; Seelen, W. von: Entropie als Maß des lokalen Informationsgehalts in Bildern zur Realisierung einer Aufmerksamkeitssteuerung. In: Jähne, B. (Hrsg.) ; Geissler, P. (Hrsg.) ; Haussecker, H. (Hrsg.) ; Hering, F. (Hrsg.): *Mustererkennung 1996*, Springer Verlag Heidelberg, 1996, S. 627–634

[Khan u. a. 2007] Khan, Saad M. ; Yan, Pingkun ; Shah, Mubarak: A homographic framework for the fusion of multi-view silhouettes. In: *In ICCV*, 2007

[Kirstein u. a. 2008] Kirstein, Stephan ; Wersing, Heiko ; Körner, Edgar: A biologically motivated visual memory architecture for online learning of objects. In: *Neural Netw.* 21 (2008), Nr. 1, S. 65–77. – ISSN 0893-6080

[Kiyokawa 2000] Kiyokawa, Takemura H. Yokoya N.: Seamless Design for 3D Object Creation. In: *IEEE Multi Media, ICMCS'99 Special Issue* 17 (2000), Nr. 1, S. 22–33

[Koelsch u. a. 2006] Koelsch, Mathias ; Bane, Ryan ; Hoellerer, Tobias ; Turk, Matthew: Multimodal Interaction with a Wearable Augmented Reality System. In: *IEEE Computer Graphics and Applications* 26 (2006), S. 62–71. – ISSN 0272-1716

[Koenderink und van Doorn 1979] Koenderink, J. J. ; Doorn, A. J. van: The internal representation of solid shape with respect to vision. In: *Biological Cybernetics* 32 (1979), S. 211–216

[Kohonen 1984] Kohonen, T.: Self-Organization and Associative Memory. In: *Springer Series in Information Sciences 8*. Springer-Verlag Heidelberg, 1984

[Kohonen 1995] Kohonen, T.: *Self-Organizing Maps*. Springer Verlag, 1995

[Laaksonen u. a. 2002] Laaksonen, J. ; Koskela, M. ; Oja, E.: PicSOM–Self-Organizing Image Retrieval With MPEG-7 Content Descriptors. In: *IEEE Transactions on Neural Networks* 13 (2002), Nr. 4, S. 841–853

[Li und Jia 2010] Li, Shanqing ; Jia, Yunde: A multimodal labeling interface for wearable computing. In: *IUI '10: Proceeding of the 14th international conference on Intelligent user interfaces*. New York, NY, USA : ACM, 2010, S. 345–348. – ISBN 978-1-60558-515-4

[Liu u. a. 2008] Liu, Zheng ; Genest, M. ; Marincak, A. ; Forsyth, David S.: Characterization of surface deformation with the Edge of Light^{TM} technique. In: *Mach. Vis. Appl.* 19 (2008), Nr. 1, S. 35–42

[Lömker 2004] Lömker, Frank: *Lernen von Objektbenennungen mit visuellen Prozessen*, Universität Bielefeld, Technische Fakultät, phdthesis, 2004. – URL http://bieson.ub.uni-bielefeld.de/volltexte/2004/549/

[Locher und Nodine 1987] Locher, P. J. ; Nodine, C. F.: Symmetry Catches the Eye. In: Levy-Schoen, A. (Hrsg.) ; O'Reagan, J. K. (Hrsg.): *Eye Movements: From Physiology to Cognition*. Elsevier Science Publishers B. V. (North Holland), 1987, S. 353–361

[Lowe 1999a] Lowe, David G.: Object recognition from local scale-invariant features, 1999, S. 1150–1157

[Lowe 1999b] Lowe, David G.: *Object Recognition from Local Scale-Invariant Features*. 1999

[Lowe 2004] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60 (2004), S. 91–110

[M. K. Chandraker und Pinz 2003] M. K. Chandraker, C. S. ; Pinz, A.: Real-Time Camera Pose in a Room. In: *Proc. Int'l Conf. Cognitive Vision Systems* Bd. 2626. Graz, Austria, 2003, S. 98–110

[M. Stoerring und Granum 2001] M. Stoerring, H. J. A. ; Granum, E.: Physics-based modelling of human skin colour under mixed illuminants. In: *Robotics and Autonomous Systems* 35 (2001), Nr. 3–4, S. 131–142

[Mahesh S. Raisinghani und Schmedding 2004] Mahesh S. Raisinghani, Jianchun Ding Maria Gomez Kanak Gupta Victor Gusila Daniel P. ; Schmedding, Oliver: Ambient Intelligence: Changing Forms of Human-Computer Interaction and their Social Implications. In: *Journal for Digital Information* 5 (2004), Nr. 4. – URL http://jodi.tamu.edu/Articles/v05/i04/Raisinghani/

[Makihara u. a. 2005] Makihara, Yasushi ; Miura, Jun ; Shirai, Yoshiaki ; Shimada, Nobutaka: Strategy for Displaying the Recognition Result in Interactive Vision. In: *CW '05: Proceedings of the 2005 International Conference on Cyberworlds*. Washington, DC, USA : IEEE Computer Society, 2005, S. 467–474. – ISBN 0-7695-2378-1

[Makihara u. a. 2004] Makihara, Yasushi ; Shirai, Yoshiaki ; Shimada, Nobutaka: Online Learning of Color Transformation for Interactive Object Recognition under Various Lighting Conditions. In: *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3*. Washington, DC, USA : IEEE Computer Society, 2004, S. 161–164. – ISBN 0-7695-2128-2

[Malamas u. a. 2003] Malamas, Elias N. ; Petrakis, Euripides G. M. ; Zervakis, Michalis E. ; Petit, Laurent ; Legat, Jean-Didier: A survey on industrial vision systems, applications, tools. In: *Image Vision Comput.* 21 (2003), Nr. 2, S. 171–188

[Manjunath u. a. 2001] Manjunath, B. S. ; Ohm, J.-R. ; Vasudevan, V. V. ; Yamada, A.: Color and Texture Descriptors. In: *IEEE Trans. on Circuits and Systems for Video Technology* 11 (2001), Nr. 6, S. 703–715

[Manjunath u. a. 2002] Manjunath, B. S. (Hrsg.) ; Salembier, Philippe (Hrsg.) ; Sikora, Thomas (Hrsg.): *Introduction to MPEG-7: Multimedia Content Description Language*. Wiley, April 2002. – URL http://www.amazon.com/Introduction-MPEG-Multimedia-Description-Language/dp/0471486787/sr=8-1/qid=1159379196/ref=pd_bbs_1/104-6549776-2426327?ie=UTF8&s=books

[Maquet 2001] Maquet, P.: The role of sleep in learning and memory. In: *Science* 294 (2001), November, Nr. 5544, S. 1048–1052. – URL http://dx.doi.org/10.1126/science.1062856. – ISSN 0036-8075

[Martinetz und Schulten 1994] Martinetz, T. ; Schulten, K.: Topology Representing Networks. In: *Neural Networks* 7 (1994), Nr. 3, S. 507–522

[McGuire u. a. 2002] McGuire, P. ; Fritsch, F. ; Steil, J. J. ; Röthling, F. ; Fink, G. A. ; Wachsmuth, S. ; Sagerer, G. ; Ritter, H.: Multi-Modal Human-Machine Communication for Instructing Robot Grasping Tasks. In: *Proc. IROS 2002*, 2002. – Accepted

[McNeil 1992] McNeil, D.: *Hand and Mind:What Gestures Reveal about Thought*. Chicago IL : University of Chicago Press, 1992

[Mel 1997] Mel, B. W.: SEEMORE: Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. In: *Neural Computation* 9 (1997), S. 777–804

[Messer u. a. 2005] Messer, K. ; Christmas, W. ; Jaser, E. ; Kittler, J. ; Levienaise-Obadia, B. ; Koubaroulis, D.: A unified approach to the generation of semantic cues for sports video annotation. In: *Signal Processing* Special issue on Content Based Image and Video Retrieval (2005), Nr. 83, S. 357–383

[Mian u. a. 2006] Mian, Ajmal S. ; Bennamoun, Mohammed ; Owens, Robyn: Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2006), Nr. 10, S. 1584–1601. – ISSN 0162-8828

[Moody und Darken 1988] Moody, J. ; Darken, C.: Learning with Localized Receptive Fields. In: *Proc. of the 1988 Connectionist Models Summer School*. San Mateo, CA : Morgan Kaufman Publishers, 1988, S. 133–143

[Murase und Nayar 1995] Murase, H. ; Nayar, S. K.: Visual Learning and Recognition of 3-D Objects from Appearance. In: *Int'l J. of Computer Vision* 14 (1995), S. 5–24

[Nene u. a. 1996a] Nene, S. A. ; Nayar, S. K. ; Murase, H.: Columbia Object Image Library: COIL-100 / Dept. Computer Science, Columbia Univ. 1996 (CUCS-006-96). – Forschungsbericht

[Nene u. a. 1996b] Nene, S. A. ; Nayar, S. K. ; Murase, H.: Columbia Object Image Library (COIL-20) / Dept. Computer Science, Columbia Univ. New York, N.Y. 10027. 1996 (CUCUS-006-96). – Forschungsbericht

[Norman und O'reilly 2003] Norman, Kenneth A. ; O'reilly, All C.: Modeling hippocampal and neocortical contributions to recognition memory: A complementary learning systems approach. In: *Psychological Review* 110 (2003), S. 611–646

[Oja u. a. 1999] Oja, Merja ; Kaski, Samuel ; Kohonen, Teuvo: *Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum*. 1999. – URL citeseer.ist.psu.edu/683884.html

[Oviatt und Cohen 2000] Oviatt, S. ; Cohen, P.: Multimodal Interfaces That Process What Comes Naturally. In: *Communications of the ACM* 43 (2000), Nr. 3, S. 45–53

[Park u. a. 2000] Park, Dong K. ; Jeon, Yoon S. ; Won, Chee S.: Efficient use of local edge histogram descriptor. In: *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*. New York, NY, USA : ACM, 2000, S. 51–54. – ISBN 1-58113-311-1

[Privitera und Stark 2000] Privitera, C. M. ; Stark, L. W.: Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22 (2000), Nr. 9, S. 970–982

[Quek 1995] Quek, F. K. H.: Eyes In The Interface. In: *Image and Vision Computing* 13 (1995), Nr. 6, S. 511–525

[Quek u. a. 2002] Quek, Francis ; McNeill, David ; Bryll, Robert ; Duncan, Susan ; Ma, Xin-Feng ; Kirbas, Cemil ; McCullough, Karl E. ; Ansari, Rashid: Multimodal human discourse: gesture and speech. In: *ACM Trans. Comput.-Hum. Interact.* 9 (2002), Nr. 3, S. 171–193. – ISSN 1073-0516

[Rae 2000] Rae, R.: *Gestikbasierte Mensch-Maschine-Kommunikation auf der Grundlage visueller Aufmerksamkeit und Adaptivität.* Technische Fakultät, Univ. Bielefeld, Dissertation, 2000

[Reisfeld u. a. 1995] Reisfeld, D. ; Wolfson, H. ; Yeshurun, Y.: Context-Free Attentional Operators: The Generalized Symmetry Transform. In: *Int'l J. of Computer Vision* 14 (1995), S. 119–130

[Reitmayr und Schmalstieg 2004] Reitmayr, Gerhard ; Schmalstieg, Dieter: Collaborative augmented reality for outdoor navigation and information browsing. In: *In Proceedings of the Symposium on Location Based Services and TeleCartography*, Wiley, 2004, S. 31–41

[Riesenhuber und Poggio 2000a] Riesenhuber, Maximilian ; Poggio, Tomaso: Cbf: A new framework for object categorization in cortex. In: *1st IEEE Int. Worksh. Biologically Motivated Computer Vision, Seoul (Korea*, Springer-Verlag, 2000, S. 1–9

[Riesenhuber und Poggio 2000b] Riesenhuber, Maximilian ; Poggio, Tomaso: Computational models of object recognition in cortex: A review / and 190, Artificial Intelligence Laboratory and Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology. 2000. – Forschungsbericht

[Ritter u. a. 1992] Ritter, H. J. ; Martinetz, T. M. ; Schulten, K. J.: *Neuronale Netze.* München : Addison-Wesley, 1992. – 258 S

[Saalbach 2001] Saalbach, A.: *Self-Organizing Maps zur halbautomatischen Erzeugung datennaher Klasseneinteilungen.* Technische Fakultät, Univ. Bielefeld, Diplomarbeit, 2001

[Sanger 1989] Sanger, T. D.: Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network. In: *Neural Networks* 2 (1989), S. 459–473

[Schmalsteig 1996] Schmalsteig, Fuhrmann A. Szalavari Z. Gervautz M.: Studierstube - An Environment for Collaboration in Augmented Reality. In: *CVE 96 Workshop Proceedings*, 1996

[Schmid u. a. 2000] Schmid, C. ; Mohr, R. ; Bauckhage, C.: Evaluation of Interest Point Detectors. In: *Int'l J. of Computer Vision* 37 (2000), Nr. 2, S. 151–172

[Schneider 1977] Schneider, R. M.: Controlled and automatic human information processing: I. Detection, search, and attention. In: *Psychological Review 84* (1977), S. 1–66

[Schwald u. a. 2003] Schwald, Bernd ; Laval, Blandine D. ; Sa, Thales O. ; Guynemer, Rue: An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context. In: *In The 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2003, Plzen, Czech Republic*, 2003, S. 425–432

[Shannon 1948] Shannon, C. E.: A Mathematical Theory of Communication. In: *Bell Systems Technical J.* 27 (1948), S. 379–423

[Sheperd 1986] Sheperd, Findlay J. M.; Hockey R. J.: The relationship between eye movements and spatial attention. In: *The Quarterly J. Experimental Psychology* 38A (1986), S. 475–491

[Sikora 2001] Sikora, Thomas: The MPEG-7 visual standard for content description-an overview. In: *IEEE Trans. Circuits Syst. Video Techn.* 11 (2001), Nr. 6, S. 696–702

[Somervuo und Kohonen 1999] Somervuo, P. ; Kohonen, T.: Self-Organizing Maps and Learning Vector Quantization for Feature Sequences. In: *Neural Processing Letters* 10 (1999), Nr. 2, S. 151–159

[Soriano u. a. 2000] Soriano, M. ; Martinkauppi, B. ; Huovinen, S. ; Laaksonen, M.: Skin detection in video under changing illumination conditions. In: *Proc. CVPR 2000* Bd. 1, 2000, S. 839–842

[Steels und Kaplan 2000] Steels, Luc ; Kaplan, Frederic: AIBO's first words: The social learning of language and meaning. In: *Evolution of Communication* 4 (2000), Nr. 1, S. 3–32. – URL http://www3.isrl.uiuc.edu/~junwang4/langev/localcopy/pdf/steels02aiboFirst.pdf

[Stiedl 2006] Stiedl, Thomas: Framework für multimodale Bediensysteme in der Automatisierungstechnik. In: *GI Jahrestagung (2)*, 2006, S. 13–20

[Tipping und Bishop 1999] Tipping, M. E. ; Bishop, C. M.: Mixtures of Probabilistic Principal Component Analyzers. In: *Neural Computation* 11 (1999), Nr. 2, S. 443–482

[Tokutaka u. a. 1999] Tokutaka, H. ; Yoshihara, K. ; Fujimura, K. ; Iwamoto, K. ; Obu-Cann, K.: Application of self-organizing maps (SOM) to Auger electron spectroscopy (AES). In: *Surface and Interface Analysis* 27 (1999), Nr. 8, S. 783–788

[Treisman 1982] Treisman, A.: Perceptual grouping and attention in visual search for features and for objects. In: *Journal of Experimental Psychology: Human Perception and Performance* 8 (1982), S. 194–214

[Treisman 1992] Treisman, A.: Spreading suppression or feature integration: A reply to Duncan and Humphreys (1992). In: *Experimental Psychology: Human Perception and Performance* 18 (1992), Nr. 2, S. 589–593

[Treisman 1980] Treisman, Gelade G.: A feature-integration theory of attention. In: *Cognitive Psychology 12* (1980), S. 97–136

[Treisman 1988] Treisman, Gormican S.: Feature analysis in early vision: Evidence from search asymmetries. In: *Psychological Review* 95 (1988), Nr. 1, S. 15–48

[Turk und Pentland 1991] Turk, M. ; Pentland, A.: Eigenfaces for Recognition. In: *J. Cognitive Neuroscience* 3 (1991), S. 71–86

[VAMPIRE Consortium 2002–5] VAMPIRE Consortium: *Visual Active Memory Processes and Interactive Retrieval*. 2002-5. – URL http://www.vampire-project.org.IST-2001-34401

[Vernon 2005] Vernon, David: Cognitive Vision - The Development of a Discipline. In: *KI-Zeitschrift Künstliche Intelligenz* Special Issue on Cognitive Computer Vision (2005), S. 38–41

[Vernon 2008] Vernon, David: Cognitive Vision - The Case for Embodied Perception. In: *Image and Vision Computing* 26,Special Issue on Cognitive Vision (2008), Nr. 1, S. 127–141

[Vesanto 1999] Vesanto, Juha: SOM-Based Data Visualization Methods. In: *Intelligent Data Analysis* 3 (1999), Nr. 2, S. 111–126

[Villmann u. a. 2003] Villmann, T. ; Merenyi, E. ; Hammer, B.: *Neural maps in remote sensing image analysis*. 2003. – URL citeseer.ist.psu.edu/villmann03neural.html

[Viola und Jones 2001] Viola, Paul ; Jones, Michael: Robust Real-time Object Detection. In: *International Journal of Computer Vision*, 2001

[Wachsmuth u. a. 2000] Wachsmuth, S. ; Fink, G. A. ; Kummert, F. ; Sagerer, G.: Using Speech in Visual Object Recognition. In: Sommer, G. (Hrsg.) ; Krüger, N. (Hrsg.) ; Perwass, C. (Hrsg.): *Mustererkennung 2000, 22. DAGM-Symposium Kiel*, Springer, 2000 (Informatik Aktuell), S. 428–435

[Walther u. a. 2002a] Walther, D. ; Itti, L. ; Riesenhuber, M. ; Poggio, T. ; Koch, C.: Attentional Selection for Object Recognition – a Gentle Way. In: *Proc. 2nd Workshop on Biologically Motivated Computer Vision (BMCV'02)*. Tübingen, Germany, 2002

[Walther u. a. 2002b] Walther, D. ; Itti, L. ; Riesenhuber, M. ; Poggio, T. ; Koch, C.: Attentional Selection for Object Recognition - a Gentle Way. In: *Proc. 2nd Workshop on Biologically Motivated Computer Vision (BMCV'02), Tuebingen, Germany*, 2002

[Wersing und Körner 2003] Wersing, Heiko ; Körner, Edgar: Learning optimized features for hierarchical models of invariant object recognition. In: *Neural Comput.* 15 (2003), Nr. 7, S. 1559–1588. – ISSN 0899-7667

[Wexelblat 1998] Wexelblat, Alan: Research Challenges in Gesture: Open Issues and Unsolved Problems. In: *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*. London, UK : Springer-Verlag, 1998, S. 1–11. – ISBN 3-540-64424-5

[Wisneski u. a. 1998] Wisneski, Craig ; Ishii, Hiroshi ; Dahley, Andrew ; Gorbet, Matt ; Brave, Scott ; Ullmer, Brygg ; Yarin, Paul: Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information. In: *Lecture Notes in Computer Science* 1370 (1998), S. 22–??. – URL citeseer.ist.psu.edu/wisneski98ambient.html

[Ziemke 2005] Ziemke, T.: Cybernetics and Embodied Cognition: On the Construction of Realities in Organisms and Robots. In: *Kybernetes* 34 (2005), Nr. 1/2, S. 118–128