Universität Karlsruhe (TH)

Proaktive Initiierung von Dialogen für humanoide Roboter

Diplomarbeit am Institut für Theoretische Informatik
Prof. Dr. Alex Waibel
Fakultät für Informatik
Universität Karlsruhe (TH)

von

Christoph Schaa

Betreuer:
Prof. Dr. Alex Waibel
Dipl.-Inform. Hartwig Holzapfel
Dipl.-Inform. Kai Nickel

September 2005

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 14.September 2005 .....

Kurzfassung

Roboter werden uns zukünftig vermehrt im alltäglichen Leben begegnen. Gerade die Entwicklung sogenannter humanoider Roboter soll dabei einen natürlichen Umgang mit diesen Systemen ermöglichen. Solche Roboter werden vor allem kommunikative Aufgaben übernehmen, welche momentan noch von Menschen ausgeführt werden. Dabei gibt es viele Anwendungsmöglichkeiten bei denen es wichtig ist, dass der Roboter nicht nur auf Personen reagiert, von welchen er angesprochen wird, sondern dass er auch selbständig einen Dialog beginnen kann.
Daher sollte in dieser Arbeit ein System erstellt werden, welches dies ermöglicht. Ausgehend von den Beobachtungen, welche der Roboter mit einer Stereo-Kamera macht, soll entschieden werden, ob und wie das System auf einen Benutzer reagiert. Dabei wird ein proaktives Vorgehen angewendet. Der Roboter versucht einen Dialog zu beginnen, ohne dass der Benutzer explizit seinen Wunsch dazu ausgedrückt hat. Dazu ist es notwendig das Interesse einer Person zu erwecken und ihre Aufmerksamkeit auf den Roboter zu lenken. Aus diesem Grund wurden Experimente (Kapitel 5) durchgeführt, in denen die Wirkung verschiedener Roboteraktionen auf einen Benutzer untersucht wurden. Nachdem der Roboter das Interesse einer Person erweckt hat, führt er weitere Aktionen aus um einen Dialog zu initiieren. Die Beurteilung, ob eine Person interessiert ist, beruht dabei auf den Beobachtungen einer Stereokamera. Ausgehend von den Bewegungen eines Benutzers wird sein Interesse klassifiziert. Dabei werden Personen in zwei Klassen aufgeteilt. Erstens Personen, die ein mögliches Interesse am Roboter haben und zweitens Personen, die nicht am Roboter interessiert sind.

Abstract

This thesis describes a System, that is able to proactively initiate a human-robot dialog. Therefore the robot uses special initial actions to gain the attention of a person. By observing his environment with a stereo camera the robot is able to detect people and to decide whether they are interessted in a dialog or not. Depending on this decision it uses further actions to lead the user into a dialog. Experiments were made to find out, how the attention of a person could be gained and how the system interacts with a user. The Experiments showed that 44% of the persons passing by the robot were lead into a dialog.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

1.1 Motivation

Der Einsatz von Robotern in der Industrie ist schon lange weit verbreitet. Hierbei führen die hochspezialisierten Roboter Arbeiten aus, welche sie sicherer und schneller als jeder Mensch bewältigen. In diesem Bereich erfolgt die Steuerung der Roboter auf eine Weise, welche der technischen Natur des Systems angepasst ist. D.h sie werden mit Hilfe einer Tastatur und eines Monitors programmiert und gesteuert. In Zukunft werden Roboter auch vermehrt in anderen Gebieten eingesetzt und so mehr und mehr im alltäglichen Leben Anwendung finden. Dabei werden, im Gegensatz zur Industrie, besonders multifunktionale Systeme interessant sein, welche in der Lage sind, eine Vielzahl an Aufgaben zu bewältigen. Viele solcher Aufgaben können Roboter erst durch eine Interaktion mit Menschen bewältigen. Oft ist diese Interaktion sogar die eigentliche Aufgabe, zum Beispiel indem ein Roboter Menschen unterhält oder informiert. In diesen Fällen eignet sich eine spezielle Form von Robotern besonders gut, die sogenannten „humanoiden" Roboter. Diese Systeme sind dem Menschen nachempfunden, um dem Benutzer einen möglichst natürlichen, menschlichen Umgang mit dem Roboter zu ermöglichen. Neben der optischen und motorischen Ähnlichkeit, die sie zum Menschen aufweisen, müssen humanoide Roboter auch sensorisch für eine Interaktion mit Menschen ausgerüstet sein. Für eine räumliche Orientierung eignen sich zum Beispiel Laser- und Ultraschallscanner oder auch Videosysteme. Solche Videosysteme sind in der Lage, komplexe Objekte zu erkennen, wie zum Beispiel Gesichter. Es ist damit auch möglich, die Geschichtsausdrücke und Gesten einer Person zu erfassen. Für die Steuerung humanoider Roboter eignet sich ein verbaler Dialog. Zusätzlich können Roboter versuchen auch nicht-verbale Aspekte eines Dialoges zu erfassen, wie zum Beispiel Gesten, Geräusche (Lachen, Husten, ...) oder auch die Bewegung eines Benutzers (entfernt er sich vom Roboter oder nähert er sich).
Da der Dialog mit einem Benutzer für einen Großteil der Anwendungsgebiete eines humanoiden Roboters eine entscheidende Rolle spielt, ist die Überlegung, wie ein solcher Dialog überhaupt Zustande kommt, nicht unwichtig. Dabei gibt es zwei Ausgangssitationen. Zum einen geht die Initiative zum Dialog vom Benutzer aus. Das kann zum Beispiel durch „Werbung" erreicht werden. Das heißt mit Hilfe von Schildern oder einer Nachricht auf einem Bildschirm wird der mögliche Benutzer darauf hingewiesen, dass das System ihm einen Dienst anbieten kann. Das Interesse des Benutzers kann aber auch einfach durch

die Anwesenheit des Roboters geweckt werden. Zusätzlich muss dem Benutzer dann noch mitgeteilt werden, wie er das System nutzen kann. Dazu muss ihm zumindest mitgeteilt werden, wie er den Dialog mit dem System beginnen kann. Zum Beispiel indem er ein Headset aufsetzt und den Roboter begrüßt. Diese beiden Aufgaben können auch von einem menschlichen Betreuer übernommen werden. Diese Vorgehensweise hat zum einen den Vorteil, dass gezielt Benutzer ausgewählt werden können. Damit kann der Betreuer im Sinne des Roboters einen Dialog initiieren. Zum anderen kann der Betreuer den Benutzer weiter beobachten und diesem, falls er Probleme hat, die Nutzung des Systems noch mal oder auch genauer erklären. Neben diesen Vorteilen geht natürlich die Autonomie des Systems verloren. Daher scheint es naheliegend, die Aufgaben eines solchen Betreuers oder zumindest einen Teil seiner Aufgaben zu automatisieren und in das System zu integrieren. Dazu muss der Roboter in der Lage sein, mögliche Benutzer zu erfassen und je nach Aufgabenstellung zu entscheiden, ob er mit diesen Personen einen Dialog beginnen soll. Falls ein Dialog begonnen werden soll, muss der Roboter die Aufmerksamkeit des Benutzers erringen und diesem klar machen, was seine Absicht ist. Wenn er erkennt, dass ein Benutzer gegangen ist, sollte er nicht lange auf eine weitere Eingabe warten und sich sofort auf den nächsten Benutzer einstellen.
Ein mögliches Szenario, in dem ein solches System eingesetzt werden kann, sieht folgendermaßen aus. Der Roboter dient als Informationsstand (KIOSK) bei einer Ausstellung, spricht dabei die Benutzer selbständig an und erklärt seine Funktion. Ein solcher Roboter könnte auch dazu verwendet werden, Personen in einem Eingangs- oder Durchgangsbereich anzusprechen, ihren Namen und ggf. zusätzliche Informationen zu erfragen und ein Bild des Gesichts zu speichern. Somit ließe sich in einem Unternehmen automatisch eine Bilddatenbank der Mitarbeiter oder Besucher erstellen. Wenn das Videosystem Gesichter erkennen kann und ihnen eine Personenkennung (PersonID) zuordnet, wäre der Roboter in der Lage, nur unbekannte Personen anzusprechen. Damit würden Personen, die oft am Roboter vorbeikommen, wie zum Beispiel ein Mitarbeiter der sein Büro neben dem Roboter hat, nicht belästigt werden. Außerdem wäre es dann möglich, Personen gezielt anzusprechen und ihnen Nachrichten zu übermitteln.

1.2 Anforderungen

Die Aufgabe lässt sich in mehrere Schritte gliedern. Zuerst müssen die Personen in der Umgebung des Roboters mit Hilfe einer Videokamera erfasst und verfolgt werden. Im nächsten Schritt wird entschieden, ob und wie der Roboter auf diese Person reagiert. Danach soll mit Hilfe der richtigen Aktionen und des richtigen Timings dieser Aktionen der Benutzer in einen Dialog geführt werden. Als spezielle Herausforderungen an ein solches System ergibt sich vor allem die Geschwindigkeit und Fehleranfälligkeit der Personverfolgung.
Da Personen anhand des Gesichtes erfasst werden, werden sie nur erfasst solange sie ungefähr in die Richtung des Roboters sehen. Dadurch ist eine lückenlose Verfolgung nicht immer möglich. Daraus ergibt sich auch, dass eine Person, deren Gesicht eine Zeit verdeckt ist, danach nicht mehr als die gleiche Person erkannt, sondern als eine neu hinzugekommene betrachtet wird. Dieser Effekt kann durch schlechte Lichtverhältnisse und eine zu geringe Geschwindigkeit des Systems verstärkt werden. Wenn das System zum Beispiel mit nur einem Bild pro Sekunde arbeitet, kann es vorkommen, dass eine Personen, die am Roboter vorbeigeht, nur ein oder zweimal erfasst wird. Bei ungünstigen Lichtverhältnissen ist es möglich, dass Personen gar nicht erfasst werden. Außerdem wird durch

die Lichtverhältnisse die Quote der „Fehlerkennungen“ beeinflusst. Das heißt es werden Gesichter an Stellen erkannt, an denen keine sind. Um mit diesen Gegebenheiten zurecht zu kommen, sollte das System einerseits die Fehler in der Personenverfolgung minimieren und andererseits sollte es möglichst robust gegenüber solchen Fehlern sein.
Mit Hilfe der von der Personenverfolgung gelieferten Informationen soll entschieden werden, ob und wann eine Aktion für einen Benutzer durchgeführt werden soll. Das System muss dafür mit wenigen Informationen auskommen. Das Videosystem ist zum Beispiel nicht in der Lage, den Gesichtsausdruck des Menschen zu erkennen, beziehungsweise zu interpretieren. Auch für eine Erfassung der Blickrichtung, welche in [Kuno u. a. (2004)] verwendet wird, reicht die Qualität vieler Videosysteme nicht aus. Es lässt sich aber die Position eines Benutzers bestimmen und verfolgen. Allerdings muss das System dabei die schon aufgeführten Fehler einer solchen Personenverfolgung berücksichtigen und möglichst unabhängig von diesen funktionieren.

1.3 Zielsetzung

Zielsetzung der Arbeit ist es, ein in Echtzeit funktionierendes System zu entwickeln, welches in der Lage ist, proaktiv einen Dialog mit einem Benutzer zu initiieren. Zusätzlich sollten zu mehreren Zeitpunkten während der Entwicklung des Systems Experimente durchgeführt werden. Ziel dieser Experimente war es, Teile des Systems zu untersuchen, um die daraus gewonnenen Erkenntnisse in die weitere Entwicklung einzubringen.

Vorgaben bei der Umsetzung dieser Aufgabe gab es in Bezug auf die verwendeten Sensoren. So sollte das System lediglich auf die Stereokamera angewiesen sein. Damit lässt es sich zum Beispiel auch auf einem KIOSK-System verwenden, an welchem eine Stereokamera ohne großen Aufwand angebracht werden kann (wenn sie nicht schon vorhanden ist), wohingegen Sonarsensoren oder Laserscanner nicht so einfach nachzurüsten sind. Im Rahmen dieser Arbeit war es nicht vorgesehen, den Roboter im Raum zu bewegen. Daher würden sich bei einem Einsatz auf einem KIOSK-System keine Einschränkungen ergeben. Für den eigentlichen Dialog ist ein Headset vorgesehen. Allerdings ist es auch möglich, fest am Roboter angebrachte Mikrofone zu verwenden. In diesem Zusammenhang wurde ein Experiment mit einer ebenfalls an diesem Institut entwickelten Sprecherlokalisation [Walliczek (2005)] durchgeführt. Ziel dieses Experiments war allerdings lediglich die Lokalisierung von Sprache und nicht die Spracherkennung und der eigentlichen Dialog. Eine weitere Vorgabe war, wie schon erwähnt, die Echtzeitfähigkeit des Systems, wodurch die Experimente zur Interaktion erst möglich werden. Außerdem sollte das System in der Lage sein, mit mehreren Benutzern zurechtzukommen. Für die videobasierte Personenverfolgung und das Dialogsystem war es vorgesehen, bestehende Systeme zu verwenden, und diese so weit wie nötig anzupassen.

1.4 Gliederung

Kapitel 2 führt mehrere verwandte Arbeiten auf, welche ebenfalls videobasiert arbeiten und dabei die Interaktion mit einem Benutzer als Ziel haben. Im darauffolgenden Kapitel 3 werden die für diese Arbeit wichtigen Grundlagen erläutert. Danach werden die zugrundeliegenden Konzepte und deren Umsetzung im Kapitel 4 beschrieben. Kapitel 5 enthält

die Beschreibung der durchgeführten Experimente und Kapitel 6 die dazugehörigen Ergebnisse sowie die Evaluation. Am Ende (Kapitel 7) kommt noch mal eine Zusammenfassung der Arbeit und ein Ausblick darauf, welche Möglichkeiten sich aus dieser Arbeit ergeben.

2. Verwandte Arbeiten

In diesem Kapitel sollen ähnliche Arbeiten oder Arbeiten im selben Bereich vorgestellt und falls möglich auch verglichen werden. Zu Beginn werden daher drei Arbeiten vorgestellt, welche sich mit integrierten Robotersystemen befassen. Die anschließend aufgeführte Arbeit befasst sich dahingegen mit einem KIOSK-System. Ein Vergleich ist trotzdem angebracht, da dieses System im Gegensatz zu den drei vorherigen zur Erfassung seiner Umwelt ausschließlich ein Videosystem verwendet. Daher lässt sich hier die videobasierte Erfassung von Personen genauer vergleichen. Zum Ende des Kapitels soll kurz der Bereich des proaktiven Handelns bei der Mensch-Roboter-Interaktion angeführt werden, da dies auch ein Teilaspekt des hier vorgestellten Systems ist.

2.1 Integrierte Robotersysteme

Im folgenden werden verschiedene integrierte Robotersysteme vorgestellt. Neben den Besonderheiten der einzelnen Systeme werden dabei auch allgemeine Merkmale solcher Systeme, wie die Sensoren eines Roboters genauer betrachtet.

2.1.1 Multimodale Aufmerksamkeitssteuerung für mobile Roboter

Ein mobiles Robotersystem stellt Sebastian Lang in seiner Dissertation [Lang (2005)] vor. Der Roboter soll hier als Serviceroboter dienen. Dabei ist eine möglichst natürliche und intuitive Interaktion mit dem Roboter wünschenswert, so dass der Benutzer ohne ein umständliches Training in der Lage ist, den Roboter zu bedienen. Dazu muss das eingesetzte System menschliches, kommunikatives Verhalten nachbilden. Zum einen spielt dabei ein natürlichsprachlicher Dialog eine wichtige Rolle und zum anderen das Aufmerksamkeitsverhalten. Dieses Aufmerksamkeitsverhalten muss der Roboter bei einem Benutzer erkennen können, d.h. er muss wissen wann und von wem er angesprochen wird. Er muss dann aber auch seine eigene Aufmerksamkeit zum Ausdruck bringen, so dass der Benutzer weiß der Roboter hört zu. Um dies zu erreichen, beobachtet der Roboter seine Umgebung und reagiert, sobald er sich angesprochen fühlt, auf geeignete Weise.

2.1.1.1 Robotersystem

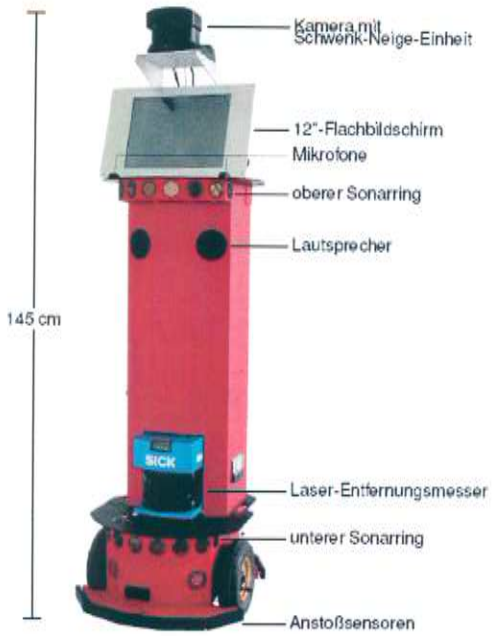

Abbildung 2.1: BIRON (**Bi**elefeld **R**bot Compani**on**)

Anhand des hier eingesetzten Robotersystems sollen die im Bereich der Serviceroboter verbreitetsten Sensoren beschrieben werden. Der für dieses System verwendete Roboter trägt den Namen BIRON (**BI**elefeld **R**obot Compani**ON**) (siehe Abbildung 2.1.1.1). Er baut auf dem Robotermodell PeopleBot der Firma ActiveMedia Robotics auf und ist mit zusätzlichen Komponenten erweitert worden. So wurde auf die obere Plattform ein Gestell für einen Flachbildschirm und eine Kamera montiert. Durch seine hohe und schlanke Bauweise ist der Roboter besonders für den Umgang mit stehenden Menschen geeignet. Die Fortbewegung funktioniert über zwei seitliche Antriebsräder, welche für ebene Untergründe geeignet sind. Kleine Unebenheiten wie Türschwellen und Teppichkanten lassen sich aber auch überwinden. Der Roboter besitzt fünf Arten von Sensoren:

1. Anstoßsensoren:

Sie reagieren auf Kontakt und können den Roboter bei einer Kollision mit einem Hindernis stoppen. Sie sind an der unteren Kante angebracht und dienen als Sicherheitsvorkehrung, um Gegenstände erkennen, welche von den anderen Sensoren nicht erfasst werden.

2. Sonarsensoren:

Mit Hilfe der Sonarsensoren lässt sich der Abstand zu Objekten in der Umgebung messen. Dazu befinden sich in 20cm und 100cm Höhe jeweils acht nebeneinander angeordnete Sonarsensoren, welche ein gerichtetes Ultraschallsignal aussenden. Durch die Laufzeit des Signals beziehungsweise durch die Zeit, die bis zum Eintreffen der Reflektion vergeht, und dem Abstrahlwinkel des Sensors, lässt sich der Abstand und die Richtung eines Objektes bestimmen. Die Erfassung ist dabei auf die Ebene, in der die Sensoren angebracht sind, beschränkt. Durch die geringe Anzahl von Sensoren in einer Reihe wird nur ei-

ne sehr geringe Winkelauflösung erreicht. Daher sind diese Sensoren vor allem für die Kollisionsvermeidung geeignet. Da dies in dem hier vorgegebenen Interaktionsszenario eine untergeordnete Rolle spielt, werden sie bei diesem System nicht verwendet und abgeschaltet. Der Grund dafür sind die hörbaren Ultraschallsignale, welche die akustische Datenverarbeitung stören würden.

3. Laser-Entfernungsmesser:

Zur Entfernungsmessung wird ein zweidimensionaler Laserscanner eingesetzt. Bei diesen Geräten wird ein Laserstrahl in einer Ebene, nacheinander in verschiedene Richtungen gesendet. Durch die Messung der Laufzeit bis zum Eintreffen des von einem Hindernis reflektierten Strahles lässt sich die Entfernung zu diesem Hindernis berechnen. In dem der Scanner seinen kompletten Öffnungsbereich schrittweise ausleuchtet, lassen sich alle Objekte in einer Ebene erfassen. Die Punkte in diesem zweidimensionalen Tiefenbild lassen sich mit Hilfe einer Koordinatentransformation in dreidimensionale Raumkoordinaten umwandeln.
Der hier verwendete Entfernungsmesser hat einen Öffnungswinkel von 180 Grad und ist in 30cm Höhe parallel zum Fußboden angebracht. Die Winkelauflösung beträgt 0,5 Grad und die Reichweite 30m. Für den hier relevanten Bereich bis 8m beträgt der statistische Fehler +-15mm. Die beschriebene Ausrichtung des Scanners dient dem System dazu Beinpaare zu erfassen und somit die Position des Benutzers im Raum zu bestimmen.

4. Mikrofone:

Es werden zwei Mikrofone eingesetzt, welche für die Aufnahme von Sprache optimiert sind. Sie sind in einem Abstand von 28,1cm vor dem Flachbildschirm angebracht und eignen sich somit zur Sprecherlokalisierung. Außerdem werden sie für den verbalen Dialog eingesetzt.

5. Kameras:

Über dem Flachbildschirm ist auf einer Höhe von 135cm eine Farbkamera (Modell EVI-D31 der Firma SONY) angebracht. Diese Kamera ist schwenk- und neigbar. Außerdem verfügt sie über ein Zoomobjektiv, welches für diese Anwendung allerdings fest auf den grössten Weitwinkel eingestellt ist, um möglichst viele Personen zu erfassen. Für schnellere Raten bei der Verarbeitung wurden die Bilder auf 256x192 Pixel skaliert.

Neben den Sensoren ist noch der in 120cm Höhe angebrachte Flachbildschirm zu erwähnen. Es handelt sich dabei um einen 12 Zoll Touchscreen, welcher sich für interaktive Eingaben eignet. Hier wird er allerdings nur zur Darstellung eines Avatars verwendet. Unter dem Bildschirm in 90cm Höhe sind Stereolautsprecher für die Kommunikation angebracht.
Die Akkus ermöglichen eine Einsatzdauer von 30 Minuten bei vollem Betrieb. Der Roboter lässt sich wahlweise aber auch mit Netzteil betreiben.

2.1.1.2 Aufmerksamkeitssystem

Das Gesamtsystem teilt sich in zwei Teilsysteme auf. Ein Teil beinhaltet das Interaktionssystem, welches die Komponenten zur Sprachverarbeitung und Gestenerkennung verwendet, um mit dem Benutzer zu kommunizieren. Der andere Teil beinhaltet das Aufmerksamkeitssystem. Die Aufgabe dieses Systems ist es, die Aufmerksamkeit des Roboters gezielt zu lenken und so die vom Interaktionssystem benötigten Daten zu liefern. Dazu muss ein kommunikationswilliger Benutzer ausfindig gemacht und dieser im Fokus der Sensoren gehalten werden. Hilfreich ist es dabei auch, dem Benutzer die Aufmerksamkeit des Roboters zu vermitteln. Wenn dies nicht gelingt und der Benutzer nicht weiß, ob der Roboter ihn erfasst hat, wird er sich nach kurzer Zeit vom Roboter abwenden. Das System richtet seine Aufmerksamkeit auf Personen. Zusätzlich besteht die Möglichkeit die Aufmerksamkeit des Systems auf Objekte zu richten, zum Beispiel Objekte auf die gezeigt wird oder die in der Hand gehalten werden. Der Autor führt aber an, dass dies außerhalb der gesteckten Ziele liegt.
Als Szenario für dieses System wird das Home-Tour-Szenario gewählt, in welchem der Benutzer den Roboter ansprechen und im Haus (Raum) herumführen kann, um ihm verschiedene Dinge zu zeigen. In diesem Szenario, wie auch in anderen Szenarien mit Servicerobotern, gibt es zwei Phasen. Zum einen die Interaktionsphase und zum anderen die Bereitschaftsphase. Entsprechend dieser Phasen arbeitet auch das Aufmerksamkeitssystem unterschiedlich.

In der Bereitschaftsphase hält das System nach Kommunikationspartnern Ausschau. Zu diesem Zweck wird ein multimodales Anchoring zum Verfolgen von Personen verwendet (siehe Abb. 2.2). Dabei werden die Daten mehrerer Sensoren integriert, um eine robuste Verfolgung der Personen zu erreichen.

Mit dem Laserscanner werden Beinpaare detektiert. Dazu wird das Tiefenbild segmentiert, indem benachbarte Messpunkte, deren Abstände nur eine geringe Differenz aufweisen, zu Segmenten zusammengefasst werden. Diese Segmente bilden einzelne Objekte im Raum ab. Um aus diesen Objekten Beine zu selektieren, werden sie auf bestimmte Merkmale überprüft. Solche Merkmale sind zum Beispiel der Durchmesser der Beine und damit die Breite des Segments oder der maximale Abstand zwischen zwei zusammengehörigen Beinen. Zusätzlich zu den Beinen wird das Videobild zur Erfassung von Gesichtern und Oberkörpern verwendet. Die Gesichter werden mit dem in [Fritsch u. a. (2002)] vorgestellten Verfahren detektiert. Dabei wird das Bild aufgrund der Hautfarbe segmentiert und dann mit der sogenannten „Eigenface"-Methode auf Gesichter überprüft (verifiziert). Zusätzlich kommt auch das Verfahren [Viola u. Jones (2001)] zum Einsatz, welches in Kapitel 3 (Grundlagen) genauer beschrieben wird, da es auch bei dem in dieser Arbeit vorgestellten System verwendet wird. Die dritte Art von Sensoren, die in diesem Zusammenhang verwendet werden, sind die Stereomikrophone, mit welchen eine Sprecherlokalisation durchgeführt wird.
Wenn das System auf diese Weise eine oder mehrere Personen erfasst hat, wird aus diesen ein Kommunikationspartner selektiert. Hierbei wird die so genannte „bottom-up" gesteuerte Aufmerksamkeit verwendet, wobei der Roboter seine Aufmerksamkeit auf Personen richtet, die sprechen und ihn dabei ansehen. Dabei wird keine Person dauerhaft bevorzugt oder benachteiligt ausgewählt. Wenn ein Kommunikationspartner gefunden wurde, wird der Fokus auf ihn gerichtet und das System geht in die Interaktionsphase über.

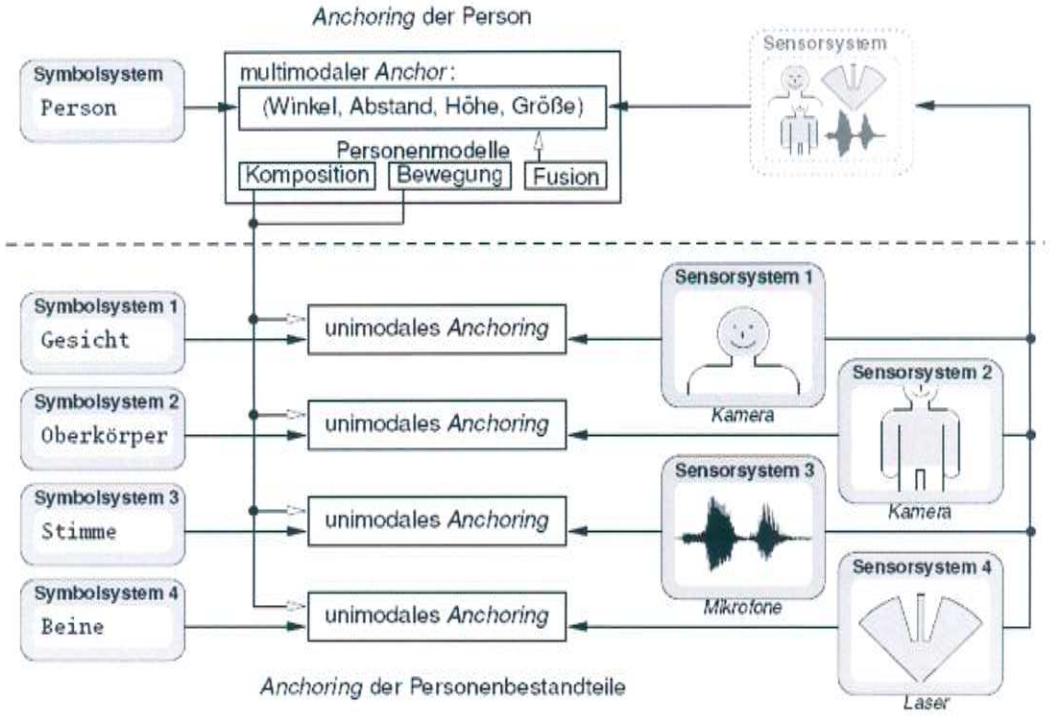

Abbildung 2.2: Beim Multimodalen Anchoring werden die Daten mehrerer Sensoren integriert, um eine robuste Verfolgung von Personen zu ermöglichen.

In der Interaktionsphase wird die Aufmerksamkeit auf eine andere Weise gesteuert. Hier wird der Fokus auf die Person gehalten, mit der die Interaktion stattfindet. Dieses Vorgehen wird hier als „top-down" gesteuerte Aufmerksamkeit bezeichnet. Das bedeutet, dass andere Personen, die den Roboter während der Interaktionsphase ansprechen, ignoriert werden. Mann muss dazu erwähnen, dass dieses Interaktionssystem, wie die meisten, für die Interaktion mit einem einzigen Benutzer ausgelegt ist. Dabei würde ein wechselnder Fokus die Interaktion unterbrechen.

Die Arbeit von [Lang (2005)] ist interessant für dieses Systems, da sich die Zielsetzungen überschneiden. Die Gemeinsamkeit der beiden Systeme besteht darin, dass sie die Umwelt beobachten und auf einen Benutzer warten, um anschließend ein Interaktion zu beginnen. Dabei ist vor allem der Bereich der Benutzerverfolgung (Tracking) interessant. In anderen Punkten lassen sich die Arbeiten jedoch schlecht vergleichen, da die Zielsetzungen nicht identisch sind. Das Ziel der Arbeit von [Lang (2005)] lässt sich am Home-Tour-Szenario zeigen, bei welchem ein Benutzer den Roboter anspricht, und während der Interaktion herumführt. Zielsetzung des in dieser Diplomarbeit entwickelten Systems ist es aber, nicht darauf zu warten angesprochen zu werden, sondern proaktiv einen Dialog herbeizuführen. Außerdem ist es nicht vorgesehen den Roboter zu bewegen, um einem Benutzer zu folgen, wie auch der eigentliche Dialog nicht zum Ziel dieser Arbeit gehört.

Abbildung 2.3: LEWIS (http://www.cse.wustl.edu/MediaAndMachines/Lewis/) ist ein mobiler Roboter, dessen Aufgabe es ist Personen zu fotografieren.

2.1.2 Lewis - The Robot Photographer

Lewis ist mobiler Roboter in Menschengröße, der an der Washington University in St. Louis[1] entwickelt wurde. Eine genaue Beschreibung des Systems findet sich in [Byers u. a. (2004)].Der Roboter kann sich durch einen Raum bewegen und dabei Fotos von den anwesenden Personen machen. Dazu muss er deren Gesichter erfassen, um dann die Kamera ausrichten und ein Foto machen zu können. Auch wenn das Ziel dieses Systems mehr im Bereich der Forschung liegt als in der Konkurrenz zu menschlichen Fotografen, gibt es trotzdem gewisse Vorteile gegenüber diesen. So hat sich gezeigt, dass sich Personen, sobald sie sich nach kurzer Zeit an den Roboter gewöhnt hatten, viel natürlicher reagierten, als wenn ein menschlicher Fotograf vor ihnen steht.
Die wichtigste Aufgabe in diesem Szenario ist die Erfassung von Gesichtern. Lewis geht dabei mit einem auf Farben basierenden Verfahren vor. Das Bild wird nach zusammenhängenden, hautfarbenen Bereichen abgesucht. Die so erfassten Bereiche werden als Kandidaten für ein Gesicht betrachtet. Zur Verifikation eines solchen Gesichtsbereichs werden zusätzliche Merkmale herangezogen. Erstens muss die Form stimmen. Hier wird eine mehr oder weniger elliptische Fläche erwartet, welche der Form des Gesichts entspricht. Des weiteren sollte sich die Größe in einem bestimmten Bereich befinden. Und zuletzt wird noch die Höhe des Objekts überprüft. Für diese beiden zuletzt erwähnten Schritte muss die Position des Benutzers bekannt sein. Deshalb werden ausgehend von der Positi-

[1] http://www.cse.seas.wustl.edu/

on eines möglichen Gesichts mit Hilfe eines Laserscanners die dazugehörigen Beine gesucht. Mit diesen Daten lässt sich dann die Position und Größe des Benutzers beziehungsweise seines Gesichts bestimmen und es lässt sich beurteilen, ob es sich dabei wirklich um ein Gesicht handelt. Durch eine Integration der Videodaten einer einzelnen Kamera mit den Daten eines Laserscanners gelingt es diesem System die dreidimensionale Position einer Person zu bestimmen. Dieses Vorgehen eignet sich allerdings nicht für das hier vorgestellte System, da kein Laserscanner verwendet werden soll.
Neben dieser Erfassung der Gesichter muss das System weitere Aufgaben erfüllen. Es muss ein geeigneter Bildausschnitt für die Fotos gewählt werden, was bei einer Person trivial ist, bei mehreren jedoch nicht. Desweiteren plant das System Pfade, auf denen sich Lewis durch den Raum bewegt. Diese beiden Punkte sind im Bezug auf die hier vorgestellte Arbeit allerdings nicht von Interesse. Getestet wurde das System unter realen Bedingungen, indem es auf mehreren Veranstaltungen als Fotograf arbeitete.

2.1.3 An Interactive Interface for Service Robots

In ihrer Arbeit stellen [Topp u. a. (2004)] ein System für einen interaktiven Service Roboter vor. Durch die Integration von Sprach-, Video und Laser-Entfernungs-Daten soll die Interaktion robust und natürlich sein. Daher soll die in dieser Arbeit durchgeführte Evaluation die Vorteile einer solchen Datenintegration bewerten.
Verwendete Sensoren: bewegte Kamera, Laserscanner, Mikrofon.

Das System arbeitet zustandsbasiert mit den folgenden vier Hauptzuständen.

1. „warte"-Zustand
2. „starte Kommunikation"-Zustand
3. „Kommunikations"-Zustand
4. „stop"-Zustand

Im ersten Zustand wird auf einen möglichen Benutzer gewartet. Dazu wird die Umgebung mit dem Entfernungsmesser beobachtet. Der Laserscanner ist dazu in einer Höhe von 93cm angebracht und eignet sich daher nicht dafür Beine zu erfassen. Deshalb werden Personen anhand der Abbildung ihres Körpers im Tiefenbild erfasst. Ein Körper ergibt im Tiefenbild eine konvexe Form, deren Größe in einem bestimmten Bereich liegt. Zusätzlich werden Informationen über die Bewegung von Objekten gesammelt. Sich bewegende Objekte werden dabei bevorzugt als Person in Erwägung gezogen. Für die Berechnung der Bewegung muss der Roboter in diesem Zustand still stehen. Ein System mit einem sich im Gegensatz hierzu bewegenden Roboter wird zum Beispiel in [Schulz u. a. (2001)] beschrieben. Wenn eine Person im Tiefenbild ausgemacht wurde, richtet der Roboter seine Aufmerksamkeit auf diese, indem er die Kamera in deren Richtung dreht. Dies dient dazu, dem möglichen Benutzer die Aufmerksamkeit des Roboters zu zeigen. Mit Hilfe der Kamera wird versucht das Gesicht und die Hände der Person zu erfassen, um zu bestätigen, dass es sich bei der im Tiefenbild erfassten Kontur um einen Menschen handelt. Hierzu werden hautfarbene Bereiche segmentiert und diese dann durch ihre Größe, Form und Lage als Gesicht und Hände verifiziert. Die Informationen über Größe und Lage ergeben sich aus der Kombination der Größe und Position der hautfarbenen Bereiche im Videobild mit den Entfernungsdaten des Laserscanners.

Wenn ein Benutzer vom Videosystem verifiziert wurde, wird versucht die Kommunikation zu beginnen (Zustand 2). Dazu wird die Person angesprochen (siehe Abbildung 2.4).

Abbildung 2.4: Nachdem der Roboter einen möglichen Benutzer erfasst hat, dreht er die Kamera in dessen Richtung und spricht ihn an.

Wenn die Spracherkennung eine Bestätigung des Benutzers liefert, geht das System in den Kommunikationszustand über. Falls der Benutzer die Kommunikation zurückweist oder über einen gewissen Zeitraum eine Antwort ausbleibt, ignoriert das System diesen Benutzer und beginnt wieder mit der Suche nach einem neuen Benutzer. Während der Kommunikation kann der Benutzer dem System bestimmte Befehle geben. Es berücksichtigt dabei auch Gesten, welche mit Hilfe der vom Tracker erfassten Hände erkannt werden. Die Bewegung oder Haltung der Hände alleine lässt jedoch noch nicht eindeutig auf eine bestimmte Geste schließen. Hierzu wird ein zusätzlicher Dialogkontext benötigt. Dieser wird durch eine weitere Aufteilung des Kommunikationszustandes in Teilzustände bereitgestellt. Der aktuelle Kommunikationszustand ergibt sich aus dem Verlauf des Dialoges. Mit Hilfe dieser Zustände ist es möglich, nur dann eine Gestenerkennung durchzuführen, wenn auch eine Geste zu erwarten ist, wodurch die Erkennung robuster wird. Mit einem entsprechenden Befehl kann der Benutzer die Kommunikation beenden und das System kehrt in seinen Anfangszustand zurück. Im Hinblick auf das hier entwickelte System, sind die ersten beiden Zustände interessant. Dabei wird die Umgebung des Roboters beobachtet und solbald ein Benutzer in der Nähe ist, wird dieser angesprochen. Das System reagiert dabei allerdings nur auf Personen, die Interesse am Roboter zeigen. Es wird nicht versucht das Interesse eines Benutzers zu erzeugen.

2.1.4 Virtual Mirror

Im Bereich der Benutzererfassung war für diese Arbeit ein visueller Ansatz geplant, da der eingesetzte Roboter auf jeden Fall über Videokameras verfügt und das System nicht von zusätzlichen Sensoren abhängig sein sollte. Weitere Möglichkeiten wie Laserscanner oder Sonarsensoren kamen daher nicht in Frage. Durch zwei auf dem Roboter vorhandene Kameras lassen sich Stereobilder aufnehmen und Tiefeninformationen berechnen. Ein mit Stereokameras arbeitendes System wurde auch von Darell u. a. (1998) erstellt.

Ziel dieses Systems war eine robuste Erfassung von Gesichtern, wobei die Position und die Größe des Gesichts wichtig waren. Um dies zu erreichen, vor allem um die Fehlerquote zu verringern, wurden dazu drei Bildverabeitungstechniken kombiniert. Der Aufbau ist in Abbildung (2.5) zu erkennen. Im ersten Verabeitungsschritt werden ausgehend vom Stereobild Tiefeninformationen berechnet. Dadurch kann der Vordergrund vom Hintergrund segmentiert werden und es lässt sich eine Maske erzeugen, welche nur Bildregionen im Vordergrund umschließt. Damit wird der Suchraum für den nächsten Verarbeitungsschritt eingeengt. Dies hat den Vorteil, dass einerseits die weitere Verarbeitung beschleunigt wird und andererseits die Anzahl der fälschlicherweise erkannten Personen sinkt, da

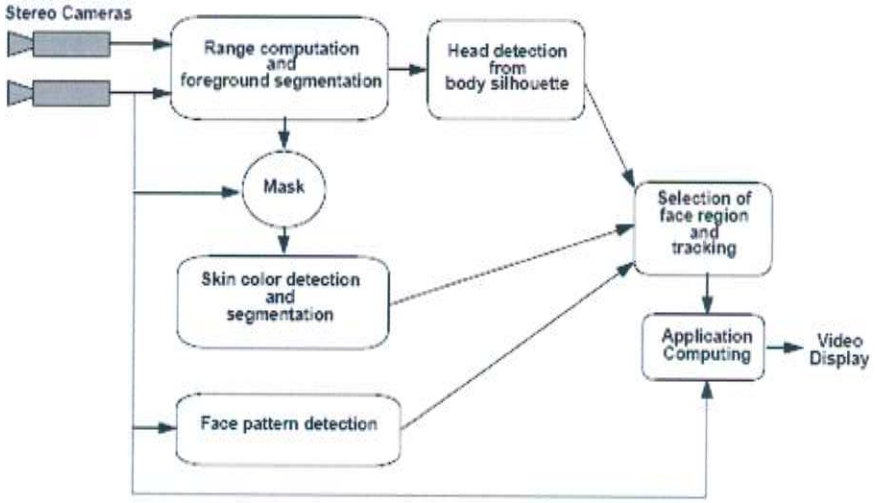

Abbildung 2.5: Systemaufbau

es keine Fehlerkennungen im Hintergrundbereich mehr geben kann. Im nun folgenden Verabeitungsschritt werden hautfarbene Regionen segmentiert. Im letzten Schritt werden dann, wie zuvor, nur diese segmentierten Bereiche untersucht, mit den schon erwähnten Vorteilen. Dieser letzte Schritt sieht eine Mustererkennung vor, um das Gesicht von den Händen und anderen hautfarbenen Bereichen zu unterscheiden. Für die Vordergrundsegmentierung verwenden die Autoren den „census correspondence"-algorithmus [Zabih u. Woodfill (1994)], welcher ein „disparity image" erzeugt. Implementiert haben sie diesen Algorithmus auf einer multi-FPGA-pci-Karte und konnten damit 24 „disparities" bei einer Auflösung von 320x240 Punkten mit 42 Bildern pro Sekunde berechnen. Das Disparitätenbild liefert Tiefeninformationen an kontrastreichen Stellen, wie zum Beipsiel am Übergang zwischen zwei Objekten. Somit wird eine Person durch ihre Silhouette abgebildet. Ausgehend von dieser Annahme wird das dreidimensionale Bild nach dem am nähesten gelegenen Umriss, welcher in seiner Größe einem Menschen entspricht abgesucht. Wenn ein solcher Umriss gefunden ist, wird er solange von Bild zu Bild verfolgt, bis er verschwunden ist. Dazu werden die Positionsänderungen der Umrisse geschätzt und die so berechnete zukünftige Position eines Umrisses mit der tatsächlich aufgenommenen in Zusammenhang gebracht. Wie schon erwähnt, dient die so erfasste Silhouette als Maske für die nächsten Schritte. Unter der Annahme, dass der Kopf der höchste Punkt des Umrisses ist, lässt sich die Kopfposition an dieser Stelle schon schätzen. Dieser Wert kann bei Ausfall der übrigen Systeme immer noch herangezogen werden. Ähnlich ist es mit der Gesichtsgröße. Diese lässt sich auch Aufgrund der Entfernung abschätzen.

Für die Farbsegemtierung werden einzelne Pixel in zwei Klassen aufgeteilt, hautfarben und nicht hautfarben. Dabei beruht die Klassifikation auf dem Farbton und ist weitestgehend unabhängig von der Intensität und der Sättigung. Damit ist dieses Verfahren robust gegenüber verschiedenen Lichtverhältnissen und Hautfarben, da diese nicht den Farbton ändern. Der verwendete Klassifikator arbeitet mit einem Gauß'schen Wahrscheinlichkeitsmodel und ist mit dem in [Fleck u. a. (1996)] verwendeten Klassifikator vergleichbar. Als Ergebnis dieser Klassifiaktion liegt für jeden Pixel die Wahrscheinlichkeit vor, mit der er hautfarben ist. Daraus lassen sich, wie schon im vorherigen Schritt Bereiche des Bildes segmentieren und als Maske an den nächsten Schritt weiterreichen. Die so erfassten Hautbereiche werden wie im vorigen Schritt verfolgt. Falls die anderen Systeme ausfallen ist

auch hier eine Schätzung der Gesichtsposition möglich, indem der am höchsten gelegene hautfarbene Bereich, dessen Proportionen einem Gesicht entsprechen, ausgewählt wird.

Falls die ersten beiden Schritte erfolgreich waren, kann die rechenintensive Mustererkennung auf wenigen kleinen Bereiche des Gesamtbildes durchgeführt werden. Ziel dieses Schrittes ist es, das Gesicht von anderen hautfarbenen Bereichen, wie zum Beispiel den Händen, zu unterscheiden. Der hier verwendete Detektor basiert auf dem an der CMU entwickelten Face Detector (Rowley u. a. (1996)), welcher mit Hilfe eines neuronalen Netzes frontale Gesichter erkennt. Im einfachsten Fall wird so ein Gesichtsmuster in einer Hautregion gefunden. Diese wird dann als Ziel markiert. In dem Fall, dass schon eine Region als Ziel markiert ist, aber nicht erneut ein Muster erkannt wird, bleibt diese Region, welche ja verfolgt wird, weiterhin als Ziel markiert. Es sei denn, die Größenänderung dieser Region überschreitet einen bestimmten Wert. Zusätzlich werden Fälle berücksichtigt, in denen sich Hautregionen verbinden oder aufteilen. Dies kann zum Beispiel vorkommen, wenn eine Hand vor das Gesicht bewegt wird. Bei Ausfall der anderen Systeme muss die Mustererkennung auf dem gesamten Bild durchgeführt werden.

Als Abschluss ihrer Arbeit haben die Autoren ihr System im gesamten, aber auch die Teilsysteme einzeln getestet. Dabei zeigt sich, wie zu erwarten, eine Verbesserung durch die Kombination der Teilsysteme. Die Tiefensegementierung alleine stellte sich als bester Erkennungsschritt heraus. Das Gesamtsystem hat eine Fehlerrate von 3,3% die Tiefensegmentierung mit Schätzung der Gesichtsposition eine Fehlerrate von 5,4%.

Auch wenn die Ergebnisse gut sind, eignet sich dieser Ansatz nicht für das hier vorgestellte System. Einerseits werden keine Angaben zur Geschwindigkeit des Gesamtsystems gemacht. Alleine die Mustererkennung benötigt schon 0,8 Sekunden. Da aber vor allem die Echtzeitfähigkeit für die Interaktion mit einem Benutzer wichtig ist, würde es hier wahrscheinlich Problemen geben. Des Weiteren ist das System ungeeignet, da es nicht mit mehreren Personen gleichzeitig funktioniert. Der wichtigste Punkt gegen eine Verwendung ist allerdings die Beschränkung auf den Vordergrund. Dadurch lassen sich nur Personen erfassen, welche schon ziemlich nahe am Roboter sind. Die restlichen potentiellen Benutzer können nicht berücksichtigt werden und gehen dem System verloren.

Eine weitere Arbeit zur Gesichtserfassung ist von Viola u. Jones (2001). Das in dort vorgestellte Verfahren arbeitet ausschließlich mit einer Mustererkennung, welche nicht auf die Erkennung von frontalen Gesichtern oder von Gesichtern allgemein beschränkt ist, sondern auch auf andere Muster trainiert werden kann. Der eigentliche Vorteil und auch ein Hauptziel dieses Verfahrens ist die hohe Geschwindigkeit, mit der die Einzelbilder bearbeitet werden. Da die Details bei den Grundlagen in Kapitel 3.1 beschrieben werden, soll hier nur ein kurzer Überblick gegeben werden. Wie schon erwähnt ist das Ziel des Verfahrens eine hohe Verarbeitungsgeschwindigkeit, bei einer gleichzeitig guten Erkennungsquote. Grundlage für die Erkennung sind monochrome Einzelbilder. Das Verfahren ist merkmalsbasiert und arbeitet mit mehreren sogenannten „weak-classifiers", welche in einer Kaskade angeordnet sind. Um die Bearbeitungszeit dieser „weak-classifiers" zu minimieren, kommt ein spezielles Bildformat zu Anwendung, welches die Autoren als Integralbild bezeichnen. Damit lassen sich die von den „weak-classifiers" durchgeführten Intensitätsberechnung von rechteckigen Flächen schnell durchführen. Dieses Verfahren ist die Grundlage für den in der hier vorgestellten Arbeit verwendeten Multi-User-Tracker. Der Vorteil des Verfahrens ist, dass es sehr schnell arbeitet und somit auch auf einem Ro-

boter mit begrenzter Rechenleistung betrieben werden kann. Außerdem ist es in der Lage, mehrere Gesichter gleichzeitig zu erfassen.

2.2 Proaktives Handeln

Wörtlich übersetzt bedeutet „proaktiv" zuhandelnd. Oft wird proaktiv als Gegenteil von reaktiv betrachtet. Es beschreibt also eine Aktion die im Gegensatz zur Reaktion nicht die Folge einer anderen Aktion ist. Es ist somit die erste Aktion einer Folge miteinander verbundener Aktionen. Allerdings geht die Bedeutung noch weiter. Eine proaktive Handlung ist zielgerichtet und ist dafür bestimmt, ein gewünschtes Ereignis auszulösen. Systeme, die sich mit proaktivem Handeln befassen, gibt es besonders im Bereich der Geschäftswelt und der Politik. Es gibt aber auch Robotersysteme, die sich mit dem proaktiven Ausführen von Aktionen beschäftigen. Ein solches System wird in der Arbeit [Schmid u. a. (2005)] vorgestellt, welches eine im Vergleich zu anderen Systemen intuitivere Kommunikation zwischen Mensch und Roboter zum Ziel hat. Um dies zu erreichen, sollen implizite Absichten bei der Kommunikation berücksichtigt werden. Das heißt der Roboter reagiert nicht nur auf explizite Befehle und Wünsche des Benutzers, sondern ist in der Lage, die Absichten des Benutzers zu erkennen und proaktiv auf diese zu reagieren.

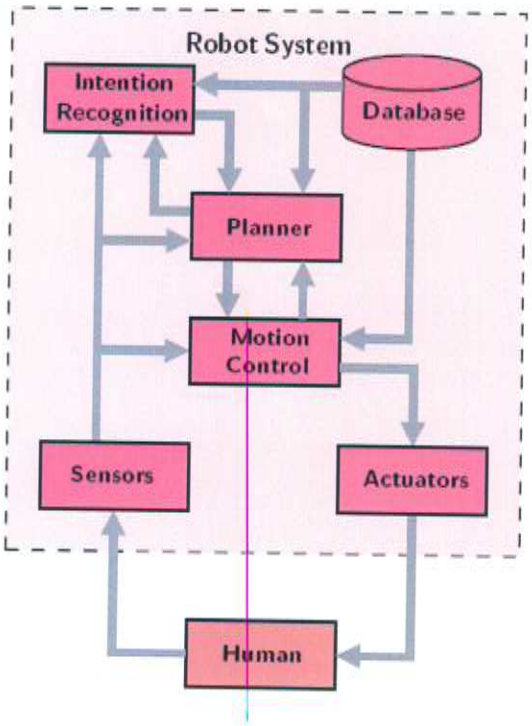

Abbildung 2.6: Proaktives Interaktionssystem

Die Architektur dieses Systems ist in Abb. 2.6 zu sehen. Als zentrales Modul des Systems ist das „Planner"-Modul zu betrachten. Es steuert die anderen Module und plant im speziellen die Ausführung der Aktionen. Die Entscheidungen werden ausgehend von den Informationen getroffen ,welche durch „Database", „Intention Recognition" und Sensoren bereitgestellt werden. Das „Motion Control"-Modul führt die geplanten Aufgaben aus, indem es die notwendigen Steuersignale für die „Actuators" erzeugt und die Ausführung überprüft. Grundlage für die proaktiven Handlungen des Systems ist das „Intention

Recognition"-Modul. Es dient dazu die (unausgesprochenen) Ziele und Absichten des Benutzers zu erkennen. Es wird davon ausgegangen, dass die Handlungen einer Person die Folge seiner Absichten sind. Daher soll durch die Beobachtung der Handlungen auf die Absichten geschlossen werden. Da die Folgerungen nicht sicher sind, werden gegebenenfalls proaktive Handlungen eingeplant, um die Absicht des Benutzers zu überprüfen und damit die Unsicherheit des Systems zu verringern.

3. Grundlagen

In diesem Kapitel werden zugrundeliegende Techniken und Verfahren beschrieben. Die ersten beiden Abschnitte beinhalten Verfahren zur Videoverabeitung. Der erste Abschnitt erklärt die hier verwendeten Verfahren zur Erfassung und Verfolgung von Personen. Die Erfassung der Personen liefert zweidimensionale Koordinaten. Deshalb ist eine seperate 3D-Berechnung nötig. Die Grundlagen für diese 3D-Berechnungen werden im zweiten Abschnitt vorgestellt. Der dritte Abschnitt liefert die Grundlagen für die Interessenklassifikation mit Hilfe eines neuronalen Netzes. Im letzten Abschnitt wird der Dialogmanager Tapas beschrieben.

3.1 Erfassen von Gesichtern und Tracking

Der in diesem System verwendete „Multi-Personen-Tracker" basiert auf einer musterbasierten Erkennung von Gesichtern. Dazu wird die durch die OpenCV-Bibliothek [1] bereitgestellte Erkennung von Gesichtern verwendet. Die zugrundeliegenden Techniken dieser Gesichtserkennung beruhen auf der Arbeit von Viola u. Jones (2001). Ziel dieser Arbeit ist eine Gesichtserkennung, die schnell ist und trotzdem eine gute Erkennungsquote aufweist. Die Autoren führen drei Schlüsseltechnologien ihrer Arbeit an. Erstens eine spezielle Repräsentation von Bildern, welche sie Integralbild nennen, zweitens einen auf AdaBoost [Freund u. Schapire (1995)] basierenden Lernalgorithmus und als letzes die spezielle Anordnung mehrerer Klassifikatoren als Kaskade.
Das System verwendet weder Informationen, welche sich aus den Veränderungen in einer Videosequenz berechnen lassen, noch Farbinformationen. Das heißt es arbeitet mit einzelnen Graubildern. Die Autoren führen jedoch an, dass sich die Verarbeitung dieser zusätzlichen Informationen in das System integrieren ließe, um damit im Besonderen die Verarbeitungsgeschwindigkeit zu verbessern. Zur Erkennung von Gesichtern wird das Bild nach bestimmten Merkmalen durchsucht. Hierbei finden „haar-features" Anwendung, wie sie auch in [Papageorgiou u. a. (1998)] verwendet werden. Diese Merkmale lassen sich nach ihrer Art in drei Klassen aufteilen. Zum ersten die sogenannten „two-rectangle features", welche aus zwei deckungsgleichen Rechtecken bestehen, die zusammen wieder ein Rechteck bilden. Der Wert dieser Merkmale errechnet sich aus der Summe

[1] **Open** source **C**omputer **V**ision; http://www.intel.com/technology/computing/opencv/index.htm)

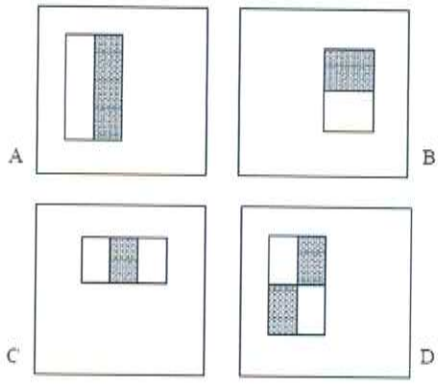

Abbildung 3.1: Die Summe der Pixel in den weißen Rechtecken wird von denen in den grauen subtrahiert. Bild (A) und (B) zeigen sogenannte „two-rectangle features", Bild (C) ein „three-rectangel feature" und Bild (D) ein „four-rectangle feature".

aller Pixel eines Rechtecks, welche dann von der Summe aller Pixel des anderen Rechtecks abgezogen wird. Die Summe der Pixel ist dabei die Summe der Intensitätswerte des Graubilds. Dementsprechend gibt es noch „three-rectangle features", bei denen die äußeren Rechtecke vom Inneren abgezogen werden, und „four-rectangle features", bei denen die diagonal gegenüberliegenden Rechtecke ein Paar bilden und von den beiden übrigen abgezogen werden. Veranschaulicht werden diese Merkmale in Abbildung 3.1.

Der Detektor arbeitet mit quadratischen Feldern von 24x24 Pixeln. Ein spezielles Merkmal innerhalb einer Merkmalsklasse wird durch seine Position (den obersten linken Punkt) innerhalb des Feldes, sowie den Seitenlängen der Rechtecke bestimmt. Da bei diesem Vorgehen sehr viele Pixelsummen berechnet werden müssen und jeder einzelne Pixel mehrfach darin vorkommt, verwenden die beiden Autoren an dieser Stelle das Integralbild. Bei diesem Vorgehen werden alle Pixel nur einmal betrachtet um dieses Integralbild zu berechnen. Das Integralbild ist mit dem in [Crow (1984)] aufgeführten „summed area table" zu vergleichen. Danach können die Summenberechnungen in wenigen Schritten durch Auslesen der Tabelle und einfache Berechnungen durchgeführt werden. Für das Integralbild wird zu jedem Punkt des Originalbildes ein Wert berechnet. Dieser Wert ergibt sich aus der Summe der Intensitätswerte aller Punkte, die oberhalb und links von diesem Punkt liegen (siehe Abbildung 3.2).

Folgende Gleichung beschreibt diesen Zusammenhang:

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y')$$

wobei ii(x,y) das Integralbild und i(x,y) das Originalbild ist. Die einzelnen Punkte lassen sich iterativ nach den folgenden zwei Gleichungen berechnen:

$$s(x,y) = s(x,y-1) + i(x,y)$$
$$ii(x,y) = ii(x-1,y) + s(x,y)$$

wobei s(x,-1) = 0 und ii(-1,y) = 0
s(x,y) bildet die Summe der Spalte x bis einschließlich Punkt (x,y). Damit lässt sich das Integralbild in einem Durchgang durch das Originalbild erstellen.

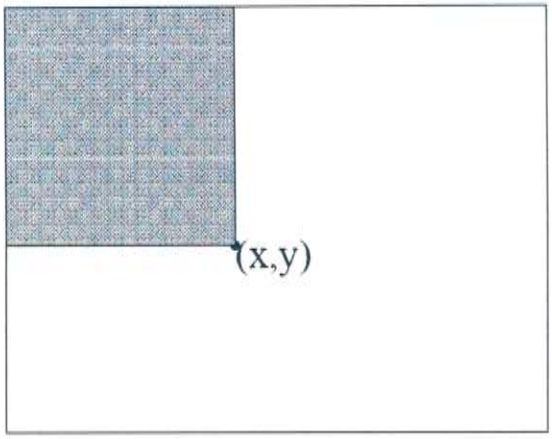

Abbildung 3.2: Der Wert des Integralbildes am Punkt (x,y) ergibt sich aus der Summe aller Intensitätswerte, die oberhalb und links von diesem Punkt liegen (grau hinterlegter Bereich)

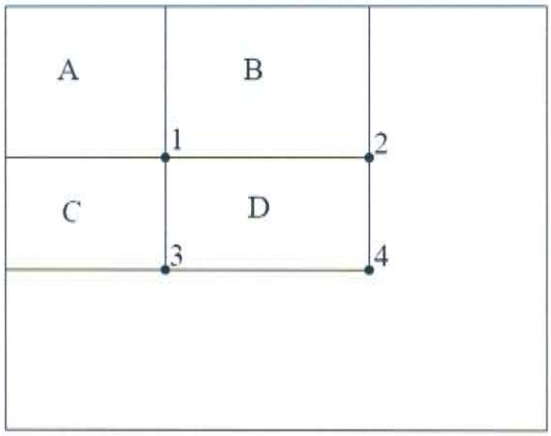

Abbildung 3.3: Die Summe der Intensitätswerte des Rechtecks D lässt sich aus den Integralwerten seiner Eckpunkte ((1),(2),(3),(4)) berechnen. Der Integralwert von D lässt sich mit folgender Formel berechnen. I(D) = (4)-(2)-(3)+(1)

Jeder Punkt (x,y) des Intagralbildes enthält jetzt die Summe der Bildpunkte eines Rechtecks, welches am linken oberen Bildrand beginnt und im Punkt (x,y) endet.
Durch vier solcher Rechtecke lässt sich ein beliebiges, parallel zum Bild ausgerichtetes Rechteck berechnen. Dies wird in Abbildung(??) veranschaulicht. Zur Berechnung des Integralwertes I(D) von Rechteck D wird der Wert an Punkt 4 genommen, welcher dem Wert der Rechtecke (A+B+C+D) entspricht. Punkt 2 (A+B) wird davon subtrahiert, ebenso wie Punkt 3 (A+C). Am Ende muss Punkt 1 (A) wieder addiert werden, da er doppelt abgezogen wurde. Daraus ergibt sich folgende Gleichung:

$$I(D) = (4) - (2) - (3) + (1)$$

3.2 3D-Berechnung

Für die 3D-Berechnungen wurde die SVS-Library[2] der Firma Videre Design verwendet . Eine gut verständliche Beschreibung dieses Verfahren findet sich in [Nickel (2003)].

[2]http://www.videredesign.com/

3.2.1 Optische Grundlagen

Die 3D-Berechnungen dieses Systems beruhen auf dem gleichen optischen Prinzip, welches auch die Grundlage der menschlichen 3D-Wahrnehmung bildet. Durch das Aufnehmen zweier Bilder mit versetzten Kameras findet sich ein Objekt in beiden Bildern an verschiedenen Positionen. Indem man den Versatz der beiden Abbildungen des Objektes misst, lassen sich dann dessen dreidimensionale Koordinaten berechnen. Um diese Berechnungen durchzuführen wird die vereinfachte Annahme getroffen, dass es sich bei den Kameras um zwei verzerrungsfreie (Loch-)Kameras handelt, welche in einer horizontalen Ebene parallel zueinander angeordnet sind. Diese Annahme ermöglicht einfache Berechnungen mit Hilfe des Strahlensatzes. Für diese Berechnungen müssen die folgenden Werte bekannt sein.

1. b : der horizontale Abstand der beiden Kameras auf der Grundlinie
2. f : die Brennweite der Kameras
3. x_L, x_R : die horizontale Bildkoordinate des Objektes in der linken und der rechten Kamera

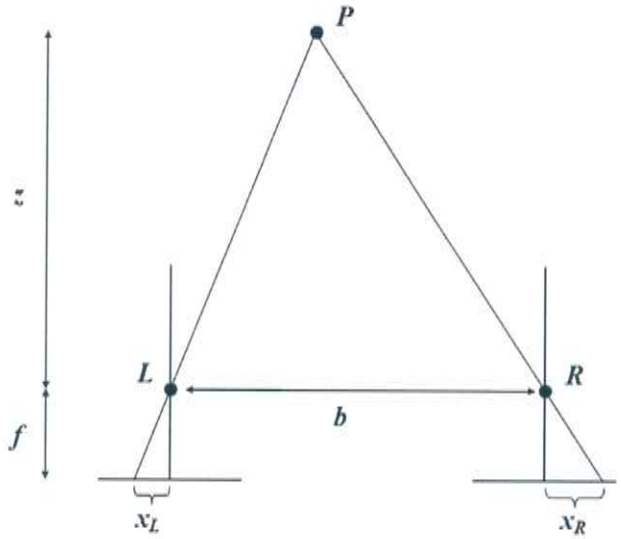

Abbildung 3.4: Zwei parallel im Abstand b angeordnete Kameras mit Brennweite f bilden den selben Punkt P auf verschiedene Bildpunkte ab. Durch die Differenz der Bildkoordinaten x_L und x_R lässt sich die Entfernung z berechnen.

Aus diesen Daten lässt sich mit Hilfe des Strahlensatzes die Entfernung z eines Objektes berechnen. Die Zusammenhänge werden in Abbildung 3.4 deutlich. Zunächst wird die sogenannte (horizontale) Disparität d (siehe [Jähne (1997)]) folgendermaßen berechnet: $d = x_L - x_R$.
Aus der Disparität d lässt sich dann die Entfernung z berechnen:

$$z = \frac{f * b}{d}$$

Ist die z-Koordinate des Objekts bekannt, lassen sich auch dessen x- und y-Koordinaten berechnen:

$$x = \frac{x_L * z}{f}, y = \frac{y_L * z}{f},$$

Diese Koordinaten gelten für ein Koordinatensystem, dessen Ursprung im Brennpunkt der linken Kamera liegt. Ebenso ließe es sich für die rechte Kamera berechnen, indem die Bildkoordinaten dieser Kamera x_R, y_R verwendet werden. Die Entfernung ist, wie die Formel zeigt, umgekehrt proportional zur Disparität. Je näher ein Objekt kommt, desto größer wird die Disparität. Diese kann jedoch nicht beliebig groß werden, bzw. der Abstand beliebig klein, da bei einem zu geringen Abstand das Objekt nicht mehr von beiden Kameras gleichzeitig erfasst werden kann. Daraus folgt, dass die Tiefenberechnung nur bis zu einem Mindestabstand funktionieren kann. Die Abgrenzung dieses Bereichs wird als Horopter bezeichnet.

3.2.2 Kalibrierung

Wie schon erwähnt wurde die Annahme zweier optimaler Kameras getroffen. Da dies in der Praxis nicht zutrifft, müssen diese vom System durch eine künstliche Entzerrung der Bilder simuliert werden. Dieser Vorgang nennt sich Rektifizierung.
Für die Rektifizierung werden verschiedene Informationen bezüglich der Kameras benötigt. Um diese Information zu berechnen, beinhaltet die SVS-Bibliothek ein Programm zu Kalibrierung des Videosystems. Dabei findet einerseits eine so genannte interne Kalibrierung statt, bei der beide Kameras einzeln kalibriert werden. Dazu werden die Verzerrungen der Optik berechnet, um diese ausgleichen zu können. In einem auf diese Weise ausgeglichenen Bild werden gerade Linien in der aufgenommenen Szene auch gerade abgebildet. Zusätzlich zu dieser internen findet auch noch eine externe Kalibrierung statt. Dabei wird die Position der Kameras zueinander berechnet.
Zur Durchführung dieser Kalibrierungen wird ein Schachbrettmuster aus mehreren Perspektiven aufgenommen.

3.2.3 Korrespondenzen

Eine zweite Annahme die getroffen wurde ist, dass man einem Objekt dessen Abbildung in beiden Bildern zuordnen kann. Dazu ist es aber nötig zu wissen, an welchen Bildkoordinaten sich ein Objekt jeweils im linken und im rechten Bild befindet. Dies wird als Korrespondenzproblem bezeichnet.
Es gibt verschiedene Möglichkeiten dieses Aufgabe zu lösen, wobei die SVS-Library ein in [Konolige (1997)] beschriebenes Verfahren verwendet. Dabei werden quadratische Suchfenster mit einer Seitenlänge von 5-13 Punkten gebildet. Es sollen nun die zusammengehörigen Bildausschnitte im linken und rechten Bild gefunden werden. Dazu wird in jedem Punkt des linken Bildes ein solches Fenster positioniert. Im rechten Bild werden dann die zu diesen Fenstern korrespondierenden Fenster gesucht. Dazu wird die Korrelation der beiden Fenster berechnet, wozu die Summe der (absoluten) Differenzen der Helligkeitswerte von zusammengehörigen Punkten des linken und des rechten Fensters gebildet wird. Je geringer diese Differenzen sind, desto größer ist die Korrelation der beiden Fenster. Um nun zusammengehörige Objekte im linken und rechten Bild zu finden, wird für jedes Fenster ein maximal korrelierendes Fenster im anderen Bild gesucht. Ein Problem bei diesem Vorgehen ist, dass sich innerhalb einer schwach texturierten Fläche kaum ein maximal korrelierendes Fenster bestimmen lässt, da sich alle in dieser Fläche liegenden Fenster sehr ähnlich sind.
Durch die Kalibrierung ist es möglich die Kamerabilder zu entzerren. Außerdem lassen sich die Kameras parallel zueinander ausrichten und es lässt sich der vertikale Versatz ausgleichen. Dadurch werden alle horizontalen Linien in der aufgenommenen Szene auf beiden Bildern in der selben Zeile abgebildet. Daher müssen korrespondierende Bildteile

in beiden Bildern in der gleichen Zeile liegen, wodurch sich der Suchraum einschränken lässt, indem das maximal korrelierende Fenster nur in der jeweils entsprechenden Zeile gesucht wird. Hierzu wird das Fenster schrittweise durch die Zeile bewegt und jeweils die Korrelation berechnet. Wenn ein korrenspondierendes Fenster gefunden wird, ergibt sich die Disparität des dazugehörigen Objekts aus dem Versatz der beiden korrepondierenden Fenster. Der Suchraum lässt sich weiter einschränken, indem eine maximale Disparität festgelegt wird. Somit muss nicht die ganze Zeile durchsucht werden, dadurch erhöht sich allerdings auch der schon erwähnte Mindestabstand.

3.3 Klassifikation

Das Interesse eines Benutzers soll mit Hilfe eines Klassifikationsverfahrens bestimmt werden. Die Art des Interesses wird hierzu in verschiedene Klassen unterteilt. Anschließend soll, ausgehend von bestimmten Merkmalen einer Situation, eine Entscheidung getroffen werden, in welcher Klasse diese Situation einzuordnen ist. Es gibt verschiedene Klassifikationsverfahren, welche sich durch ihre Eigenschaften unterscheiden lassen [3].

1. Manuelle und maschinelle Verfahren
2. Statistische und verteilungsfreie Verfahren
3. Überwachte und nicht-überwachte Verfahren

Automatische Verfahren sind diejenigen, welche von einer Maschine mit Hilfe eines Algorithmus ausgeführt werden. Ziel ist es dabei durch erlernen von Entscheidungen in bekannten Situationen, in der Lage zu sein, auch in unbekannten Situationen eine Entscheidung zu treffen. Die maschinelle Klassifikation ist somit ein Teilgebiet des maschinellen Lernens. Statistische und verteilungsfreie Verfahren unterscheiden sich in der Art des Ergebnisses. Während statistische Verfahren Wahrscheinlichkeiten ausdrücken, indem sie keine eindeutigen Entscheidungen treffen, liefern verteilungsfreie Verfahren diskrete Ergebnisse. Damit geht allerdings auch die Information verloren, wie sicher dieses Klassifikationsergebnis ist. Im Bezug auf die Lernweise lassen sich überwachte und nicht überwachte Verfahren unterscheiden. Bei den überwachten Verfahren gibt ein Benutzer dem System vor, wie das Ergebnis in einer bestimmten Anzahl bekannter Situationen(Trainingsdaten) auszusehen hat. Er gibt im speziellen die möglichen Klassen vor, zwischen denen das System zu entscheiden hat. Beim unüberwachten Lernen greift der Benutzer nicht in den Lernprozess ein. Das System bildet anhand der Gemeinsamkeiten des Datensatzes selbständig Klassen. Der Benutzer muss nachträglich entscheiden, ob diese Einteilung sinnvoll ist. Eine spezielle Form des unüberwachten Lernens ist das „Reinforcement Learning“, bei dem das System ein Feedback über die Qualität der Klasseneinteilung in Bezug auf eine spezielle Aufgabenstellung bekommt, und die Einteilung dementsprechend anpassen kann.

3.3.1 Neuronale Netze

Bei der Umsetzung dieser lernenden Klassifikatoren finden Neuronale Netze Anwendung. Eine gute Einführung in dieses Thema gibt es in [AG]

[3] Quelle: http://www.wikipedia.de

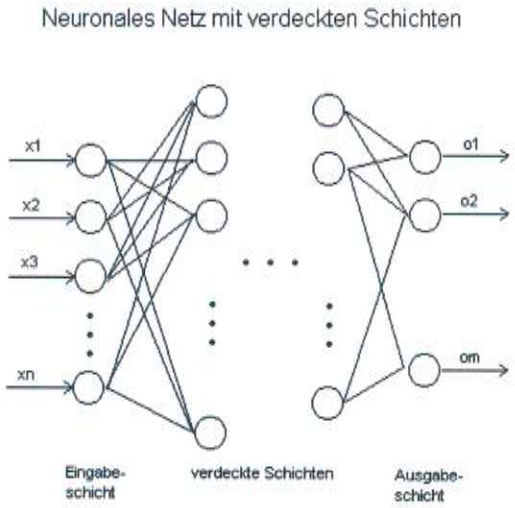

Abbildung 3.5: Neuronales Netz

Wie in vielen Bereichen der Wissenschaft dient auch bei dieser Technik die Natur als Vorbild. Neuronale Netze haben ihr biologisches Vorbild im Nervensystem. Dieses wird in vereinfachter Form in einem Algorithmus umgesetzt. Die interessanteste Eigenschaft dieser Struktur ist dabei die Lernfähigkeit. Das Netz besteht aus mehreren eigenständigen Neuronen, welche durch Synapsen miteinander verbunden sind. Die Neuronen besitzen Eingänge aus Ausgänge.

Das Netz entsteht durch die Verbindung der Ausgänge mit den Eingängen anderer Neuronen (siehe Abbildung 3.5). Die Neuronen sind dabei in Schichten angeordnet. Ein Neuron kann Eingangswerte von mehreren Neuronen erhalten, berechnet daraus einen Ausgabewert, und gibt diesen weiter. Die Berechnung des Ausgabewertes ergibt sich aus der Propagierungsfunktion und der Aktivierungsfunktion. Die Propagierungsfunktion bildet eine gewichtete Summe über alle Eingänge. Dadurch werden die Eingangssignale bewertet, je nach Bedeutung fließen sie mehr oder weniger (oder negativ) ins Ergebnis ein. Die Aktivierungsfunktion liefert dann das Ergebnis. Die Neuronen lassen sich entsprechen ihrer Aktivierungsfunktion in Klassen einteilen. Es gibt lineare, sprunghafte(Schwellert) und stetige(Sigmoid) Aktivierungsfunktionen. Neuronale Netze können außerdem durch den verwendeten Lernprozesse unterschieden werden. Die meisten Netze lernen nach der Hebb'schen Regel. Dazu zählen die beiden schon aufgegührten Formen des überwachten und unüberwachten Lernens. Das überwachte Lernen funktioniert durch einen Vergleich der vorgegebenen SollWerte mit den, vom Netz errechneten, Ist-Werten. Der daraus resultrierende Fehler soll durch eine Veränderung der Gewichtung verringert werden. Im pratkischen Einsatz wird bei den meisten neuronalen Netzen das sogenannte Backpropagation-Verfahren zum Lernen verwendet, welches in [Boenke (2003)] genauer beschrieben wird. Ein Problem beim Lernen stellen die verdeckten Schichten dar (s. Abb 3.5). Im Gegensatz zur Ausgabeschicht ,bei der der Sollwert und und damit die Differenz zum Istwert vorgegebenen ist, muss diese für die vorhergehenden Schichten berechnet werden. Beim Backpropagation werden diese Differenzen, schichtenweise rückwärts berechnet. Ausgehend von der Ausgangsschicht, errechnen sich die Soll-Ist-Differenz eines Neurons der vorhergehenden Schicht aus den Differenzen aller Ausgabeneuronen, an deren Ergebnis es beteiligt ist. Da die Gewichtung des Eingänge die Bedeutung dieser für das

Ergebnis angibt, lässt sich Umgekehrt daraus auch die Beteiligung am Fehler bestimmen. Aus diesem Grund werden die Fehler mit der Gewichtung eines Eingangs multipliziert und über diesen Eingang zurückgereicht (Back Propagation). Ein Neuron der vorhergehenden Schicht bildet die Summer alle zurückgereichten Fehler und erhält damit seinen eigenen Fehler (Soll-,Ist-Differenz)

3.4 Der Dialogmanager Tapas

Für das Ausführen verschiedener Aktionen und den späteren Dialog des Robotersystems wird der Dialogmananger Tapas verwendet. Aus diesem Grund soll dieses System hier kurz beschrieben werden. Tapas basiert auf den in [Denecke (2002)] beschriebenen Dialogmodellen. Es unterstützt multimodale Kommunikation (z.B in Form einer Gestenerkennung, wie sie in [Stiefelhagen u. a. (2004)] beschrieben wird) und für eine multilinguale Verwendung ausgelegt[Holzapfel (2005)].
Tapas arbeitet informationsbezogen und zielorientiert. D.h es versucht die Absicht eines Benutzers zu deuten, indem es sie einem vorgegebenem Ziel zuordnet. Solange die Zuordnung nicht eindeutig möglich ist, werden die vorhandenen Teilinformationen dazu verwendet, die Zielauswahl einzuschränken. Durch zielgerichtetes Nachfragen, kann das System die Informationen, falls nötig in mehreren Schritten, komplettieren und so das Dialogziel (die Absicht des Benutzers) ausfindig machen.
Für dieses Vorgehen müssen dem System Beschreibungen der Ziele vorliegen. Darin werden einerseits die für ein Ziel benötigen Informationen und Bedingungen festgehalten. Andererseits wird angegeben, was die Folge dieses Zustandes ist. Ein solches Ziel kann zum Beispiel darin bestehen, ein bestimmtes Lied abzuspielen. Damit das System dieses Ziel erreicht müssen mehrere Bedingungen erfüllt sein. Der Wunsch des Benutzers (ein Lied zu hören) muss festgestellt werden und alle benötigten Informationen müssen vorhanden sein, wie zum Beispiel der Titel und die Lautstärke. Als Folge kann das System die für dieses Ziel vorgesehene Aktion ausführen und ein Lied abspielen.
Neben den Zielen beinhaltet das System auch so genannte *moves*. Diese sind sozusagen Vorstufen eines Ziels und dienen dazu den Dialog zu steuern, indem zum Beispiel Rückfragen gestellt werden. Die Auswahl eines *moves* hängt vom abstrakten Dialogzustand ab, welcher durch die Systemvariablen bestimmt wird. Eine genaue Beschreibung der abstrakten Dialogzustände findet sich in [Holzapfel u. Gieselmann (2004)].

1. Intention
2. SpeechAct
3. SelectetGoals
4. FinalizedGoals

Die Variable *Intention* beschreibt die Zustände der Ziele. Dies sind die folgenden vier: *deselected, selected, determined, finalized. Selected* sind alle Dialogziele, die in der augenblicklichen Dialogsituation zur Auswahl stehen, aber noch nicht vollständig sind. Als *determined* gilt ein Ziel, wenn es das einzige ist, welches noch in Betracht kommt, aber wie schon zuvor noch Informationen fehlen. *Finalized* ist ein Ziel letztendlich, wenn es erreicht ist. Das heißt es ist eindeutig bestimmt und alle benötigten Informationen sind

vorhanden. *Deselected* ist das Komplement zu *selected*, also alle Ziele, welche nicht mehr in Frage kommen. Die Variable *SelectedGoals* enthält alle Ziele welche sich im Zustand *selected* befinden. Wenn dies nur für ein Ziel zutrifft, bekommt dieses Ziel den Zustand *determined*. Die Variable *Finalized Goals* enthält dementsprechend alle Ziele im Zustand *finalized*. Ein *move* kann als Dialogschritt zum Erreichen des Ziels betrachtet werden. Er wird abhängig von den Systemvariablen und weiteren Variabeln, in welchen der Dialogkontext gespeichert wird, ausgeführt. So wird ein *move* zum Beispiel ausgeführt, sobald ein bestimmtes Ziel ausgewählt wird („Spiele ein Lied") (der Zustand der Variable *Intention* wechselt zu *selected*) und ein bestimmter Dialogkontext gegeben ist (die Kontextvariable mit dem gewünschten Titel ist noch nicht belegt). Dieser *move* würde dann dazu verwendet, die fehlenden Informationen zu erfragen.

4. Konzepte und deren Umsetzung

In diesem Kapitel soll das grundlegende Konzept des Systems und dessen Umsetzung dargelegt werden. Dazu wird im ersten Abschnitt der Aufbau des Gesamtsystems und die verwendete Roboterplattform beschrieben. In den darauf folgenden Abschnitten werden die einzelnen Teilsysteme erklärt. Diese Abschnitte bestehen jeweils aus zwei Teilen. Im ersten Teil wird das Konzept skizziert und im zweiten Teil wird die Umsetzung der einzelnen Konzepte dargestellt. Dabei werden Details der Implementierung erklärt und es werden Probleme aufgezeigt, die sich während der Umsetzung der Konzepte ergaben. Außerdem werden Möglichkeiten zur Lösung dieser Probleme vorgestellt.

4.1 Gesamtsystem

Dieser Abschnitt soll einen Überblick über das Gesamtsystem geben. Dazu wird einerseits der konzeptuelle Aufbau der Software beschrieben und andererseits wird die Roboterplattform vorgestellt, auf der das System aufgebaut wurde.

4.1.1 Gesamtkonzept

Das Ziel dieses Systems ist es, einen Dialog zwischen einem Benutzer und dem Roboter zu initiieren. Dazu werden schrittweise zuerst Informationen gesammelt und dann Aktionen ausgeführt. Entsprechend diesem schrittweisen Vorgehen lässt sich das Gesamtsystem in Module zerlegen. Diese Module sind, wie in Abbildung 4.1 zu erkennen, folgende: Videosystem, Klassifikator, Benutzermodell, Dialogmanager, Aktoren und Sensoren.

Zur Beobachtung der Umwelt kommen die Videokameras zum Einsatz, mit Hilfe derer Personen erfasst und verfolgt werden sollen. Dazu wird das Videobild nach Gesichtern durchsucht. Zur Verfolgung (Tracking) werden Gesichter in aufeinander folgenden Bildern in Zusammenhang gebracht und so ein Track erstellt. Die zweite Videokamera ermöglicht dann eine 3D-Berechnung des Kamerabildes. Somit lassen sich zu den Gesichtern dreidimensionale Koordinaten bestimmen. Da dies in Abhängigkeit der Bildqualität nicht für jedes Gesicht möglich ist, werden die Koordinaten in den übrigen Fällen aufgrund der Größe und Lage der Gesichter im Videobild geschätzt. Die Koordinaten werden einer eindeutigen TrackId zugewiesen und werden dann im Klassifikationsschritt dazu verwendet, ein grundlegendes Interesse oder Desinteresse zu klassifizieren. An diesem

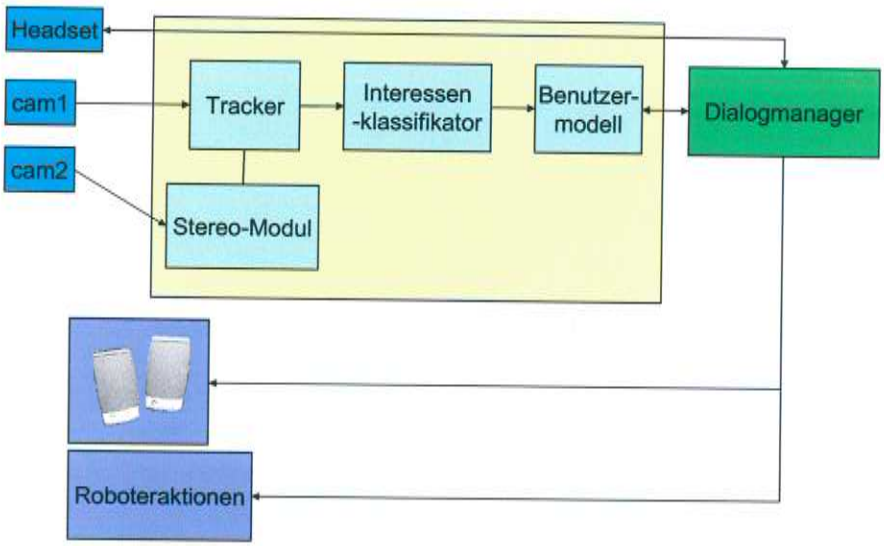

Abbildung 4.1: Das System ist in die folgenden Module gegliedert: Videosystem, Klassifikator, Benutzermodell, Dialogmanager, Aktoren und Sensoren.

Punkt werden die Informationen im Benutzermodell gespeichert und die Verarbeitung beginnt mit dem nächsten Bild von vorne. Das Benutzermodell wird regelmäßig überprüft, um dann ,wenn bestimmte Bedingungen erfüllt sind, Aktionen auszuführen. Diese Bedingungen hängen einerseits damit zusammen, wie lange ein Benutzer sich schon in einem Zustand befindet, das heißt wie lange er zum Beispiel schon vor dem Roboter steht. Andererseits ist auch wichtig, wie oft gewisse Aktionen schon ausgeführt wurden. Die einzelnen Konzepte dieser Teilschritte und deren Umsetzung werden in den folgenden Abschnitten genauer erläutert. Dazu ist der Rest dieses Kapitels in vier weitere Abschnitte aufgeteilt: Personenverfolgung (Videosystem), Klassifikation, Benutzermodell, Aktionen des Roboters.

4.1.2 Roboterplattform

Das System wurde auf der Entwicklungsplattform für den im Sonderforschungsbereich Humanoide Roboter an der Universität Karlsruhe (SFB 588[1]) eingesetzten Roboter ARMAR (siehe [Asfour u. a. (2001)] und [Stiefelhagen u. a. (2004)]) entwickelt und getestet. Die am Roboterkopf angebrachten Stereokameras werden dazu verwendet Personen zu erfassen und zu beobachten. Um auf Benutzer zu reagieren, können über einen Lautsprecher Geräusche und Sprache ausgegeben werden. Außerdem kann der Kopf des Roboters geneigt und gedreht werden. Die Bewegung der Roboterplattform war im Rahmen dieser Arbeit nicht vorgesehen. Die Software lief während der Experimente auf zwei Computern. Bis auf den Dialogmanager lief das System auf einem Barebone PC. Daher wurden der

[1] http://www.sfb588.uni-karlsruhe.de/

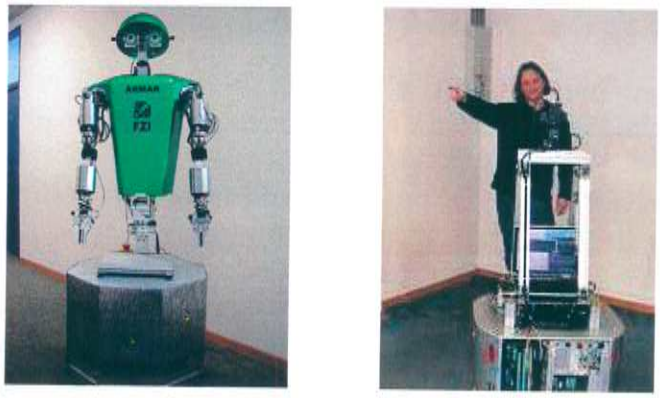

Abbildung 4.2: Roboter ARMAR (links) und Entwicklungssystem (rechts) (siehe [Asfour u. a. (2001)] und [Stiefelhagen u. a. (2004)]).

Roboter für die Tests mit dem Stromnetz verbunden. Der Dialogmanager wurde auf einem separaten Laptop ausgeführt, um Verzögerungen bei der Sprachausgabe zu vermeiden.

4.2 Videobasierte Personenverfolgung

Dieser Abschnitt beschreibt den Aufbau der Personenverfolgung. Dazu wird zuerst das zugrundeliegende Konzept beschrieben. Anschließend wird die Umsetzung dieses Konzepts in zwei getrennten Abschnitten erklärt. Der erste Abschnitt behandelt die zweidimensionale Personenverfolgung und im zweiten Abschnitt werden Details der 3D-Berechnungen aufgeführt.

4.2.1 Konzept der Personenverfolgung

Für die Videoverarbeitung stehen zwei Kameras zur Verfügung, welche durch eine Kalibrierung zur Berechnung von 3D-Koordinaten verwendet werden können. Entsprechend der Anforderung an das System, auch mehrere Personen verfolgen zu können, wurde für das Tracking ein im Rahmen des CHIL-Projektes[2] entwickelter Multi-Personen-Tracker verwendet. Dieser arbeitet zweidimensional und verwendet lediglich die Bilder einer Kamera. Dabei werden die Bilder mit Hilfe einer auf „Haar-Kaskaden" [Rowley u. a. (1996)] basierenden Gesichtserkennung [Viola u. Jones (2001)] nach Gesichtern durchsucht. Gesichter, deren Position und Größe sich in aufeinanderfolgenden Bildern nur geringfügig ändert, werden gruppiert und so zu einem Track zusammengefasst. Auf diese Weise entstehen Tracks, welche die Bewegungen der Gesichter im Videobild beschreiben. Die Tracks bestehen aus Trackpunkten, welche jeweils die Lage und Größe des erfassten Gesichtes, sowie den dazugehörigen Zeitstempel enthalten. Um den Zusammenhang zwischen den Trackpunkten zu erhalten, wird diesen noch eine eindeutige Track-Id zugeordnet.

Die erfassten Trackpunkte liefern noch keine Information über die Position im Raum. Da für die folgende Bewertung des Interesses aber die Position einer Person von Bedeutung ist, müssen die zweidimensionalen in dreidimensionale Tracks umgewandelt werden. Dabei findet die zweite Kamera Verwendung. Durch den Versatz zwischen den beiden aufgenommenen Stereobildern lassen sich zu einem Objekt im Videobild dessen dreidimensionale Koordinaten bestimmen (siehe Kapitel 3).

[2] http://chil.server.de

4.2.2 Umsetzung des „Multi-Personen-Trackers“

Der „Multi-Personen-Tracker“ liefert eine Liste von Rechtecken, welche die erfassten Gesichter im Videobild umschließen. Um dies zu erreichen werden die einzelnen Videobilder mit Hilfe des in [Viola u. Jones (2001)] beschriebenen Verfahrens nach Gesichtern durchsucht. Für das eigentliche Tracking werden die neu erfassten Gesichter mit den zuletzt erfassten bezüglich ihrer Position und Größe verglichen. Wenn auf diese Weise ein zusammengehörendes Paar aus altem und aktuellem Trackpunkt gefunden wurde, wird daraus, je nach Lernrate des Systems, ein neuer Punkt zwischen den beiden berechnet und als aktuell gespeichert. Die Lernrate liegt zwischen 0 und 1. Bei einem Wert von 1 geht der von der Gesichtserkennung erfasste Punkt voll in den neuen Trackpunkt über. Je näher der Wert bei 0 ist, desto stärker fällt der alte Trackpunkt ins Gewicht. Dadurch reagiert das System weniger empfindlich. Durch dieses Vorgehen lassen sich Gesichter über die Zeit verfolgen und zu Tracks zusammenfassen. Diese Tracks lassen sich innerhalb des Multitrackers durch ihre Position in einem Array, welches die aktuellen Trackpunkte speichert, identifizieren. Die Größe dieses Arrays ist begrenzt und ergibt sich aus der Anzahl der maximal gleichzeitig zu verfolgenden Personen. Durch diese Beschränkung ist es nötig das Array bei Bedarf aufzuräumen, indem lange nicht aktualisierte Tracks gelöscht werden, um Platz für neue zu schaffen. Aus diesem Grund gilt die Eindeutigkeit der Tracks nur temporär, während ihrer „Lebenszeit“. Wenn ein alter Track aufgeräumt wird um Platz zu schaffen, kann ein neuer Track dieselbe Position einnehmen, wodurch die beiden nicht mehr voneinander zu unterscheiden sind. Für die folgenden Schritte des Systems ist es jedoch notwendig, die Benutzer zumindest für den Zeitraum einer möglichen Interaktion eindeutig zuordnen zu können. Das heißt solange das System einen Benutzer im Benutzermodell führt, muss dieser auch noch im Tracker vorhanden sein, da sonst dessen Platz von einem neuen Benutzer besetzt werden könnte, und er damit im Benutzermodell die Rolle des vorigen Benutzers einnehmen würde. Der Zeitraum in den die Tracks eindeutig sind, ließe sich durch eine großzügige Wahl dieses Array verlängern. Die benötigte Größe lässt sich dabei allerdings nicht genau bestimmen. Das größere Problem war jedoch, dass diese Datenstruktur nur zur internen Speicherung der Tracks dient. Als Ausgabe liefert der Tracker eine Liste, in welcher die Trackpunkte aller im aktuellen Videobild erfassten Personen aufgelistet ist. Hier geht die Eindeutigkeit schon verloren, wenn ein Gesicht nur für einen Frame verdeckt ist. Dadurch kann es sehr leicht passieren, dass das System während der Interaktion mit einem Benutzer diesen kurz verliert, und sofort eine andere Person seine Rolle einnimmt. Um dies zu vermeiden, wird jedem Track eine eindeutige TrackId zugeordnet. Diese ergibt sich aus der Konkatenation der Systemzeit beim ersten Auftauchen des Tracks und seiner internen Position im Array zu diesem Zeitpunkt.

$$trackId = xxxxxxxxyy \tag{4.1}$$

xxxxxxxx gibt den Zeitpunkt an, zu dem der Track das erste Mal erfasst wurde und yy ist der Index für die Stelle im Array, an welcher der neue Track eingeordnet wurde.

Nachdem der Tracker die Gesichter der Personen erfasst hat, ist es wichtig deren Position zu bestimmen.

4.2.3 Umsetzung der 3D-Berechnungen

Um die Position eines Objektes im Raum zu bestimmen, werden wie schon erwähnt die Bilder zweier Kameras verwendet. Zur 3D-Berechnung müssen die korrespondierenden

Punkte im linken und im rechten Bild gefunden werden. Dazu werden kleine quadratische Bildausschnitte verwendet, welche in beiden Bildern in Deckung gebracht werden. Um einen Zusammenhang zwischen zwei Bildausschnitten herzustellen, muss an dieser Stelle ein möglichst großer Informationsgehalt in Form von Kontrasten vorliegen. Durch dieses Verfahren lassen sich also keine Koordinaten in kontrastlosen Bereichen bestimmen.
Ziel ist es nun, einen Punkt des Gesichtes zu finden, zu welchem sich die Koordinaten bestimmen lassen. Problematisch dabei ist, dass der vom Tracker erfasste Gesichtsbereich nicht immer kontrastreich ist und dass der Ausschnitt nicht immer optimal zum Gesicht positioniert ist. So sind manche Ausschnitte zu groß, manche zu klein oder sogar verschoben, d.h. nicht mittig zum Gesicht ausgerichtet. Der Kontrast zwischen dem Gesicht und dem Hintergrund, welcher sich gut für eine Berechnung eigenen würde, ist zum Beispiel in vielen Ausschnitten gar nicht enthalten, da diese ein bisschen kleiner als das Gesicht sind. Um dennoch einen Punkt mit gültigen 3D-Koordinaten zu finden, werden mehrere Punkte im Bildausschnitt untersucht. Dazu werden zehn gleichmäßig über den Ausschnitt verteilte Zeilen durchlaufen. Ziel ist es dabei, gültige Punkte zu finden und die Gesichtspunkte von den Hintergrundpunkten zu unterscheiden. Dazu wird zu allen gültigen Punkten die Distanz berechnet. Dann wird der gesamte Bereich der gemessenen Distanzen (von der kleinsten bis zur größten Distanz) in 20cm große Abschnitte aufgeteilt. Die Punkte werden entsprechend ihrer Distanz diesen Abschnitten zugeordnet. Da die Punkte des Gesichts im Gegensatz zu den (wenigen) Hintergrundpunkten alle in einem kleinen Distanzbereich liegen, wird davon ausgegangen, dass diese im Abschnitt mit der größten Anzahl an Punkten, zu finden sind. Von diesen Punkten wird einer ausgewählt und zur Koordinatenberechnung herangezogen.
Durch das hier beschriebene Vorgehen lässt sich die Anzahl der 3D-Trackpunkte, welche dem System zur Verfügung gestellt werden, erhöhen. Allerdings gibt es immer noch einige Fälle, in denen der Tracker ein Gesicht erfasst, aber trotzdem keine Koordinaten berechnet werden können. So kann es wie schon erwähnt vorkommen, dass wegen der Lichtverhältnisse im kompletten Gesichtsausschnitt keine gültigen Punkte gefunden werden. Entsprechend der eingestellten Lernrate kann der erfasste Bildausschnitt bei schnellen Bewegungen so weit versetzt sein, dass das Gesicht darin nicht mehr enthalten ist. Da der Hintergrund in vielen Bereichen kontrastlos ist, gibt es in diesen Fälle meisten auch keine gültigen Koordinaten. Ein weiterer Problemfall sind Personen die zu nahe sind. Je näher ein Objekt ist, desto größer wird der Versatz zwischen den korrespondierenden Punkten. Dadurch steigt der Rechenaufwand und die Fehlerquote. Es zeigte sich, dass das System, wenn es einen maximalen Versatz von 64 Punkten berücksichtigt, bis zu einem Abstand von ca. 90cm funktioniert. Bei näheren Objekten liefert es falsche Ergebnisse, welche aber nicht als solche erkannt werden können, da sie im normalen Wertebereich liegen.
Um diese Probleme berücksichtigen zu können, wird die Position des Gesichts zusätzlich geschätzt. Als Grundlage für diese Schätzung dienten mehrere in einem Meter Entfernung aufgenommenen Gesichter. Mit diesen Werten lässt sich die durchschnittliche Breite S eines in einem Meter Entfernung aufgenommenen Gesichtes berechnen. Ausgehend von diesem Wert wird die Entfernung d_i eines Gesichtes mit Seitenlänge s_i folgendermaßen geschätzt (entsprechend dem Strahlensatz):

$$d_i = \frac{S}{s_i}$$

Mit Hilfe der Entfernung und der Abweichung des Rechtecks von der Mitte des Bildes lässt sich dann auch der Winkel zur Mittelgeraden schätzen. Dieser Wert wird als Merkmal

für die Klassifikation verwendet (siehe Kapitel 4.3.1). Zwei weitere Klassfikationsmerkmale sind Geschwindigkeit und Fehlwinkel. Diese werden bei allen geschätzten Werten gleich gesetzt. Der Fehlwinkel wird auf 0.0 gesetzt, da für dieses Merkmal eine Schätzung zu ungenau wäre. Da eine Schätzung oft in Fällen vorkommt, in denen der Benutzer sich zu schnell bewegt, wird die Geschwindigkeit auf $3.0\frac{m}{s}$ gesetzt.
Die zweite Situation in der eine Schätzung der Position nötig wird, ergibt sich aus dem schon erwähnten Mindestabstand, welcher für eine 3D-Berechnung eingehalten werden muss. Da die 3D-Berechnung in diesen Fällen aber einfach falsche Werte liefert anstatt gar kein Ergebnis, muss diese Situation erst festgestellt werden. Dazu wird die Entfernung ständig geschätzt. Wenn ein Benutzer aufgrund der Rechtecksgröße näher als 1 Meter zu sein scheint, wird der geschätzte Wert anstatt dem berechneten verwendet. In diesem Fall wird eine Geschwindigkeit von $0\frac{m}{s}$ angenommen.

4.3 Klassifikation des Interesses

Dieser Abschnitt befasst sich mit der Klassifikation des Benutzerinteresses. Wie in den vorherigen Abschnitten wird zuerst das Konzept und anschließend dessen Umsetzung erklärt.

4.3.1 Konzept der Klassifikation

Der Klassifikator bestimmt ausgehend von der Position und Bewegung einer Person deren Interesse am Roboter. Das Interesse wird in zwei Klassen aufgeteilt. Einerseits interessierte Personen, deren Aufmerksamkeit auf den Roboter gerichtet ist, und andererseits alle übrigen Personen, welche vom Videosystem erfaßt werden. In dieser Klasse wird ein mögliches Interesse angenommen. Nicht erfasste Personen sehen entweder in eine andere Richtung oder sie bewegen sich zu schnell, um erfasst zu werden. Diese Fälle werden als nicht interessiert betrachtet und werden nicht weiter berücksichtigt. Durch den Einsatz zusätzlicher Sensoren, wie zum Beispiel einem Bewegungsmelder, wäre es allerdings auch möglich diese Fälle zu erfassen und darauf zu reagieren. Zu diesem Zweck könnte auch ein Mikrofon verwendet werden, mit der Möglichkeit Sprache oder auch Schritte zu erhören. Für die Klassifikation wird ein neuronales Netz (siehe Kapitel 3) verwendet. Durch überwachtes Lernen wird dem Netz in einer Trainingsphase beigebracht, wie es die erfassten Personen in die Klassen einteilen soll. Dazu wurden die erfassten Personen, bzw. ihre Trackpunkte, anhand aufgenommener Videos manuell annotiert. So entsteht ein Datensatz, welcher die dreidimensionalen Koordinaten aller Trackpunkte inklusive ihrer jeweiligen Klassenzugehörigkeit enthält. Da der Zusammenhang zwischen den dreidimensionalen Koordinaten einer Person und ihrem möglichen Interesse recht schwer zu beurteilen ist, wurden spezielle Merkmale für das Neuronale Netz berechnet. Dafür werden die Daten aufbereitet, so dass sie aussagekräftiger für das Interesse eines Benutzers sind. Dadurch soll das Training des Netzes erleichtert werden, so dass auch mit recht wenigen Daten ein gutes Netz trainiert werden kann. Zur Bewertung des Interesses werden die folgenden vier Merkmale herangezogen.

1. Distanz:

   Die Entferung zwischen dem Benutzer und dem Roboter.

2. Winkel:

   Der Betrag des Winkels zwischen einer gedachten Mittelgeraden längs zum Blickfeld des Roboters und einer Geraden, die durch den Roboter und den Benutzer läuft. Dieses Merkmal drückt damit aus, wie zentral der Benutzer vor dem Roboter steht. Dadurch dass der Betrag des Winkels berechnet wird, unterscheidet die Klassifikation nicht zwischen links und rechts.

3. Geschwindigkeit:

   Die Geschwindigkeit mit der sich der Benutzer bewegt.

4. Fehlwinkel:

   Der Differenzwinkel zwischen der tatsächlichen Bewegungsrichtung des Benutzers und der Richtung, in die er sich bewegen müsste, um direkt auf den Roboter zuzugehen.

Das Netz liefert kontinuierliche Werte zwischen 0 und 1. Diese Werte werden jedoch nur bei der Evaluation des Netzes genutzt. Für die weitere Verwendung wird das Ergebnis diskretisiert. Das heißt Werte kleiner als 0,5 werden zu 0 (nicht interessiert) und die übrigen Werte werden zu 1(interessiert).

4.3.2 Umsetzung der Klassifikation

Zur Klassifikation des Interesses wird ein neuronales Netz verwendet. Anfangs wurden mehrere Netztopologien ausprobiert. Die durchgeführten Experimente (Kapitel 5) haben ergeben, dass ein Netz mit zwei versteckten Schichten, welche jeweils vier Neuronen enthalten, die besten Ergebnisse liefert. Das Netz wurde mit dem Java-basierten Tool Joone erstellt und trainiert. Das trainierte Netz kann nach dem Training als Netz-Objekt exportiert werden und lässt sich auf diese Weise in einer Java Umgebung verwenden. Zur Integration des benötigten JAVA-Codes in Python wird JPype (http://jpype.sourceforge.net/) verwendet. Mit diesem Programm lässt sich in Python eine JVM starten und Java Code ausführen.

```
from jpype import *

def __init__(self):
        startJVM("/.../jdk1.5.0_04/.../libjvm.so","-Djava.class.path=.:....")
        self.javaCl = JClass('classification.Classifier')()

def classify(self,a,b,c,d):
        result = self.javaCl.getClassification(a,b,c,d)
        return result

def close(self):
        shutdownJVM()
```

4.4 Benutzermodell

In diesem Abschnitt werden der konzeptuelle Aufbau des Benutzermodells und die Details der Umsetzung dieses Konzeptes beschrieben.

4.4.1 Konzept des Benutzermodells

Die Entscheidung, wann welche Aktionen vom System ausgeführt werden, wird mit Hilfe eines Zusandsmodells getroffen. Der grundlegende Aufbau dieses Modells ist in Abbildung 4.3 zu sehen.
Das Modell beinhaltet einen *Vorzustand*, in welchem sich alle neuen Benutzer befinden, sobald sie in den Aufmerksamkeitsbereich des Roboters geraten. Ab einem vorgegebenen Alter werden sie, entsprechend des Klassifikationsergebnisses einem der folgenden Zustände (0 = nicht interessiert und 1 = interessiert) zugeordnet. Ziel des Systems ist es, den Benutzer in einen Dialog zu führen. Gelingt dies trotz wiederholter Versuche nicht, wird der betreffende Benutzer ignoriert.
Die hier aufgeführten Zustände beziehen sich jeweils auf einen Benutzer. Damit das System auch funktioniert, falls sich mehrere Personen im Aufmerksamkeitsbereich des Roboters befinden, wird immer nur eine der erfassten Personen als aktueller Benutzer ausgewählt. Für diesen Benutzer werden dann die Zustände modelliert und entsprechende Aktionen ausgeführt. Falls sich noch weitere Personen im Wahrnehmungsbereich des Roboters befinden, werden diese weiterhin verfolgt, allerdings werden sie bei der Auswahl der Aktionen nicht berücksichtigt.
Die Zustände des Systems sind von äußeren Umständen abhängig. Sie ergeben sich aus dem Verhalten des Benutzers. Auf dieses Verhalten kann der Roboter allerdings nur in gewissem Maß Einfluß nehmen. Dazu führt er eine Aktion aus die mit einer gewissen Wahrscheinlichkeit einen Zustandübergang herbei führt, indem sie den Benutzer auf geeignete Weise beeinflußt. Aus der Ungewissheit darüber, ob der gewünschte Zustandübergang stattfindet, ist es notwendig, je nach Erfolg oder Misserfolg einer Aktion diese zu wiederholen. Was auf der einen Seite der Vorteil solcher Wiederholungen ist, nämlich den Benutzer nachdrücklich in den gewünschten Zustand zu führen, kann auf der anderen Seite dazu führen den Benutzer zu belästigen oder, wenn eine Fehlerkennung vorliegt, auch das System zu blockieren, da ständig auf eine gar nicht vorhandene Person reagiert wird. Um diesen Problemen zu begegnen, war es notwendig Timer und Atkionszähler einzuführen. Damit der Roboter nicht zu aufdringlich wirkt, müssen zwischen den Wiederholungen der Aktionen bestimmte Pausen eingehalten werden. Außerdem wird die Anzahl der Wiederholungen gezählt. Somit können Fehlerkennungen entlarvt werden, da sich deren Zustand trotz mehrmaliger Einflussnahme nicht ändert. Falls für einen Benutzer eine bestimmte Anzahl an Aktionen ausgeführt wurden, ohne dass dieser seinen Zustand verändert hat, wird er in den *Ignoriert*-Zustand versetzt und wird nicht weiter beachtet. Das System geht wieder in den Startzustand über. Wenn sich jetzt noch weitere Personen im Aufmerksamkeitsbereich des Roboters befinden, wählt er aus diesen seinen nächsten Benutzer. Für ein solches Zustandsmodell müssen bestimmte Informationen zu den einzelnen Tracks gespeichert und aktualisiert werden, wie zum Beispiel das Alter des Tracks und die Art und Anzahl der ausgeführten Aktionen.

4.4.2 Umsetzung des Benutzermodells

Für die Umsetzung des Benutzermodells ist es nötig, verschiedene Informationen zu speichern. So muss das System eine Liste der erfassten Benutzer führen. Zu jedem Benutzer sollten außerdem Informationen bezüglich seines Zustands gespeichert werden. Dazu

wird in Python die Datenstruktur „Dictionary“ verwendet, welche als Schlüssel die BenutzerId erhält. Zusätzlich werden für jeden Benutzer folgende Informationen gespeichert.

1. t_1: Timer 1

   Hier wird der Zeitpunkt festgehalten, zu dem der Benutzer das erste mal erfasst wurde. Damit lässt sich jederzeit das Alter des Tracks berechnen.

2. t_2 Timer 2

   Zeitpunkt seit dem der Benutzer im Zustand 1 ist.

3. t_3 Timer 3

   Zeitpunkt seit der der Benutzer einen Dialog führt.

4. a_1 Aktionszähler 1

   Zähler für die Aktionen vom Typ 1.

5. a_2 Aktionszähler 2

   Zähler für die Aktionen vom Typ 2.

In Situationen, in denen das System mehrere Tracks erfasst hat, muss ein Benutzer ausgewählt werden, für den dann entschieden wird, welche Aktionen auszuführen sind. Die Auswahl sollte so funktionieren, dass es immer einen aktiven Benutzer gibt. Falls es mehrere Benutzer gibt, muss das System einen auswählen. Dieser ausgewählte Benutzer soll so lange aktiv bleiben, bis er entweder verschwunden ist, oder bis er vom System ignoriert wird. Aus diesem Grund wird bei der Auswahl eines Benutzers folgendermaßen vorgegangen. Wenn es noch keine ausgewählten Benutzer gibt, aber erfasste Tracks, wird einer dieser Tracks ausgewählt. Als Auswahlkriterium wird dabei die Abweichung des Trackpunkts von der Mittelgeraden verwendet. Je mittiger sich eine Person vor dem Roboter befindet, desto eher wird sie ausgewählt. Diese Auswahl eines Benutzers findet nach jedem Trackingschritt statt. Falls es schon einen ausgewählten Benutzer gibt, wird als erstes überprüft, ob dieser wieder vom Tracker erfasst werden konnte. Wenn dies der Fall ist wird überprüft, ob er ignoriert werden muss. Wird er nicht ignoriert, bleibt er als ausgewählter Benutzer bestehen. In den anderern Fällen wird die Auswahl zurückgesetzt und ein neuer Benutzer gesucht.
Für den ausgewählten Benutzer wird in jedem Durchlauf entschieden, welche Aktion durchzuführen ist. Die Handlungen des Roboters ergeben sich dabei aus dem aktuellen Zustand des Benutzers und dessen zeitlicher Entwicklung. Der genaue Ablauf hängt von den folgenden Werten ab.

1. ChangeStateTime:

   Solange das Alter des Benutzers kleiner als dieser Wert ist, ist der Benutzer im Vorzustand (siehe Abbildung 4.3). Er wird als neu betrachtet und seine Aufmerksamkeit soll mit einer Aktion vom Typ 0 auf den Roboter gelenkt werden.

2. Action0Time: Gibt an in welchem Alter das erste mal eine Aktion vom Typ 0 für einen Benutzer durchgeführt wird.

3. Action0Repeat:

   Gibt an nach welcher Zeit eine Aktion vom Typ 0 wiederholt wird.

4. Action1Time:

   Wie zuvor nur für Aktionen vom Typ 1

5. Action1Repeat:

   Wie zuvor nur für Aktionen vom Typ 1

6. Action2Time:

   Wie zuvor nur für Aktionen vom Typ 2

7. Action2Repeat:

   Wie zuvor nur für Aktionen vom Typ 2

8. Ignore0Counter:

   Gibt an nach wie vielen vergeblichen Versuchen, den Benutzer von Zustand 0 (uninteressiert) in Zustand 1(interessiert) zu führen, dieser ignoriert wird.

9. Ignore1Counter:

   Gibt das gleiche für die Versuche an den Benutzer von Zustand 1 in einen Dialog zu führen.

Abbildungen 4.3 und 4.4 veranschaulichen den Zusammenhang zwischen den Zuständen und Zeiten. Ein neuer Benutzer befindet sich anfangs im *Vorzustand*. Hier werden vom System, wenn die jeweiligen Wartezeiten abgelaufen sind, die Aktionen *a01* und *a02* ausgeführt. *a01* wird nach der Zeit *Action0Time* ausgeführt, danach wird jeweils nach der in *Action0Repeat* vorgegebenen Zeit Aktion *a02* wiederholt ausgeführt. Nach der durch *ChangeStateTime* vorgegebenen Dauer, wechselt der Zustand des Benutzers automatisch. Er geht dann in den vom Tracker aktuell erfassten Zustand über. Der Ablauf der Aktionen und Wiederholungen verläuft hier ebenso wie im *Vorzustand*. Nach einer bestimmten Anzahl vergeblicher Versuche den Zustand des Benutzers zu beeinflussen, wird dieser in einen *Ignoriert*-Zustand versetzt und nicht weiter beachtet.
Bei der Umsetzung des *Vorzustandes* gibt es zwei Möglichkeiten. Er kann, wie die anderen Zustände, benutzerabhängig implementiert werden, oder aber unabhängig vom Benutzer. Benutzerabhängig ist trivial, und wird durch einen Vergleich des *Benutzeralters* mit der *ChangeStateTime* erreicht. Ein Benutzer befindet sich so lange im *Vorzustand* bis er dieses vorgegebene Alter überschritten hat. Die benutzerunabhängige Vorgehensweise funktioniert für den ersten Benutzer genau so wie die benutzerabhängige. Der Unterschied ergibt sich, wenn ein ausgewählter Benutzer seinem Vorgänger ohne Pause folgt. Das heißt, wenn der aktive Benutzer wechselt, ohne dass dazwischen eine Phase war, in

der kein Benutzer erkannt wurde. Dieses Vorgehen soll in Fällen, in denen der Tracker einen Benutzer auf zwei Tracks verteilt, wie es zum Beispiel bei der Drehung des Kopfes oft vorkommt, die Möglichkeit bieten auf den vermeintlich neuen Benutzer sofort reagieren zu können.
Nachdem hier erklärt wurde, wie die Ausführung der einzelnen Aktionen geplant wird, sollen im nächsten Abschnitt die verschiedenen Aktionen vorgestellt werden.

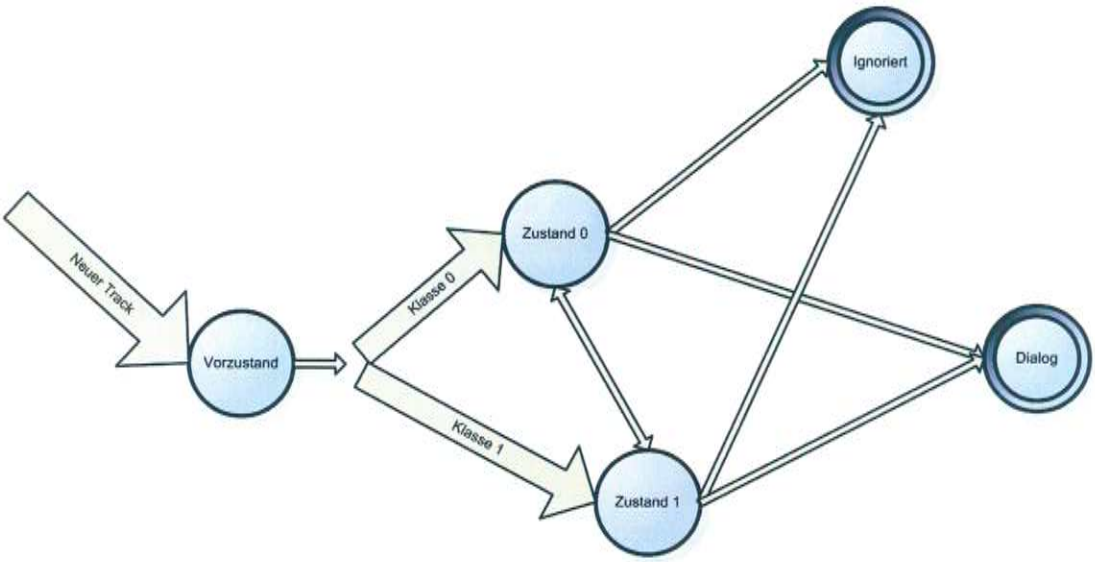

Abbildung 4.3: Zustände

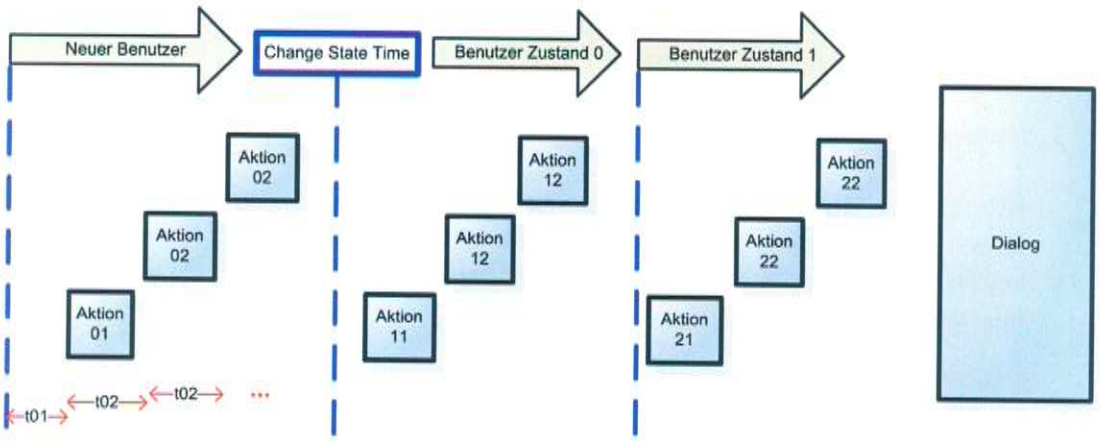

Abbildung 4.4: Zeitlicher Ablauf der Interaktion

4.5 Aktionen des Roboters

Dieser Abschnitt führt Aktionen auf, die verwendet werden um Interesse am Roboter zu wecken und den Benutzer in einen Dialog zu führen. Im ersten Teilabschnitt werden Aktionen vorgestellt, welche sich generell für diese Aufgaben eignen. Im zweiten Teilabschnitt werden dann die hier verwendeten Aktionen detailliert beschrieben.

4.5.1 Mögliche Aktionen

Zur Interaktion mit dem Benutzer ist der Roboter in der Lage, verschiedene Aktionen auszuführen. Ziel dieser Aktionen ist es die Aufmerksamkeit des Benutzers zu erlangen und diesen in einen Dialog zu führen. Aktionen die sich hierzu eignen lassen sich in drei Kategorien aufteilen.

1. Geräusche:

   Hier können künstliche Geräusche verwendet werden, aber auch natürliche, wie zum Beispiel Husten, Räuspern oder Pfeifen. Die meisten Geräusche, vor allem die künstlichen, werden nicht als personenbezogen empfunden. Wie sich in den Experimenten (siehe Kapitel 6) zeigte, eignen sich diese Geräusche dazu, ein allgemeines Interesse zu wecken. Allerdings reichen sie alleine nicht aus, um einen Benutzer zum Roboter zu führen. Im Gegensatz zu den meisten Geräuschen gibt es aber auch einige, die an eine Person gerichtet sind, wie zum Beispiel „jemandem nachpfeifen".

2. Verbale Äusserungen:

   Die Äußerungen des Roboters können personenbezogen oder allgemein sein. Der Bezug einer Aussage auf eine bestimmte Person muss sich dabei aus deren Inhalt ergeben. Der Roboter hat zwar die Möglichkeit, die Richtung seiner Aufmerksamkeit durch ein Drehen seines Kopfes anzudeuten. Allerdings hat er keine Möglichkeit seine Kommunikation an eine Person zu binden, wie dies Menschen zum Beispiel durch Blickkontakt tun. Daher soll dem Benutzer durch personenbezogene Äußerungen dieser Zusammenhang vermittelt werden. Solch eine Äußerung ist zum Beispiel („Hello! Please come closer!")

3. Optische Reize:

   Hier können Blinklichter verwendet werden, um Aufmerksamkeit zu erringen. Es ist aber auch möglich durch einen Avatar oder durch eine Bewegung des Roboterkopfes, dem Benutzer das Gefühl zu vermitteln, dass er vom Roboter wahrgenommen wird.

Für das hier verwendete System wurde die folgende Kombination von Aktionen ausgewählt. Zuerst sollte eine allgemeine, unpersönliche Aktion (Geräusch) ausgeführt werden, um die Achtsamkeit des Benutzers zu erhöhen. Danach soll diesem durch eine zielgerichtete Aktion gezeigt werden, dass der Roboter etwas von ihm möchte. Zur Unterstützung dieses Eindrucks wird die Drehung des Kopfes in die Richtung des Benutzers verwendet.

Um die Eignung verschiedener Aktionen für diese Aufgabe zu bewerten wurden Experimente durchgeführt, welche in den Kapiteln 5 und 6 aufgeführt sind.

4.5.2 Verwendete Aktionen und deren Integration in das System

Das hier vorgestellte System verwendet die folgenden Aktionen um die Aufmerksamkeit einer Person zu erringen und den Benutzer in einen Dialog zu führen:

1. Geräusche:

   Der Roboter hat bei den durchgeführten Experimenten als erste Aktion ein technisches Geräusch erzeugt. Dieses Geräusch entspricht dem aus Filmen bekannten Tauchalarm eines U-Bootes.

2. Verbale Äusserungen:

   Während der Interaktion mit einem Benutzer wurden die folgenden zwei Aussprachen des Roboters verwendet, um den Benutzer zu beeinflussen.
   1.„Hello! Please come closer!"
   2."Please use the Headset to say hello!"

3. Optische Reize:

   Als optischer Reiz diente dem Roboter die Drehung des Kopfes. Um dem Benutzer die Aufmerksamkeit des Roboters zu vermitteln, wurde der Kopf in Richtung des erfassten Benutzers gedreht. Dies wurde auf verschiedene Arten ausprobiert. Entweder indem die Position des Kopfes ständig angepasst, der Benutzer somit verfolgt wurde, oder indem der Kopf beim Erfassen eines neuen Benutzers einmal in dessen Richtung gedreht wird. Im ersten Fall ergibt sich für den Benutzer stärker der Eindruck, dass der Roboter auf ihn reagiert. Allerdings erhöht sich bei dieser Vorgehensweise auch die Fehlerquote des Videotrackers.

Für die Ausgabe der Geräusche und der Sprache wird das Dialogsystem *TAPAS* verwendet, welches wiederum für die Sprachausgabe ein TTS-Programm (Text To Speech) benötigt. Zusätzlich können je nach Anwendung weitere Module verwendet werden. Für die Kommunikation werden Nachrichten zwischen den Modulen ausgetauscht. Die einfachste Möglichkeit eine Eingabe an TAPAS zu übermitteln besteht darin, eine Textnachricht direkt über ein Socket zu senden. Außer dieser Möglichkeit unterstützt TAPAS auch integrierte Kommunikationsprotokolle. Dazu zählen *one4all* (ein am ISL - Universität Karlsruhe[3] entwickeltes System, welches eine FIPA/ACL[4] Nachrichtenstruktur verwendet.), *OAA* (Open Agent Architecture) und *koios* (an der Carnegie-Mellon University entwickelt[5]). Damit TAPAS die gewünschten Aktionen ausführt werden spezielle Nachrichten erstellt und diese an TAPAS übermittelt. Durch eine semantische Interpretation der Nachricht ordnet TAPAS diese einem festgelegten Ereignis zu und reagiert mit der entsprechenden Aktion. Es gibt zwei Arten von Aktionen, welche von TAPAS ausgelöst werden. Zum einen das Aussprechen von Wörtern. Dazu wird von TAPAS die Aussprache erzeugt und dann über ein sogenanntes TTS(Text to Speech)-System ausgegeben. TAPAS kann dabei mit verschiedenen TTS-Systemen kombiniert werden. Die zweite Art von Aktionen ist das Abspielen von Geräuschen. TAPAS kann zu diesem Zweck Audiodateien wiedergeben.
Neben diesen beiden Arten von Aktionen werden Bewegungen des Roboterkopfes als optischer Reiz eingesetzt. Der Kopf lässt sich in zwei Achsen bewegen (horizontal und vertikal). Zum Einstellen der Position lässt sich für beide Achsen der Winkel und die Geschwindigkeit, mit welcher dieser Winkel angefahren wird, angeben. Während der durchgeführten Tests wurde der Kopf nur in der horizontalen Ebene bewegt.

[3] http://isl.ira.uka.de/
[4] http://www.fipa.org/
[5] http://www.cmu.edu/

5. Experimente

Im Laufe der Arbeit wurden mehrere Experimente durchgeführt. Zu Beginn mussten Videodaten gesammelt werden, um den Trackingalgorithmus zu testen. Die ersten guten Trackingdaten wurden dann zum Trainieren des Klassifikators verwendet. Es zeigte sich beim Testen des Trackers, dass dessen Geschwindigkeit sehr gering ist. So lief das System lediglich mit einem Bild pro Sekunde. Die Videos für den Klassifikator wurden daher vorab aufgenommen und nachträglich mit dem Tracker nach Personen durchsucht. Dadurch ergab sich eine Trackingrate von sechs Bildern pro Sekunde. Für die spätere Verwendung im Gesamsystem ist dieses nachträgliche Tracking nicht geeignet. Daher war eine Anpassung des Systems nötig, so dass mehr Trackpunkte pro Benutzer berücksichtigt werden können. Aufgrund dieser Anpassungen ergab sich ein weiteres Experiment zum Trainieren des Trackers. Anschließend sollte die grundlegende Aufmerksamkeit, welche der Roboter mit bestimmten Aktionen erringt, getestet werden. Zum Abschluss wurden Experimente mit dem Gesamtsystem durchgeführt. Während dieser letzten Experimente wurden Videodaten gespeichert, um damit weiteres Trainingsmaterial für den Klassifikator zu erhalten. Die durchgeführten Experimente lassen sich in drei Bereiche unterteilen:

1. Experimente für den Prototyp des Interessenklassifikators und für den fertigen Klassifikator

2. Experimente zur Aufmerksamkeit

3. Experimente zur Interaktion des Benutzers mit dem Gesamtsystem

Die Experimente werden in diesem Kapitel genauer beschrieben. In Kapitel 6 wird anschließend die Evaluation dieser Experimente vorgestellt.

5.1 Prototyp für Interessenklassifikation

Der Roboter wurde im Gang aufgestellt, um Stereovideos aufzunehmen. Die Gangbeleuchtung war eingeschaltet.

Ziel:
Ziel dieses Experiments war es, einen Datensatz für das Training des Klassifikators sowie diesen Klassifikator selbst zu erstellen.
Insgesamt wurden in 12 Min 3420 Frames aufgenommen (6 Frames pro Sekunde).
Zuerst wurden die aufgenommenen Videos mit dem MultiTracker nach Gesichtern durchsucht. Der Tracker lieferte 160 Frames. Nach dem Aussortieren der Fehlerkennungen waren noch 106 Frames übrig. Die so erfassten Tracks stammen von 17 verschiedenen Personen, welche jeweils einmal am Roboter vorbeigekommen sind. Die Trackpunkte wurden anschließend manuell annotiert. Dazu wurden die Tracks im Video gekennzeichnet, indem Rechtecke um die Gesichter gelegt wurden. Ausgehend von diesen Videos wurde den Trackpunkten jeweils ein grundlegendes Interesse oder Desinteresse zugeordnet. Der Datensatz wurde anschließend in einen Trainingssatz und einen Evaluationssatz aufgeteilt.

5.2 Endgültige Interessenklassifikation

Während der Arbeit stellte sich die Geschwindigkeit des Systems als großes Problem heraus. Es zeigte sich, dass der zeitliche Abstand, in dem der Klassifikator Trackpunkte geliefert bekommt, sehr wichtig für die Reaktionsfähigkeit des Gesamtsystems ist. Um die Anzahl der Trackpunkte pro Zeiteinheit zu erhöhen, wird der „MultiTracker" beschleunigt, indem wird nicht das ganze Bild nach Gesichtern durchsucht. Am oberen und am unteren Rand wird jeweils 1/8 ausgespart. Eine weitere Möglichkeit dem Klassifikator mehr Punkte zu liefern ergibt sich aus der Tatsache, dass nicht zu jedem Gesicht, das vom Tracker erfasst wird, die entsprechenden 3D-Koordinaten berechnet werden können. Aus diesem Grund werden die 3D-Koordinaten mit Hilfe der Gesichtsrechtecke geschätzt. Für den Datensatz dieses Klassifiaktor wurden Aufnahmen verwendet, welche während den Benutzertests erstellt wurden.

Ziel:
Ziel war es einen neuen Datensatz für den zweiten Klassifikator zu erstellen.
Für den zweiten Klassifikator wurden komplett neue Aufnahmen gemacht. Für diesen Klassifikator gelten andere Voraussetzungen, da er neben den errechneten Merkmalen auch geschätzte Werte als Eingabe erhält und somit eine Entscheidung ausgehend von ungenaueren Daten treffen muss. Die aufgenommenen Videos wurden wie zuvor annotiert. Insgesamt ergab sich durch die Berechnungen und Schätzungen ein Datensazt von 1000 Trackpunkten. Diese 1000 Trackpunkte bestehen zu 76,5% aus geschätzen Werten.

5.3 Aufmerksamkeit erringen

In dieser Arbeit wurden verschiedene Möglichkeiten untersucht die Aufmerksamkeit einer Person auf den Roboter zu lenken. Diese gliedern sich folgendermaßen:

1. Nicht verbale (natürliche/künstliche) Geräusche
2. Aussprachen
3. Optische Reize

Um die Auswirkungen der möglichen Aktionen auf die Aufmerksamkeit des Benutzers zu untersuchen, wurden drei Experimente durchgeführt. Im ersten wurde untersucht wie sehr ein Roboter auffällt, wenn er keine Aktionen ausführt. Im zweiten Experiment reagierte der Roboter dann mit einem Geräusch auf Personen und das dritte Experiment diente dazu, Aktionen aus den verschiedenen Bereichen zu vergleichen.

5.3.1 Aufmerksamkeit 1

Bei diesem Experiment wurde der Roboter im Gang aufgestellt (so wie auch in allen folgenden Experimenten). Sechs Personen, die sich in einem Raum am einen Ende des Ganges befanden und noch nicht am Roboter vorbeigekommen waren, wurden gebeten, sich in einem Raum am anderen Ende des Ganges zu melden. Dabei mussten sie zwangsläufig am Roboter vorbei gehen. Im zweiten Raum angekommen wurden sie dann befragt, ob ihnen der Roboter aufgefallen ist und ob sie ihn sich genauer angesehen haben (Interesse).

Ziel:
Ziel dieses Experiments ist es herauszufinden, ob und wie wichtig Aktionen des Roboters sind, um die Aufmerksamkeit des Benutzers zu erringen.

5.3.2 Aufmerksamkeit 2

Der Aufbau des Roboters entspricht wieder dem im vorherigen Experiment. Diesmal wurde aber niemand vorbei geschickt. Der Roboter sollte auf Personen, die zufällig vorbeikommen, reagieren. Als Reaktion sollte er, sobald er eine Person erfasst, einmal ein Geräusch machen. Dazu wurde ein technisches Geräusch verwendet. Zur Bewertung wurden Personen, auf die der Roboter reagiert hat, befragt. Insgesamt reagierte der Roboter auf zehn Personen.

Ziel:
Hier sollte im Unterschied zum vorigen Experiment die Wirkung einer Aktion des Roboters im Bezug auf die Aufmerksamkeit untersucht werden.

5.3.3 Aufmerksamkeit 3

Bei diesem Experiment wurden verschiedene Aktionen verglichen. Damit die äußeren Umstände bei den einzelnen Durchläufen vergleichbar sind, wurde die Situation bei diesem Experiment gestellt. Dazu wurden zehn Benutzer nacheinander gebeten, jeweils fünf mal am Roboter vorbei zu laufen und seine Reaktion zu bewerten. Die fünf Durchgänge pro Benutzer ergeben sich folgendermaßen. Es wurde zu jedem der einführend erwähnten Aktionsbereiche eine Aktion gewählt. Danach wurden zwei weitere Durchläufe mit einer Kombination aus zwei Aktionen gemacht.

1. Optischer Reiz: Der Roboter dreht den Kopf in die Richtung des Benutzers (eine einzige Bewegung).
2. Geräusch: Hierzu wurde ein künstliches, technisches Geräusch verwendet.
3. Sprache: Der Roboter hat den folgenden Satz gesprochen: „Please come closer!“
4. Kombination aus 1. und 2.

5. Kombination aus 1. und 3.

Ziel:
Ziel dieses Experiments ist der Vergleich verschiedener Aktionen. Neben den Einzelaktionen sollte auch untersucht werden, ob eine Kombination mehrerer Aktionen eine Verbesserung mit sich bringt.

5.4 Interaktion mit Benutzer

Der Roboter wurde im Gang aufgestellt. Fünf Benutzer wurden gebeten, den Gang entlang zu laufen und auf den Roboter zu reagieren, wie sie es ihrer Auffassung nach in einer nicht gestellten Situation getan hätten. Dabei wurde das System in vier verschiedenen Modi betrieben, wobei jeder Benutzer jeden Modus fünf mal getestet hat. Daraus ergeben sich 100 Durchgänge. Die vier verschiedenen Modi unterscheiden sich in den vom Roboter ausgeführten Aktionen und im jeweils verwendeten Vorzustand. Wie die Unterschiede im Detail aussehen zeigt die folgende Aufzählung. Die Bedeutung der Aufzählung ergibt sich aus den Abbildungen 4.3 und 4.4. Abbildung 5.1 zeigt zur Veranschaulichung einen Teil der vom Roboter während eines Durchgangs aufgenommenen Bilder.

1. Modus:

   a01 = Geräusch+Kopfdrehung(einmalig), a02 = –
   a11 = a12 = "Hello, please come closer!"
   a21 = "Use the Headset to say hello!", a22= "Thanks for your attention! Good Bye!"

2. 2.Modus:

   a01 = Geräusch+Kopfdrehung (einmalig), a02 = –
   a11 = a12 = "Hello, please come closer!"
   a21 = "Use the Headset to say hello!", a22="Thanks for your attention!"

   Der Vorzustand (siehe Abbildung 4.4) ist bei diesem Modus nicht an einen Benutzer gebunden.

3. 3.Modus

   a01 = Geräusch+Kopfdrehung (einmalig), a02 = Kopfdrehung (Verfolgen)
   a11 = a12 ="Hello, please come closer!"
   a21 ="Use the Headset to say hello!", a22 = "Thanks for your attention!"

   Der Vorzustand ist wie in Modus 2 nicht an einen Benutzer gebunden. In diesem Zustand wird der Kopf nicht nur einmalig gedreht, sondern folgt dem Benutzer.

4. 4.Modus
   a01 = a02 = Geräusch

a11 = a12 ="Hello, please come closer!"
a21="Use the Headset to say hello!", a22="Thanks for your attention!"

Dieser Modus arbeitet wieder mit einem benutzerunabhängigen Vorzustand. Außerdem wird hier komplett auf eine Bewegung des Kopfes verzichtet.

Die Timings sind in allen Modi gleich. Sie sollen hier der Vollständigkeit halber aufgeführt werden. Die Bedeutung der einzelnen Werte wird in 4.4.2 beschrieben.

1. ChangeStateTime = 3,5s
2. Action0Time = 0,5s
3. Action0Repeat = 1s
4. Action1Time = 0,5s
5. Action1Repeat = 12s
6. Action2Time = 5s
7. Action2Repeat = 15s

Abbildung 5.1: Bilder der Roboterkammera während eines Durchlaufs des Experiments.

Ziel:
Ziel dieses Experiments ist es, die Interaktion zwischen Roboter und Benutzer zu bewerten. Dabei soll nicht nur festgestellt werden, in wie vielen Fällen der Roboter sein Ziel erreicht und einen Dialog initiiert, sondern auch der Zusammenhang zwischen der Erkennungsrate des Trackers und den Erfolgsaussichten des System untersucht werden.
Zur späteren Auswertung des Experiments wurden mehrere Daten gespeichert. Die vom

Roboter ausgeführten Aktionen wurden in einer Logdatei gespeichert. Zusätzlich dazu wurde jeder vom Tracker erfasste Punkt gespeichert, sowie alle von der Kamera aufgenommenen Bilder. Dabei wurden die Log-Dateien durch Angabe des Videoframes, Zeitstempels und der Benutzerkennung indiziert, so dass sich die Dateien untereinander und mit dem Videomaterial vergleichen lassen.

6. Evaluation

Dieses Kapitel enthält die Evaluation zu den in Kapitel 5 vorgestellten Experimenten. Dabei soll das jeweilige Vorgehen bei der Auswertung der einzelnen Experimente beschrieben werden. Dazu gehört die Erklärung der verwendeten Evaluationsmaße und eine Beschreibung, wie diese berechnet werden. Des weiteren werden die hierbei entstandenen Ergebnisse vorgestellt und deren Bedeutung für das vorher gesetzte Ziel erarbeitet, beziehungsweise weiterführende Schlüsse gezogen. Wie Kapitel 5 ist auch dieses Kapitel entsprechend den einzelnen Experimenten aufgeteilt.

6.1 Prototyp für Interessenklassifikation

Während der Entwicklung des Systems wurden zu zwei Zeitpunkten Klassifikatoren trainiert. Am Anfang der Arbeit war auch aufgrund nicht allzu guter Erkennungsquoten des Trackers nur ein recht kleiner Datensatz zum Training des Klassifikators vorhanden. Daher wurde mit diesen Daten ein Prototyp für die Interessenklassifikation erstellt. Dieser sollte im Laufe der Arbeit mit zusätzlichen Daten aus späteren Experimenten verbessert werden. Allerdings ergaben sich im Laufe der Entwicklung Änderungen am Tracker, so dass ein komplett neuer Klassifikator erstellt wurde. Daher wird die Evaluation der beiden Klassifikatoren getrennt aufgeführt.
In diesem Abschnitt folgt das Training und die damit verbundene Evaluation des Prototyps, um dann im darauf folgenden Abschnitt den endgültigen Klassifikator zu untersuchen.

6.1.1 Beschreibung:

Es wurden Videos mit drei Bildern pro Sekunde aufgenommen. Die in diesen Videos erfassten Gesichter wurden durch Rechtecke hervorgehoben, um sie beim Durchsehen annotieren zu können. Dabei wurden die Personen entsprechend dem Eindruck, den sie beim Betrachten des Videos machten, in zwei Klassen eingeteilt. Die zwei Klassen sind, wie schon in Kapitel 5 beschrieben, interessierte und uninteressierte Personen. In der hier durchgeführten Evaluation wurden die so erlangten Daten mit verschiedenen neuronalen

Netzen getestet, um die beste Netztopologie zu finden.
Die Aufteilung der Daten sieht folgendermaßen aus:
Von den insgesamt 106 Trackpunkten wurden 80 für das Training des Netzes und die restlichen 26 als Evaluationssatz verwendet.
Die getesteten Netze weisen eine mehrschichtige vorwärtsgerichtete Topologie auf (mehrschichtiges Perzeptron[1]). Sie unterscheiden sich in der Anzahl der verdeckten Schichten und der darin enthaltenen Neuronen. Die Anzahl der Schichten und Neuronen der einzelnen Netze ergibt sich aus Tabelle 6.1. Das Training und die Messung der Fehler wurde mit dem Programm Joone[2] durchgeführt. Das Training wurde beendet, sobald sich der Fehler auf den Evaluationsdaten verschlechterte.

6.1.2 Berechnungen und Maße:

1. RMSE auf Evaluationsdaten

   Für den RMSE (Root Mean Square Error) wird der Durchschnitt über den Quadraten aller Fehler gebildet und aus diesem die Wurzel gezogen.

$$rmse = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}} \tag{6.1}$$

   wobei e_i die Differenz zwischen, dem i-ten Klassifikationsergebnis und dem gewünschten Wert ist.

2. Fehlerquote

   Zur Berechnung der Fehlerquote werden die kontinuierlichen Ergebnisse des Klassifikators diskretisiert, indem Klassifikationsergebnisse unter 0,5 als 0 (nicht interessiert) gewertet werden und die restlichen als 1 (interessiert). Somit können die richtigen Ergebnisse durch einen Vergleich mit den ebenfalls als 0 oder 1 annotierten Daten ausgezählt werden. Die Fehlerquote ergibt sich aus dem Verhältnis der Anzahl der falsch klassifizierten Trackpunkte zur Gesamtanzahl der Punkte.

6.1.3 Ergebnisse und Fazit:

Netztopologie	RMSE	Fehlerquote
1 verdeckte Schicht mit 4 Neuronen	0,245	7,7%
1 verdeckte Schicht mit 12 Neuronen	0,170	11,5%
2 verdeckte Schichten mit 4 Neuronen	0,140	3,8%
2 verdeckte Schichten mit 12 Neuronen	0,135	3,8%

Tabelle 6.1: Fehler der vier verschiedenen Netztopologien auf dem Evaluationssatz

[1] engl. multi-layer perceptron (MLP)
[2] http://www.jooneworld.com/

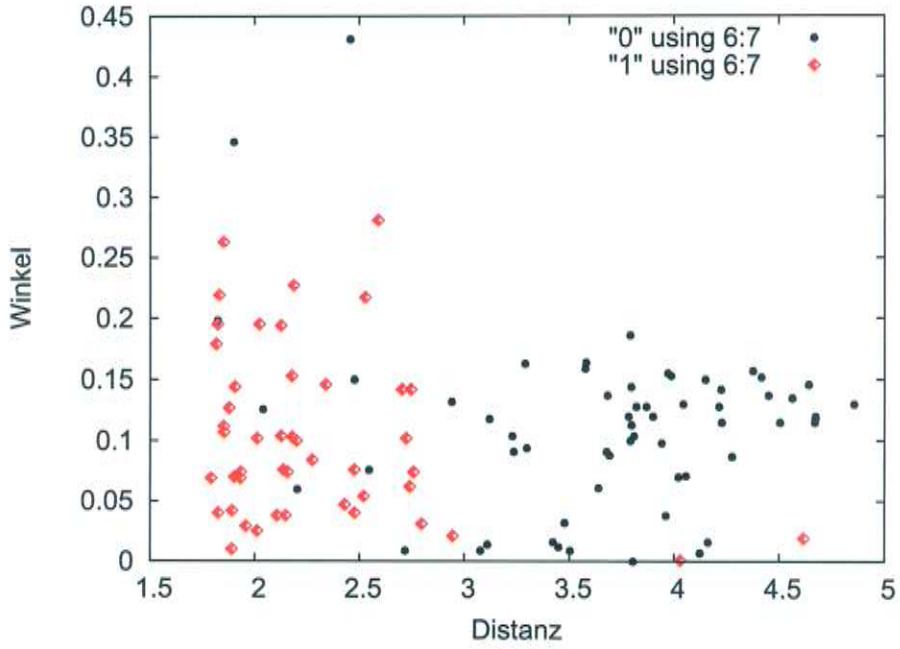

Abbildung 6.1: Beziehung zwischen Winkel, Distanz und Klassifikationsergebnis. Die roten Punkte sind interessierte Personen und die schwarzen uninteressierte.

1. Die Ergebnisse der Tabelle 6.1:

   Die Ergebnisse der Tabelle 6.1 zeigen, dass ein komplexeres Netz bessere Ergebnisse liefert. Allerdings zeigte sich während des Trainings, dass der Fehler auf den Evaluationsdaten sehr schnell schlechter wurde. Dies deutet darauf hin, dass der beim Training verwendete Datensatz zu klein ist. Daher wurde zu einem späteren Zeitpunkt ein erneutes Training des Klassifikators mit mehr Daten durchgeführt.

2. Abbildung 6.1:

   Diese Abbildung veranschaulicht die Merkmale Distanz und Winkel im Hinblick auf die Klassifikation. Die roten Punkte stehen für die interessierten Benutzer und die schwarzen für die uninteressierten. Es lässt sich annehmen, dass die Distanz ein aussagekräftiges Merkmal für die Klassifikation ist. Mann kann auch vermuten, dass Winkel größer als 0,3 (Bogenmaß) für ein Desinteresse sprechen.

6.2 Endgültige Interessenklassifikation

In diesem Abschnitt wird das Training und die damit verbundene Evaluation des zweiten Klassifikators beschrieben.

6.2.1 Beschreibung:

Im Unterschied zum ersten Klassifikator waren unter den hier verwendeten Trackpunkten auch einige mit geschätzten Werten enthalten. Ein weiterer Unterschied ist, dass die Aufnahmen während Benutzertests, in denen der Roboter Aktionen ausführte, gemacht wurden. Dies macht sich bei Aufnahmen, während der Kopf in Bewegung ist, durch verwischte Bilder bemerkbar. Insgesamt bestand der verwendete Datensatz aus 1000 Trackpunkten. Von diesen 1000 Trackpunkten wurden 235 durch das 3D-Modul berechnet, die restlichen 765 sind geschätzte Werte. Da die geschätzten Werte nicht so genau sind wie die berechneten, ist zu erwarten, dass dieser Klassifikator schlechtere Ergebnisse liefert als der Prototyp.

6.2.2 Berechnungen und Maße:

1. RMSE auf Evaluationsdaten:

   Für den RMSE (Root Mean Square Error) wird der Durchschnitt über den Quadraten aller Fehler gebildet und aus diesem dann die Wurzel gezogen (siehe Gleichung 6.1)

2. Fehlerquote:

   Zur Berechnung der Fehlerquote werden die kontinuierlichen Ergebnisse des Klassifikators diskretisiert, indem Klassifikationsergebnisse unter 0,5 als 0 (nicht interessiert) gewertet werden und die restlichen als 1 (interessiert). Somit können die richtigen Ergebnisse durch einen Vergleich mit den ebenfalls als 0 oder 1 annotierten Daten ausgezählt werden. Die Fehlerquote ergibt sich aus dem Verhältnis der Anzahl der falsch klassifizierten Trackpunkte zur Gesamtanzahl der Punkte.

Zum Training und der Evaluation des Netzes wurde der Datensatz im Verhältnis 80-10-10 aufgeteilt.

1. 80% (Training)
2. 10% (Evaluation 1 / Abbruchkritterium)
3. 10% (Evalation 2 / Bewertung des trainierten Netzes)

80% der Daten wurden für das Training des Netzes verwendet, während 10% (Evaluation 1) zur Überprüfung des Abruchkriteriums genutzt wurden. Dazu wurde nach jeweils zehn Lernzyklen der Fehler (RMSE) auf dem ersten Evaluationsdatensatz berechnet. Sobald dieser Fehler sich verschlechterte wurde das Training abgebrochen, in der Annahme, dass ab diesem Zeitpunkt der Traingsdatensatz auswendig gelernt wird und damit die gewünschte Generalisierung verloren geht. Zur Überprüfung des auf diese Weise erstellten Netzes wurde anschließend der Fehler (Fehlerquote) auf dem 2. Evaluationssatz

berechnet. Dieses Verfahren wurde insgesamt fünf mal durchgeführt, wobei die 20% Evaluationsdaten (200 Samples) dabei jeweils um 200 Samples verschoben wurden, womit jedes Sample insgesamt viermal zum Trainingssatz und einmal zum Evaluationssatz gehörte. Die Ergebnisse sind in Tabelle 6.2 zu sehen. Das Netz besteht aus zwei verdeckten Schichten mit jeweils vier Neuronen.

6.2.3 Ergebnisse und Fazit:

Training Nr.	RMSE (Eval1)	Fehlerquote (Eval2)	Anteil geschätzter Werte
1	0,111	29%	57%
2	0,095	0%	42%
3	0,124	38%	74%
4	0,117	2%	88%
5	0,111	4%	91%

Tabelle 6.2: Fehler des Klassifikators auf dem Evaluationsset

Die Ergebnisse des Netzes schwanken je nach Aufteilung des Datensatzes. Daher ist Anzunehmen, dass die Daten nicht „gleichmäßig" sind. Durch die in den Daten enthaltenen, ungenaueren Schätzwerte ergibt sich abhängig davon, wie diese aufgeteilt werden, ein besseres oder schlechteres Ergebnis. Ein Fehler von 38% im schlechtesten Fall liefert keine sichere Entscheidung. Allerdings ist das System auf mögliche Fehlentscheidungen vorbereitet, da es auch mit Fehlern des Trackers zu tun hat. Wichtig ist bei diesem endgültigen Klassifikator vielmehr, dass er eine wesentlich größere Menge an Daten vom Tracker erhält und somit das Gesamtsystem wesentlich schneller reagieren kann. Da nur 23,5% der Daten berechnete Werte sind, während der Rest geschätzt wurde, erhält der Tracker in diesem Fall vier mal so viele Trackpunkte. Abbildung 6.2 zeigt die Fehlerentwicklung während des Trainings des Klassifikators (Training Nr.5). Während der Fehler auf den Trainingsdaten ständig abnimmt, erreicht der Fehler auf den Evaluationsdaten nach ungefähr 500 Durchläufen sein Minimum und nimmt danach wieder zu. Das Mininum beträgt dabei (wie in Tabelle 6.2 zu sehen) 0,111. Das dabei erstellte Netz hatte auf dem zweiten Evaluationssatz eine Trefferquote von 96%.
Wenn man die Fehlerquote über alle fünf erstellten Netze betrachtet, zeigt sich, trotz geschätzter Werte ein gutes Ergebnis. Im besten Fall (2) wurde bei 100 Samples im 2. Evaluationssatz keine einzige Fehlentscheidung getroffen.
Da das System durch die geschätzten Werte viermal so viele Trackpunkten berücksichtigt, kann es wesentlich schneller und gleichmäßiger auf einen Benutzer reagieren. Die etwas schlechteren Klassifikationsergebnisse können für diesen Vorteil in Kauf genommen werden.

6.3 Aufmerksamkeit des Benutzers

Die Experimente, die in diesem Kapitel ausgewertet werden, dienen dazu die Wirkung verschiedener Aktionen im Bezug auf die Aufmerksamkeit des Benutzers zu testen. Wie schon in Kapitel 5 erwähnt, wurden aus diesem Grund drei Experimente durchgeführt. Bei den ersten zwei Experimente wurden die Benutzer vorher nicht informiert. Im dritten Experiment dagegen wurde eine Situation gestellt und die Benutzer wurden vorher informiert.

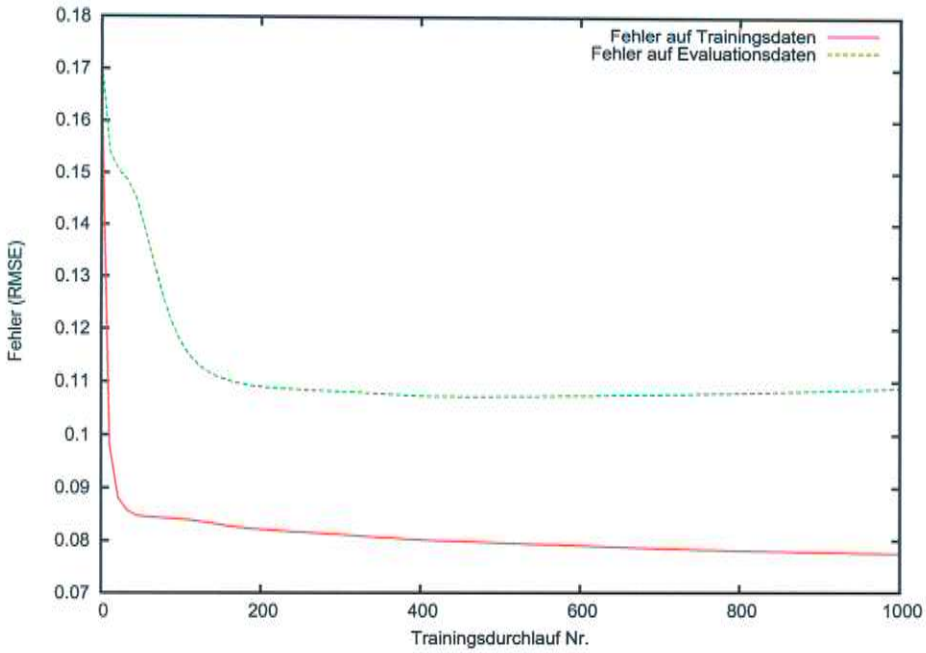

Abbildung 6.2: Fehlerentwicklung beim Training endgültigen Klassifikators auf den Trainingsdaten (rot) und den Evaluationsdaten (grün).

6.3.1 Experiment 1

6.3.1.1 Beschreibung:

Der Roboter wurde für dieses Experiment im Gang aufgestellt. Er sollte keine Aktionen ausführen. Dann wurden sechs Personen unter einem Vorwand gebeten sich in einem Raum am anderen Ende des Ganges zu melden, wodurch sie am Roboter vorbei kommen mussten. Anschließend wurden sie gefragt, ob ihnen im Gang ein Roboter aufgefallen ist.

6.3.1.2 Berechnungen und Maße:

Für die Auswertung dieses Experiment wurde gezählt, wie vielen Benutzern der Roboter aufgefallen ist.

Personen gesamt	„aufgefallen“	„Nicht Aufgefallen“	Anteil „Aufgefallen“
6	1	5	16,66%

Tabelle 6.3: Auffallen des Roboters

6.3.1.3 Ergebnisse und Fazit:

Was dieses Experiment trotz der geringen Anzahl an Benutzern zeigt, ist, dass ein Roboter nicht allein durch seine Anwesenheit auffällt. Wenn Personen mit etwas anderem

beschäftigt sind und sich nicht umsehen, wird ein Roboter nicht sonderlich wahrgenommen. Wenn der Roboter nicht wahrgenommen wird, kann auch kein Interesse an einem Dialog entstehen. Daher erscheint es notwendig für ein System, wie es hier erstellt werden sollt, dass der Roboter proaktiv handelt, um so das Interesse und die Aufmerksamkeit von Personen zu erwecken.

6.3.2 Experiment 2

6.3.2.1 Beschreibung:

Der Roboter stand im Gang und sollte auf Personen, die er erfasst, mit einem kurzen technischen Geräusch reagieren. Sobald der Roboter auf eine Person reagierte, wurde diese gefragt, wie sie die Reaktion aufgefasst hat. Insgesamt hat der Roboter dabei auf zehn Personen reagiert.

6.3.2.2 Berechnungen und Maße:

Als Maß diente bei diesem Experiment wieder das Empfinden der Personen. Dazu wurden sie gefragt, ob ihnen das Geräusch des Roboters aufgefallen ist und ob es ihnen so vorkam, als ob der Roboter auf sie reagiert hat. Diese Einschätzungen wurden ausgezählt und ins Verhältnis zur Anzahl aller Personen gesetzt.

6.3.2.3 Ergebnisse und Fazit:

Personen gesamt	Aktion bemerkt	Reaktion
10	60%	30%

Tabelle 6.4: Zehn Benutzer wurden gefragt, ob sie die Aktion des Roboters bemerkt haben und ob sie sie als Reaktion empfunden haben.

Das Ergebnisse in Tabelle 6.4 zeigen, dass der Roboter durch ein einfaches (einmaliges) Geräusch von 60% der Personen bemerkt wird. Im Unterschied zum vorigen Experiment ohne Geräusch (16%) ist dies eine große Steigerung. Die Aufmerksamkeit ließe sich durch ein lauteres Geräusches noch verstärken. Es besteht dann allerdings die Gefahr, dass das System sehr aufdringlich wirkt oder Personen nervt. Dies trifft besonders auf Personen zu, welche sich längere Zeit in der Nähe des Systems aufhalten, wie zum Beispiel Personen, die in der Nähe des Roboters arbeiten müssen.
Die dritte Spalte der Tabelle gibt an, dass nur 30% der Personen das Geräusch des Roboters als Reaktion wahrgenommen haben. Also hat nur die Hälfte derer, die das Geräusch gehört haben, es auf sich bezogen. In diesen Fällen wird der Roboter zwar bemerkt, aber das Interesse des Benutzers würde nicht gefördert werden. Daher wird es nötig sein weitere Aktionen zu verwenden, um einen Dialog zu initiieren.

6.3.3 Experiment 3

6.3.3.1 Beschreibung:

Im diesem Experiment wurden elf Personen gebeten am Roboter vorbei zu laufen und dessen Aktion zu bewerten. Dabei wurden jeweils fünf verschiedende Aktionen ausprobiert. Erstens das Drehen den Roboterkopfes in die Richtung der Person. Zweitens ein Geräusch des Roboters. Drittens ein „Hello!" des Roboters. Und viertens und fünftens jeweils die Kombination des Geräusches und der Sprache mit der Kopfdrehung.

6.3.3.2 Berechnungen und Maße:

Die Benutzer wurden gefragt, ob ihnen die einzelnen Aktionen aufgefallen sind und ob sie sie als Reaktion wahrgenommen haben. Zusätzlich sollten sie angeben, wie sie die Aktionen auf einer Skala von 1 (kaum auffällig) bis 3 (aufdringlich) bezüglich ihrer Auffälligkeit und auf einer Skala von 1 (schlecht) bis 3 (gut) bezüglich ihrer Eignung, den Benutzer in einen Dialog zu führen, bewerten. Zur Auswertung wurde jeweils das Verhältnis von den Aktionen, die aufgefallen sind, und denen, die als Reaktion wahrgenommenen wurden, zur Gesamtanzahl der Aktionen berechnet. Bei der Bewertung der Aufmerksamkeit und Eignung wurde der Durchschnitt aus den angegebenen Werten gebildet. Im vierten und fünften Durchgang wurden die Benutzer außerdem gefragt, ob die zusätzliche Drehung des Kopfes gegenüber den beiden vorherigen Durchgängen ohne diese Drehung eine Verbesserung darstellt.

6.3.3.3 Ergebnisse und Fazit:

Aktion	wahrgenommen	Reaktion	Auffälligkeit	Eignung	Verbesserung
1	100%	100%	0,90	1,72	
2	100%	100%	0,90	1,72	
3	100%	100%	0,90	2,81	
2+1	100%	100%	0,90	2,27	100%
3+1	100%	100%	1,0	2,90	100%

Tabelle 6.5: Elf Personen haben die verschiedenen Reaktionen des Roboters beurteilt.

Tabelle 8.3 zeigt, dass alle Aktionen wahrgenommen wurden. Dies liegt daran, dass die Benutzer auf eine Reaktion des Roboters vorbereitet waren. Außerdem fällt auf, dass sich die Aktionen kaum in der Bewertung ihrer Auffälligkeit unterscheiden. Damit sind alle Aktionen für einen ersten Schritt des Systems geeignet, in welchem die Aufmerksamkeit des Benutzers erlangt werden soll. Allerdings ist bei diesem Ergebnis zu erwähnen, dass die Bewegung des Kopfes mit einem Geräusch verbunden ist. Ohne dieses Geräusch ist anzunehmen, dass die Kopfdrehung weniger auffällt. Interessanter ist das Ergebnis bezüglich der Eignung einer Aktion, das Interesse eines Benutzers zu wecken und diesen in einen Dialog zu führen. In diesem Zusammenhang wurde die dritte Aktion als besonders gut geeignet bewertet. Außerdem waren alle Benutzer der Meinung, dass diese Eignung durch die zusätzliche Kopfdrehung verbessert wird.
Im Anhang finden sich fünf Tabellen (8.1, 8.2, 8.3, 8.4, 8.5), in denen die Durchläufe einzeln aufgeführt werden.

6.4 Benutzertests mit Gesamtsystem

Ziel dieses Experiments ist es, den Ablauf der Interaktion genauer zu betrachten und zu bewerten. Das System wurde in vier verschiedenen Modi (siehe Kapitel 5) getestet. Diese werden bezüglich ihrer Erfolgsquote verglichen. Dazu wurde gezählt, in wie vielen Fällen der Benutzer in einen Dialog geleitet wurde. Das Experiment soll außer dieser Erfolgsquote noch weitere Erkenntnisse liefern. Deshalb werden einzelne Aspekte der Interaktion genauer betrachtet, um einen möglichen Zusammenhang mit den besonderen Eigenschaften der einzelnen Modi aufzuzeigen. Bei diesem Experiment wurden für jedes Verfahren 25 Testdurchläufe mit fünf Benutzern durchgeführt, also jeweils fünf Durchgänge pro Benutzer pro Modus.

6.4.1 Quote begonnener Dialoge

6.4.1.1 Beschreibung:

Ziel dieser Auswertung ist es, den Erfolg der verschiedenen Verfahren zu vergleichen. Dafür wurde der Erfolg am Erreichen des Zielzustandes gemessen, also daran ob das System in der Lage ist, den Benutzer in einen Dialog zu führen. Da das System lediglich im Bezug auf die Initiierung eines Dialoges getestet werden sollte, wurde ein Durchlauf als Erfolg gewertet sobald ein Benutzer das Headset nahm, um den Dialog zu beginnen. Vor dem eigentlichen Dialog wurde der Durchlauf beendet, da der Dialog nicht Teil der Aufgabenstellung ist.
Der Begriff Erfolg ist in diesem Zusammenhang irreführend, da Fälle, in denen der Benutzer gar nicht an einem Dialog interessiert war und das System dementsprechend keinen Dialog initiiert hat, als Misserfolg betrachtet werden. Dabei wäre ein solcher Durchlauf im Sinne der Zielsetzung des Systems ein Erfolg. Diese Bewertung des Erfolgs liefert somit keine absolute Einschätzung der Leistungsfähigkeit des Systems. Sie ermöglicht aber eine tendentielle Bewertung und ein Vergleich der verschiedenen Modi.

6.4.1.2 Berechnungen und Maße:

1. Erfolgsquote pro Modus

   Hierzu wird die Anzahl der Durchläufe, die zu einem Dialog führten, pro Modus zusammengezählt und durch die Gesamtzahl der, in diesem Modus durchgeführten Durchläufe, geteilt.

2. Erfolgsquote pro Benutzer

   Ebenso so wie zuvor wird die Anzahl der Durchläufe, die zu einem Dialog führten, durch die Gesamtanzahl geteilt. Allerdings werden bei dieser Auswertung die Benutzer einzeln betrachtet.

3. Erfolgsquote bei den erfassten Personen

   Die Anzahl der Durchläufe, die zu einem Dialog führten, wird nicht durch alle geteilt, sondern jeweils nur durch die Anzahl der Durchläufe, in denen das System überhaupt eine Person erfasst hat.

6.4.1.3 Ergebnisse und Fazit:

Modus Nr.	Durchläufe mit Dialog	Erfolgsquote
1	8	32%
2	11	44%
3	11	44%
4	11	44%

Tabelle 6.6: Erfolgsquote 1

Benutzer	Erkennungsrate	Erfolgsquote
1	80%	35%
2	100%	85%
3	60%	30%
4	70%	15%
5	100%	50%

Tabelle 6.7: Erfolgsquote 2

Modus Nr.	Erkennungsrate	Erfolgsquote
1	76%	42%
2	80%	55%
3	76%	58%
4	96%	45%

Tabelle 6.8: Erfolgsquote3

Die Ergebnisse in Tabelle 6.6 liefern einen Anhaltspunkt, in wie vielen Fällen das System einen Dialog mit einem Benutzer beginnt. Hierbei sind die verschiedenen Verfahren einzelnen aufgeführt. Ein Ergebnis von 44% bei drei der vier Verfahren zeigt, dass das System noch nicht optimal ist. Allerdings lässt sich ein System, welches knapp die Hälfte der Benutzer in einen Dialog führt, in vielen Szenarien einsetzen. Vor allem ist es nicht das Ziel eines solchen Systems jede Person zu einem Dialog zu bewegen. Vielmehr sollen nur Benutzer, welche ein mögliches Interesse zeigen, in einen Dialog geführt werden.
Die folgende Tabelle 6.7 listet die Erfolgsquote im Bezug auf die einzelnen Benutzer auf. Dabei zeigt sich, dass das Ergebnis bei den verschiedenen Benutzern sehr unterschiedlich ausfällt. Das kann von der Art des Verhaltens der einzelnen Benutzer herrühren. Einige Personen sind ungeduldiger als andere und geben dem System weniger Zeit, bevor sie sich abwenden. Manche Personen bewegen sich viel vor dem Roboter, um ihn sich genau anzusehen, während andere still stehen. Ein weiterer Grund, der diese Unterschiede zwischen den einzelnen Benutzern erklärt, ist die Abhängigkeit des Trackers von äußeren Umständen. So haben sich zwischen den einzelnen Experimenten vielleicht die Lichtverhältnisse geändert oder der Tracker erkennt manche Gesichter schlechter als andere.

Modus Nr.	erfasste Gesichter	Anzahl Tracks
1	47,5	3,45 Tracks
2	29,8	3,10 Tracks
3	25,4	2,10 Tracks
4	53,2	2,75 Tracks

Tabelle 6.9: Qualität des Trackings

Um dies zu überprüfen wird in Tabelle 6.8 der Bezug zwischen der Erfolgsquote des Gesamtsystems und der Erkennungsrate des Tracker hergestellt. Dabei zeigt sich (wie zu erwarten), dass die Leistung des Gesamtsystems von der Erkennungsrate des Trackers abhängt. Aus diesem Grund wird in der dritten Spalte von Tabelle 6.8 die Erfolgsquote nur im Bezug auf die Durchläufe, in denen jemand erfasst wurde, berechnet. Dadurch lassen sich die verschiedenen Verfahren besser Vergleichen, da die ungleichen Voraussetzungen teilweise ausgeglichen werden. Allerdings wird dadurch nur ein Teil ausgeglichen, da nur die Fälle, in denen der Benutzer gar nicht erfasst wird, aus der Berechnung herausgenommen werden und die Fälle, in denen ein Benutzer schlecht (also sehr selten oder falsch) erfasst wird, trotzdem noch voll zählen.

6.4.2 Tracking

Da sich bei der vorigen Auswertung gezeigt hat, welchen Einfluss die Qualität des Trackings auf das System hat, soll dieser Zusammenhang hier genauer untersucht werden.

6.4.2.1 Beschreibung:

Bei dieser Auswertung geht es darum zu beurteilen, wie gut der Tracker bei den durchgeführten Experimenten funktioniert und ob sich ein Zusammenhang zum Erfolg des Systems zeigt.

1. Quote der erfassten Frames

   Die Anzahl der Frames, in welchen der Tracker ein Gesicht erfasst, wird durch die Anzahl aller Frames eines Durchlaufs geteilt. (Dazu zählen auch Frames, in denen der Benutzer gar nicht im Bild ist, obwohl er sich noch in der Interaktion mit dem Roboter befindet.)

2. Anzahl der Tracks pro Durchgang

   Es wird die durchschnittliche Anzahl unterschiedlicher Tracks, welche der Tracker bei einem Durchlauf erkennt, berechnet. Da bei jedem Durchgang nur ein Benutzer beteiligt war, sollte der Tracker im Idealfall auch nur einen zusammenhängenden Track erkennen.

6.4.2.2 Ergebnisse und Fazit:

Bezüglich der Anzahl der erfassten Gesichter liefern Modus 1 und 4 bessere Ergebnisse. Allerdings war (wie schon im vorigen Abschnitt erwähnt) bei den Experimenten im

Modus 4 eine allgemein bessere Erkennungsrate des Trackers gegeben, wovon das Ergebnis hier wahrscheinlich auch profitiert. Es ist anzunehmen, dass eine größere Anzahl erfasster Gesichter dem System eine bessere Grundlage gibt und somit auch ein besseres Gesamtergebnis liefert. Interessanterweise sind die Ergebnisse in Modus 1 und 4 gerade die schlechtesten (wenn man wie in Tabelle 6.8, nur die Durchläufe betrachtet in denen der Tracker überhaupt funktioniert hat). Dies zeigt, dass die gemeinsame Eigenschaft von Modus 2 und 3 (Zustand 0 nicht an Benutzer gebunden) eine Verbesserung bringt, die den Nachteil dieser Modi, dass der Benutzer seltener im erfassbaren Bereich des Roboters ist, mehr als ausgleicht.
Die durchschnittliche Anzahl verschiedener Tracks ist bei Modus 3 und 4 besser als bei den anderen beiden. Bei Modus 4 war dies zu erwarten, da sich der Kopf des Roboters nicht bewegt und der Benutzer in der meisten Zeit im Blickfeld des Roboters ist (was die großen Anzahl erfasster Gesichter zeigt). Bei Modus 3 bedeutet dies allerdings, dass trotz einer geringen Anzahl erfasster Gesichter, der Tracker in vielen Fällen einen Zusammenhang zwischen diesen herstellen konnte. Das zeigt, dass im Gegensatz zu Modus 1 und 2 das ständige Verfolgen der Person mit dem Kopf, das Tracking unterstützt, indem dadurch nicht so oft ein Track verloren wird und ein neuer begonnen werden muss.

6.4.3 Dauer der Interaktion und Anzahl der Aktionen

6.4.3.1 Beschreibung:

Bei dieser Auswertung werden die zeitlichen Abläufe der Testdurchläufe betrachtet , um dadurch eventuelle Möglichkeiten für Verbesserungen an den Timings des Systems aufzudecken. Die hier verwendeten Zeitangaben liegen in den Log-Dateien in Form von Indizes der Videoframes vor. Da jeder Frame aufgenommen wird und das System mit einer konstanten Framerate läuft, lassen sich daraus die absoluten Zeiten ermitteln.

6.4.3.2 Berechnungen und Maße:

1. Durchschnittliche Dauer:

   Für jeden Modus wird die Durchschnittliche Dauer (gemessen in Frames) der Interaktionen bestimmt. Dazu wurden zu jedem Durchgang dessen Anfang und Ende festgehalten. Als Anfang wurde der Zeitpunkt genommen, zu dem der Benutzer im Kamerabild auftaucht und als Ende der Zeitpunkt, zu dem der Benutzer den Roboter endgültig verlässt.

2. T2:

   Für die Erfolgreichen Durchläufe wird die durchschnittliche Dauer (gemessen in Frames) bis zum Erreichen des Zielzustandes berechnet.

3. T1:

   Durchschnittliche Dauer (gemessen in Frames) bis Aktion1 das erste mal ausgeführt wurde.

4. A1:

   Durchschnittliche Anzahl der benötigten Aktionen A1(„Please come closer!“) bis mit Aktion A2(„Use the headset to say hello!“) der Dialog begonnen werden konnte.

6.4.3.3 Ergebnisse und Fazit:

Modus Nr.	øDauer eines Durchlaufs	T2	T1	A1
1	45	48,57	14,2	0,5
2	57	53,45	24	0,63
3	41	47,6	16,5	0,54
4	40,7	34	9,8	1,27

Tabelle 6.10: Durchschnittliche Dauer der Benutzertests

Modus 2 benötigt am längsten, um den Benutzer in einen Dialog zu führen. Da Modus zwei (mit Modus 3) die besten Ergebnisse liefert, bedeutet dies, dass der Benutzer länger braucht bis er versteht, was der Roboter möchte, aber während dieser Zeit trotzdem nicht das Interesse verliert. Bei Modus 4 ist dies genau umgekehrt. Falls es gelingt einen Dialog zu initiieren, dann geht es verhältnismäßig schnell. Dafür verliert der Benutzer in diesem Modus am schnellsten das Interesse am Roboter (Spalte 2). Modus 4 verwendet in den erfolgreichen Fällen am meisten die Aktion 1 („Hello! Please come closer!“). Das lässt sich damit begründen, dass in den drei anderen Modi der Kopf gedreht wird. Dabei wird der Benutzer in vielen Fällen kurz vom Tracker verloren. Bis der Tracker den Benutzer wieder erfasst hat, ist dieser dann oft schon so nahe, dass direkt eine Aktion vom Typ 2 „Pleast use the Headset to say hello!“ ausgeführt wird.

7. Zusammenfassung und Ausblick

7.1 Zusammenfassung

Während dieser Arbeit wurde ein integriertes Robotersystem erstellt, welches in der Lage ist, selbständig einen Dialog mit einem Benutzer zu initiieren. Diese Zielsetzung unterscheidet das System von anderen Systemen, wie zum Beispiel dem in [Lang (2005)] vorgestellten System, bei dem der Roboter darauf wartet angesprochen zu werden. Bei dem in [Topp u. a. (2004)] vorgestellten System wird zwar der Benutzer angesprochen, allerdings muss er sich dazu vor den Roboter stellen und somit sein Interesse an einer Interaktion zeigen. In beiden Fällen geht die Initiative zum Dialog vom Benutzer aus. Im Gegensatz dazu ist das hier vorgestellte System in der Lage, auch Benutzer, die kein eindeutiges Interesse an einem Dialog zeigen, gezielt in einen solchen Dialog zu führen.
Allerdings sollen Personen, die gar kein Interesse am Roboter zeigen, auch nicht unnötig „belästigt" werden. Dazu wird das Interesse aller Personen, die der Roboter erfasst, bewertet und ausgehend von diesem Interesse entschieden, ob ein Dialog begonnen werden soll. Somit werden auch Personen berücksichtigt, die nur vielleicht an einem Dialog interessiert sind. Mit Hilfe verschiedener Roboteraktionen wird versucht, diese Personen auf den Roboter Aufmerksam zu machen. Wenn dies gelingt, wird weiter versucht einen Dialog mit dieser Person zu beginnen.
Ein weiterer Unterschied zu vielen anderen Systemen besteht in den verwendeten Sensoren. Hier werden lediglich zwei Videokameras zur Verfolgung der Personen eingesetzt. Dadurch ist das System auf vielen Roboter-Plattformen einsetzbar, da Videokameras in vielen solcher Systeme vorhanden sind.
Die durchgeführten Experimente haben gezeigt, dass es nötig ist die Aufmerksamkeit von Personen auf den Roboter zu lenken, da dieser in vielen Fällen sonst gar nicht wahrgenommen wird. Als besonders gute Aktion, um Aufmerksamkeit zu erwecken, erwies sich dabei ein vom Roboter abgespieltes Geräusch. Verstärkt wird diese Wirkung noch durch eine zusätzliche Drehung des Roboterkopfes in Richtung der erfassten Person. Um den Benutzer in einen Dialog zu führen reichen diese Aktionen allerdings nicht aus. Um dies zu erreichen werden dem Benutzer sprachliche Anweisungen gegeben. Bei den abschließend durchgeführten Benutzertests wurde in 44% der Durchläufe ein Dialog begonnen.

7.2 Ausblick

Während der Entwicklung des Systems und den damit verbundenen Experimenten zeigten sich außerdem Möglichkeiten zur Verbesserung und zur Erweiterung des Systems.
Verbesserungsmöglichkeiten ergeben sich besonders im Bezug auf die Robustheit der Personenverfolgung. Daher wäre die Integration weiterer Sensoren in das System interessant. Es wurden schon erste Versuche in Kombination mit einem System zur Sprecherlokalisierung gemacht, welches mit zwei am Roboterkopf angebrachten Mikrofonen funktioniert. Dadurch ließen sich Benutzer auch in Situationen erfassen, in denen sie den Roboter nicht anschauen. Außerdem ließe sich die Information, ob gesprochen wird, zur Beurteilung des Interesses verwenden. Wenn eine Person den Roboter ansieht und dieser aus der gleichen Richtung Sprache empfängt, kann er annehmen, dass er angesprochen wird. Eine „Geräuschlokalisierung" kann auch auf die Erkennung von Schritten trainiert werden, und so einen Benutzer anhand seiner Schritte lokalisieren.
Durch die Integration dieser Informationen würde die Verfolgung des Benutzers robuster, ebenso wie dies durch die Verwendung eines Laserscanners (wie auch in [Lang (2005)], [Topp u. a. (2004)], [Byers u. a. (2003)]) zur Unterstützung des Trackingverfahrens erreicht werden kann.
Eine Möglichkeit zur Erweiterung des Systems besteht darin, auf den erfassten Gesichtern eine Personenerkennung (PersonID) durchzuführen und so personenbezogen zu reagieren. Wie in der Einleitung schon erwähnt, ließen sich dadurch gezielt Benutzer ansprechen, um ihnen Nachrichten zu übermitteln.
Zunkünftig wären Experimente mit einem anschließenden Dialog interessant, da dies eine automatische Auswertung des Erfolgs ermöglichen würde. So ließen sich zum Beispiel durch „Reinforcement Learning" die Timings des Systems automatisch verbessern.

8. Literaturverzeichnis

Literatur

[AG]
AG, Neuro-Fuzzy: Einführung in Neuronale Netze. In: *http://wwwmath.uni-muenster.de/SoftComputing/lehre/material/wwwnnscript/*

[Asfour u. a. 2001]
Asfour, T. ; Ude, A. ; K.Berns ; Dillmann, R.: Control of armar for the realization of anthropomorphic motion patterns. In: *The second IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2001)* (2001), S. 22–24

[Boenke 2003]
Boenke, P.: Methoden der Mustererkennung und Klassifikation: Multi layer perceptron (MLP) und Backpropagation. In: *Seminar: Methoden der Mustererkennung und Klassifikation, TU Berlin* (2003)

[Byers u. a. 2003]
Byers, Zach ; Dixon, Michael ; Smart, William ; Grimm, Cindy: Say Cheese!: Experiences with a robotic photographer. In: *The Fifteenth Innovative Applications of Artificial Intelligence Conference (IAAI-03)* (2003)

[Byers u. a. 2004]
Byers, Zachary ; Dixon, Michael ; Smart, William D. ; Grimm, Cindy M.: Say Cheese!: Experiences with a Robot Photographer. In: *AI Magazine* 25 (2004), Nr. 3, S. 37–46. – This is an expanded vesion of the IAAI paper with the same name. It's a slighlty different version than the one that appears in AI Magazine. However, it's pretty close, and has almost exactly the same text.

[Crow 1984]
Crow, F.: Summed-area tables for texture mapping. In: *Proc. SIGGRAPH volume 18* (1984)

[Darell u. a. 1998]
Darell, T. ; Gordon, G. ; Woodfill, J. ; Harville, M.: A Virtual Mirror Interface

using Real-time Robust Face Tracking. In: *Third International Conference on Face and Gesture Recognition* (1998), April

[Denecke 2002]
DENECKE, M.: *Generische Interaktionsmuster für aufgabenorientierte Dialogsysteme*, Diss., 2002

[Fleck u. a. 1996]
FLECK, M. ; FORSYTH, D. ; BREGLER, C.: Finding Naked People. In: *European Conf. on Computer Vision* (1996)

[Freund u. Schapire 1995]
FREUND, Yoav ; SCHAPIRE, R.E: A decision-theoretic generalization of on-line linearning and an application to boosting. In: *Computational Learning Theory: Eurocolt* (1995)

[Fritsch u. a. 2002]
FRITSCH, J. ; LANG, S. ; KLEINHAGENBROCK, M. ; FINK, G.A. ; SAGERER, G.: Improving Adaptive Skin Color Segmentation by Incorporating Results from Face Detection. In: *IEEE Int. Workshop on Robot-Human Interactive Communication (ROMAN), IEEE Press, Berlin* (2002), April

[Holzapfel 2005]
HOLZAPFEL, Hartwig: Towards Development of Multilingual Spoken Dialogue Systems. In: *Proceedings of the 2nd Language and Technology Conference* (2005)

[Holzapfel u. Gieselmann 2004]
HOLZAPFEL, Hartwig ; GIESELMANN, Petra: A Way Out of Dead End Situations in Dialogue Systems for Human-Robot Interaction. In: *Humanoids 2004, Los Angeles* (2004)

[Jähne 1997]
JÄHNE, B: Digitale Bildverarbeitung. In: *Springer-Verlag, Berlin-Heidelberg 4. Auflage* (1997)

[Konolige 1997]
KONOLIGE, K: Small Vision Systems: Hardware and Implementation. In: *Eigth Int. Symposium on Robotics Research, Hayama* (1997)

[Kuno u. a. 2004]
KUNO, Yoshinori ; SAKURAI, Arihiro ; MIYAUCHI, Dai ; NAKAMURO, Akio: Two-way Eye Contact between Humans and Robots. In: *Sixth International Conference on Multimodal Interfaces (ICMI 2004)* (2004), October

[Lang 2005]
LANG, S.: *Multimodale Aufmerksamkeitsteuerung für eine mobilen Roboter*, Diss., April 2005

[Nickel 2003]
NICKEL, K.: Erkennung von Zeigegesten basierend auf 3D-Tracking von Kopf und Händen. In: *Diplomarbeit: Universität Karlsruhe, Institut für Logik, Komplexität und Deduktionssysteme* (2003), March

[Papageorgiou u. a. 1998]
PAPAGEORGIOU, C. ; OREN, M. ; POGGIO, T.: A general framework for object detection. In: *Int. Conf. on Computer Vision* (1998)

[Rowley u. a. 1996]
ROWLEY, H. ; BALJUA, S. ; KANADE, T.: Neural Network-Based Face Detection. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco* (1996)

[Schmid u. a. 2005]
SCHMID, A.J. ; SCHREMPF, O.C. ; HANEBECK, U.D. ; WÖRN, H.: A Novel Approach To Proactive Human-Robot Cooperation. In: *Proc. IEEE International Workshop on Robot and Human Interface Communications (RO-MAN 2005), Nashville* (2005)

[Schulz u. a. 2001]
SCHULZ, D. ; BURGARD, W. ; FOX, D. ; CREMERS, A.B.: Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association. In: *IEEE Int. Conf. on Robotics and Automation (ICRA), IEEE Press, New Orleans* (2001)

[Stiefelhagen u. a. 2004]
STIEFELHAGEN, R. ; FÜGEN, C. ; GIESELMANN, P. ; HOLZAPFEL, H. ; NICKEL, K. ; WAIBEL, A.: Natural Human-Robot Interaction using Speech, Gaze and Gestures. In: *Proceedings of the International Conference on Intelligent Robots and Systems* (2004)

[Topp u. a. 2004]
TOPP, E.A. ; KRAGIC, D. ; JENSFELT, P. ; CHRISTENSEN, H.I.: An interactive interface for service robots. In: *IEEE Int. Conf. on Robotics and Automation, IEEE Press, New Orleans* (2004)

[Viola u. Jones 2001]
VIOLA, Paul ; JONES, Michael: Robust Real-time Object Detection. In: *SECOND INTERNATIONAL WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF VISION - MODELING, LEARNING, COMPUTING, AND SAMPLING* (2001), July

[Walliczek 2005]
WALLICZEK, M.: Speaker Localization and Input Fusion on a Humanoid Robot. In: *Studienarbeit - Universität Karlsruhe, Carnegie-Mellon University Pittsburgh* (2005)

[Zabih u. Woodfill 1994]
ZABIH, R. ; WOODFILL, J.: Non-parametric Local Transforms for Computing Visual Correspondence. In: *Proc. of the third European Conf. on Computer Vision, Stockholm* (1994)

Teil I

Anhang

Benutzer Nr.	Aktion wahr-genommen	als Reaktion	Auffälligkeit	Dialog initi-ieren
1	x	x	1	0
2	x	x	1	1
3	x	x	1	0
4	x	x	1	1
5	x	x	1	1
6	x	x	1	1
7	x	x	1	0
8	x	x	0	0
9	x	x	1	2
10	x	x	1	1
11	x	x	1	1

Tabelle 8.1: Beurteilung der Drehung des Roboterkopfes durch elf Benutzer.

Benutzer Nr.	Aktion wahrgenommen	als Reaktion	Auffälligkeit	Dialog initiieren
1	x	x	1	0
2	x	x	1	1
3	x	x	1	0
4	x	x	1	1
5	x	x	1	1
6	x	x	1	1
7	x	x	1	0
8	x	x	0	0
9	x	x	1	2
10	x	x	1	1
11	x	x	1	1

Tabelle 8.2: Beurteilung eines Geräusches des Roboters durch elf Benutzer.

Benutzer Nr.	Aktion wahrgenommen	als Reaktion	Auffälligkeit	Dialog initiieren
1	x	x	1	1
2	x	x	1	2
3	x	x	1	2
4	x	x	1	2
5	x	x	0	2
6	x	x	1	2
7	x	x	1	2
8	x	x	1	2
9	x	x	1	1
10	x	x	1	2
11	x	x	1	2

Tabelle 8.3: Beurteilung eines „Hello!“ des Roboters durch elf Benutzer.

Benutzer Nr.	Aktion wahrgenommen	als Reaktion	Auffälligkeit	Dialog initiieren	besser
1	x	x	1	0	+
2	x	x	1	1	+
3	x	x	1	0	+
4	x	x	1	2	+
5	x	x	1	2	+
6	x	x	1	2	+
7	x	x	1	2	+
8	x	x	1	2	+
9	x	x	0	0	+
10	x	x	1	2	+
11	x	x	1	1	+

Tabelle 8.4: Beurteilung einer mit einem Geräusch kombinierten Kopfdrehung des Roboters durch elf Benutzer.

Benutzer Nr.	Aktion wahrgenommen	als Reaktion	Auffälligkeit	Dialog initiieren	besser
1	x	x	1	2	+
2	x	x	1	2	+
3	x	x	1	2	+
4	x	x	1	2	+
5	x	x	1	2	+
6	x	x	1	2	+
7	x	x	1	2	+
8	x	x	1	2	+
9	x	x	1	1	+
10	x	x	1	2	+
11	x	x	1	2	+

Tabelle 8.5: Beurteilung einer mit einem „Hello!“ kombinierten Kopfdrehung des Roboters durch elf Benutzer.