Universität Karlsruhe (TH)

Gesichteridentifikation auf Bildsequenzen in Mensch-Roboter-Interaktion

Studienarbeit am Institut Interactive Systems Labs
Prof. Dr. Alex Waibel
Fakultät für Informatik
Universität Karlsruhe (TH)

von

Stephan Könn
Matrikel-Nr.: 1105428

Betreuer:

Prof. Dr. Alex Waibel
Dipl.-Inform. Hartwig Holzapfel
Dipl.-Inform. Hazim K. Ekenel

Tag der Anmeldung: 1. August 2006
Tag der Abgabe: 31. Oktober 2006

Interactive Systems Labs 

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Roboter sind aus unserer heutigen modernen Gesellschaft kaum mehr wegzudenken. In der Industrie und Produktion übernehmen Automaten- und Robotersysteme vielfältige Aufgaben, die ein hohes Maß an Präzision erfordern oder schnell ausgeführt werden müssen. Desweiteren werden Erkundungsroboter eingesetzt, die an Orten operieren, die für den Menschen unzugänglich oder schlichtweg zu gefährlich sind, wie z.B. Konfliktgebiete oder die Mars-Oberfläche. Selbst die Spielzeugindustrie hat einen Markt für Roboter entdeckt und mit dem AIBO [16] einen Roboterhund geschaffen. Der langjährige Traum vieler Wissenschaftler und Forscher ist es allerdings, einen Roboter nach dem Vorbild des Menschen zu erschaffen. Abbildung 1.1 zeigt einige berühmte Vetreter dieser humanoiden Roboter. Von links nach rechts handelt es sich dabei um ARMAR, der am FZI in Karlsruhe entwickelt wird, ROBONAUT von der amerikanische Weltraumbehörde NASA und Hondas ASIMO.

Neben dem menschlich motivierten Äußeren, ist bei dieser Art von Roboter auch eine möglichst menschliche Verhaltensweise gewünscht. Der Maschinenmensch soll laufen, Gestik zeigen und sogar Gefühle empfinden können. Ein äußerst relevanter Aspekt ist in dieser Hinsicht die Kommunikation und Interaktion mit dem Menschen. Ist die Maschine in der Lage, Menschen zu erkennen und zu identfizieren, so ermöglicht dies dem Roboter, auf natürlichere Art und Weise mit Personen zu kommunizieren. Heutzutage existieren Systeme, die das Gesicht oder die Stimme eines Menschen wiedererkennen oder eine Identifikation anhand anderer biometrischer Merkmale vornehmen.

Schwerpunkt der Studienarbeit ist die Entwicklung eines Systems zur Gesichteridentifikation in der Mensch-Roboter-Interaktion basierend auf Bildsequenzen, sowie die Einbettung dieser Arbeit in ein vorhandenes, interaktives Robotersystem, welches in der Lage ist, sich zu bewegen, aktiv die Kommunikation zu suchen, Gestiken zu erkennen und mit Personen zu sprechen. Demzufolge stellt das hier präsentierte Gesichtsidentifikationssystem neben der eigentlichen Hypothese auch eine Konfidenz zur Verfügung, welche die Sicherheit in die getroffene Annahme ausdrückt.

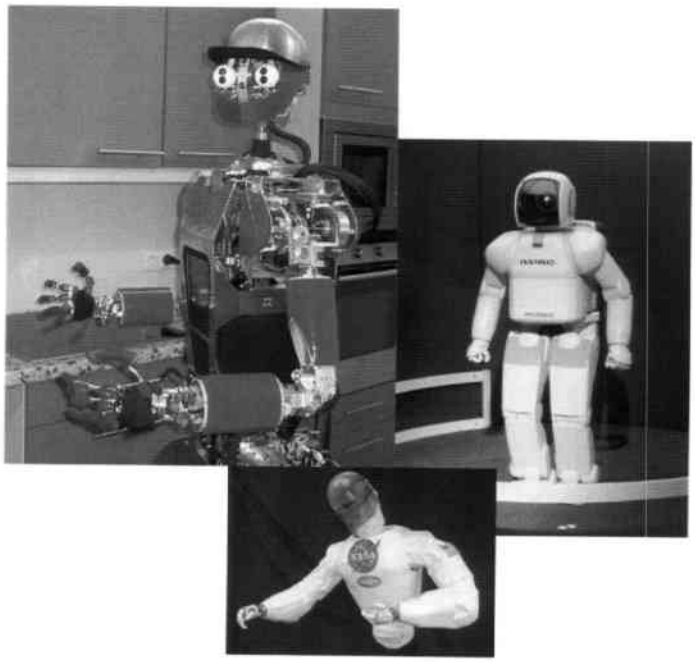

Abbildung 1.1: Berühmte Vertreter humanoider Roboter, von links nach rechts: ARMAR, ROBONAUT und ASIMO

1.2 Überblick

Die vorliegende Arbeit ist in mehrere Abschnitte unterteilt, die unterschiedliche inhaltliche Schwerpunkte setzen. Kapitel 2 erläutert grundlegende Verfahren, Algorithmen und Methoden, die in dieser Arbeit zum Einsatz kommen. Kapitel 3 beschreibt die Umsetzung und das Vorgehen bei der Gesichteridentifikation auf Bildsequenzen in der Mensch-Roboter-Interaktion. Kapitel 4 beschäftigt sich mit den gesammelten Daten und beschreibt die durchgeführten Experimente und Evaluationen. In Kapitel 5 werden die erzielten Ergebnisse diskutiert und Ansätze für zukünftige Arbeiten aufgezeigt. Schließlich erfolgt in Kapitel 6 eine Zusammenfassung der gesamten Arbeit.

Kapitel 2

Grundlagen

Dieses Kapitel verschafft einen kurzen Überblick über die verwendeten Komponenten, Methoden und Algorithmen, die im Laufe dieser Arbeit zum Einsatz kamen. Der erste, zweite und dritte Teil beschreibt dabei Vorgehensweisen die bei der Bildvorverarbeitung verwendet werden. Der vierte Teil beschreibt einen Algorithmus, mit dem neue Instanzen klassifiziert werden können. Abschließend wird ein Verfahren vorgestellt, mit dessen Hilfe Konfidenzen berechnet werden.

2.1 openCV und Haarklasifikatoren

OpenCV [1] ist eine kostenlose Programmbibliothek von Intel®. Sie ist für die Programmiersprachen C und C++ geschrieben und enthält Algorithmen für die Bildverarbeitung und Computer Vision. Im August 2005 wurde die Version 0.97 herausgegeben, die während dieser Arbeit zum Einsatz kam. Intel® proklamiert als Einsatzgebiet Echtzeitanwendungen im Bereich der Mensch-Maschine-Interaktion, Objektidentifikation, Gesichtserkennung, Gestenerkennung, Bewegungsverfolgung usw. und stellt demensprechend geeignete Funktionen und Algorithmen in dieser Bibliothek zur Verfügung.

Die in dieser Arbeit verwendete Software benutzt aus openCV die sogenannte Haar-Klassifikation zur Gesichtserkennung. Diese basiert auf der Arbeit von VIOLA und JONES [18]. Zur Erkennung von Gesichtern wird das Bild nach bestimmten Merkmalen durchsucht. Dabei kommen sogenannte „haar-like-features" zum Einsatz, wie sie auch auf Abbildung 2.1 beispielhaft dargestellt werden.

Der Detektor untersucht 24×24-Pixel große Blöcke, wobei bestimmte rechteckige Bereiche innerhalb dieses Blocks aufsummiert und voneinander abgezogen werden. Ein Summand wird hierbei durch den entsprechenden Intensitätswert des untersuchten Graubildpixels gebildet. Das System betrachtet also weder Farbinformationen noch Veränderungen aus Bildsequenzen.

„Haar like features" unterscheiden sich einerseits durch verschiedene Merkmalsklassen, die anhand der Anzahl der verwendeten Rechtecke differenzieren. Abbildung 2.1 (A) und (B) zeigen sogenannte „two rectangle features", bei (C) handelt es sich um ein „three rectangle feature" und (D) zeigt exemplarisch ein „four rectangle feature". Andererseits unterscheiden sich die Merkmale durch

[1] open Computer Vision

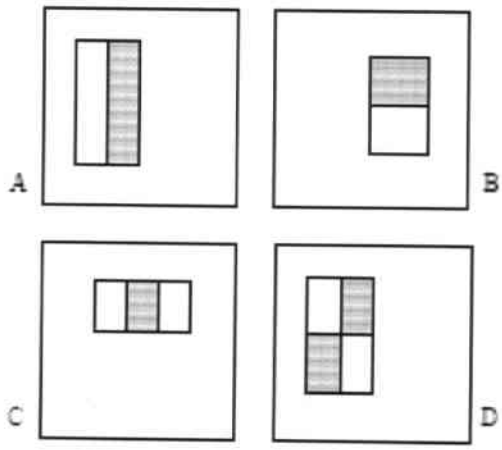

Abbildung 2.1: (A) und (B) zeigen „two rectangle features", bei (C) handelt es sich um ein „three rectangle feature" und (D) zeigt exemplarisch ein „four rectangle feature"

ihre Position, d.h. die Lage der linken oberen Ecke innerhalb des 24×24-Pixel großen Blocks. Die tatsächliche Ausprägung eines Merkmals berechnet sich nun aus der Summe der Intensitätswerte des Graubildes unter dem grauen Rechteck minus der Summe der Intensitätswerte des Graubildes unter dem weißen Rechteck.

Da bei diesem Vorgehen sehr viele Pixelsummen berechnet werden müssen und jedes Pixel mehrfach darin enthalten ist, wird das sogenannte Integralbild verwendet. Dafür wird zu jedem Pixel des Originalbildes ein Wert berechnet, der sich aus der Summe aller Intensitätswerte ergibt, die sich oberhalb und links dieses Pixels befinden. Die folgende Gleichung verdeutlicht diesen Zusammenhang.

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \tag{2.1}$$

Wobei $i(x',y')$ dem Intensitätswert des Originalbilds an Punkt (x',y') und $ii(x,y)$ der Summe an Punkt (x,y) entspricht. Die Autoren präsentieren eine itaritive Vorgehensweise zur Berechnung dieser Werte, so dass sich das Integralbild in einem Durchgang berechnen lässt.

Die letztendliche Pixelsumme eines beliebigen, parallel zum Bild ausgerichteten Rechtecks lässt sich nun ganz einfach berechnen. Wie in Abbildung 2.2 zu sehen ist, ergibt sich der Integralwert von Rechteck D aus

$$I(D) = ii(4) - ii(3) - ii(2) + ii(1)\ , \tag{2.2}$$

wobei die Punkte (1) bis (4) den Eckpunkten des Rechtecks entsprechen.

Mittels eines auf AdaBoost [6] basierenden Lernalgorithmus werden eine relativ geringe Anzahl [2] der möglichen Merkmale selektiert und mehrere Klassifikatoren trainiert, die als Kaskade angeordnet werden. Dies erlaubt es, uns

[2] 10 - 20 von 45396 möglichen Merkmalen

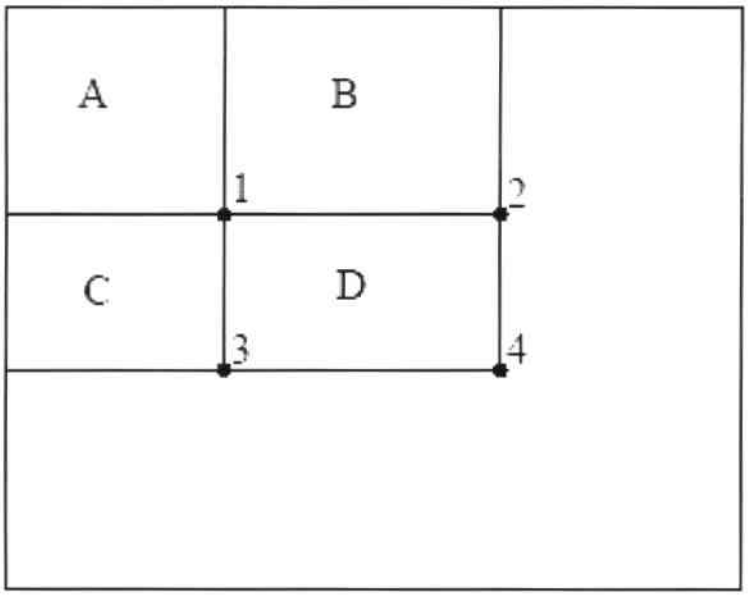

Abbildung 2.2: Der Integralwert des Rechtecks D lässt sich aus den Integralwerten seiner Eckpunkte berechnen

schnell und effizient im gesamten Bild nach Gesichtern zu suchen, da der Haar-Klassifikator leicht skaliert werden kann, um Gesichter in unbekannten Größen zu detektieren.

Dieses Verfahren ist nicht auf Gesichtserkennung in Bildern beschränkt. Haar-Klassifikatoren können auf verschiedenste Arten von Objekten trainiert werden. Die prinzipielle Vorgehensweise ist dann analog.

2.2 Diskrete Kosinustransformation

Die Diskrete Kosinustransformation (DCT) ist eine lineare, orthogonale Transformation, welche ein zeitdiskretes Signal vom Orts- in den Frequenzbereich transformiert und wieder zurück. Seit 1974 ist sie die am weitesten verbreitete Transformation zur Redundanzreduktion von Bildsignalen [14]. Die Gründe hierfür sind vielfältig. Einerseits transformiert die DCT Bilddaten effektiv in eine Form, die gut komprimiert werden kann. Andererseits kann die DCT effizient in Software oder Hardware implementiert werden. Im Gegensatz zur Diskreten Fouriertransformation rechnet man bei der Kosinustransformation nicht mit komplexen, sondern lediglich mit reellen Koeffizienten, was einen weiteren Vorzug darstellt. Für den Prozess der Gesichteridentifikation interessiert uns nur die Transformation vom Bild- in den Frequenzbereich, im Folgenden mit FDCT[3] abgekürzt. Um die Korrelation in horizontaler und vertikaler Bildrichtung zu erfassen, wird die zweidimensionale Variante der FDCT benutzt. Zu diesem Zweck wird das Bild in Blöcke von 8×8 Pixeln zerlegt. Die folgende Gleichung

[3]forward discrete cosine transformation

beschreibt die zweidimensionale FDCT für einen 8×8-Block eines Bildes.

$$F_{x,y} = \frac{2C(x)C(y)}{8} \cdot \sum_{i=0}^{7} \sum_{j=0}^{7} f_{i,j} \cdot \cos\left(\frac{(2i+1)x \cdot \pi}{16}\right) \cdot \cos\left(\frac{(2j+1)y \cdot \pi}{16}\right) \quad (2.3)$$

wobei $f_{i,j}$ der Wert des Punkts (i, j) des Eingabebildes, $F_{x,y}$ die DCT-Koeffizienten an Stelle (x, y) und $C(x)$ und $C(y)$ die Konstanten

$$C(z) = \begin{cases} \frac{1}{\sqrt{2}}, & z = 0 \\ 1, & z \neq 0 \end{cases} \quad (2.4)$$

darstellen. Die FDCT repräsentiert jeden Block eines Bildausschnittes durch gewichtete Summen von 2D-Kosinusfunktionen. Das Muster links oben hat die niedrigste Frequenz und ist nur ein Einheitsblock. Dieser Koeffizient gibt somit den durchschnittlichen Grauwert des gesamten Blocks an. Von links nach rechts nimmt die Frequenz zwischen hell und dunkel in horizontaler Richtung zu. Von oben nach unten nimmt hingegen die Frequenz zwischen hell und dunkel in vertikaler Richtung zu. Folglich nehmen sowohl die horizontalen als auch die vertikalen Frequenzen in diagonaler Richtung gleichzeitig zu. Abbildung 2.3 verdeutlicht diesen Zusammenhang

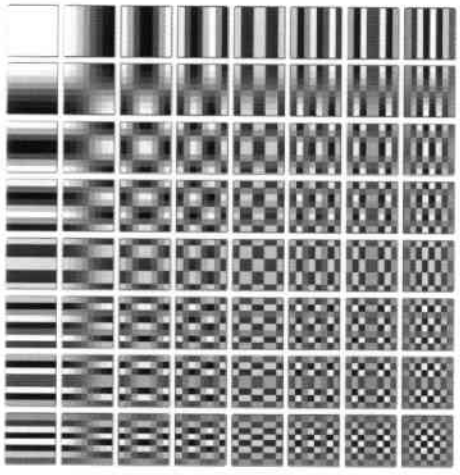

Abbildung 2.3: Basis der Diskreten Cosinustransformation

2.3 Zig-Zag-Scan

Durch die DCT erhält man für jeden 8×8-Pixelblock 64 Koeffizienten. Der sogenannte Zig-Zag-Scan bewirkt eine Umordnung dieser Koeffizienten in einem eindimensionalen Vektor. Der Zig-Zag-Scan beginnt dabei in der linken, oberen Ecke und bewegt sich fortan im Zickzackmuster in Richtung rechter, unterer Ecke. Abbildung 2.4 visualisiert diesen Vorgang. Der Zig-Zag-Scan bewirkt, dass die relevantesten Koeffizienten, also die der niedrigen Frequenzen, am Anfang des Vektors stehen.

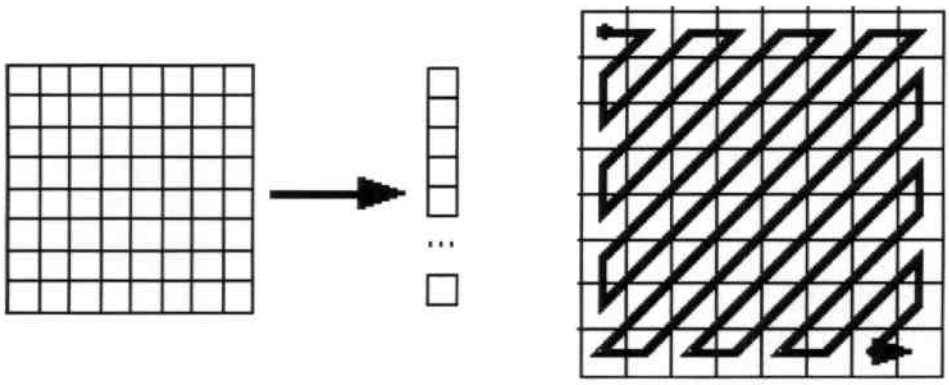

Abbildung 2.4: Der Zig-Zag-Scan ordnet die Koeffizienten in einem eindimensionalen Vektor an. Er beginnt dabei in der linken, oberen Ecke und bewegt sich im Zickzack zur rechten unteren Ecke [13]

2.4 k-Nearest-Neighbour

Die k-Nearest-Neighbour-Lernalgorithmen gehören nach MITCHELL [11] zu den instanzbasierten Lernverfahren. Anstatt eine allgemeine, explizite Beschreibung der Zielfunktion zu lernen, werden einfach die kompletten Trainingsdaten gespeichert. Die Generalisierung über die Daten hinaus erfolgt erst bei der Klassifizierung einer neuen Instanz. Bei jeder Anfrage wird die Beziehung dieser neuen Instanz zu den bisherigen Beispielen untersucht, um einen Zielfunktionswert auszugeben respektive eine Klasse zuzuordnen. Ein großer Vorteil dieser Klasse von Lernalgorithmen liegt darin, auf einfache Weise auch sehr komplexe Zielfunktionen approximieren zu können.

k-Nearest-Neighbour-Algorithmen sind die elementarsten Vertreter von instanzbasierten Lernverfahren und setzen voraus, dass eine Instanz als Punkt in einem Euklidischen Vektorraum repräsentiert werden kann. Der nächste Nachbar[4] wird definiert gemäß der Euklidischen Distanz, d.h.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\ , \tag{2.5}$$

mit x, y als Instanzen im n-dimensionalen Vektorraum. Sofern $k = 1$ gilt, erhält die zu klassifizierende Instanz die Klasse ihres nächsten Nachbarn.

Für höherwertige k werden insgesamt die nächsten k Nachbarn betrachtet. Abbildung 2.5 veranschaulicht dies für $k = 3$. In diesem Fall wird der Einfluss eines Nachbarn gewichtet durch seine Entfernung zur neuen Instanz. Dies bedeutet, dass sehr naheliegende Trainingsbeispiele einen hohen Einfluss auf die letztendliche Klasse haben. Die Klassifizierung erfolgt gemäß der Gleichung

$$c(y) = argmax_{v \in V} \sum_{i=1}^{k} w_i \cdot \delta(v, c(x_i)) \tag{2.6}$$

wobei x_i die nächsten Nachbarn, y die neue Instanz, V die Menge der Klassenlabels, $c(a)$ die Klasse von Instanz a, $\delta(a,b) = 1$ sofern $a = b$, ansonsten 0

[4] nearest neighbour

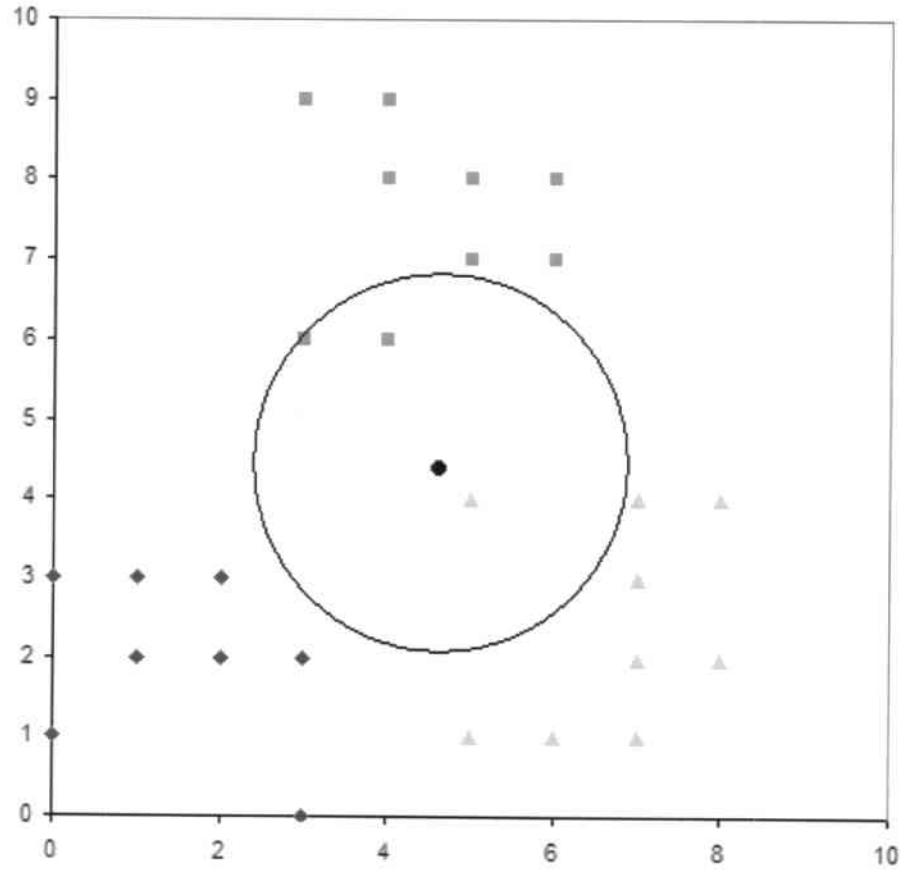

Abbildung 2.5: Der schwarze Punkt repräsentiert die zu klassifizierende Instanz. Innerhalb des Kreises liegen die drei nächsten Nachbarn

und

$$w_i = \frac{1}{d(x_i, y)^2} \tag{2.7}$$

sind.

2.5 Logistische Regression

Unter logistischer Regression[5] versteht man ein Verfahren zur multivariaten Analyse des Zusammenhangs zwischen binär abhängigen Variablen und mindestens einer unabhängigen Variablen [12]. Die Einflüsse auf solche binären Variablen können nicht mit dem Verfahren der linearen Regressionsanalyse untersucht werden, da wesentliche Anwendungsvoraussetzungen nicht gegeben sind. Typische Beispiele sind solche abhängige Variablen, die das Eintreten eines Ereignisses erfassen. Diese Variablen haben nur zwei mögliche, sich gegenseitig

[5] logistic regression

ausschließende Ausprägungen, wie z.B. „Ereignis findet statt" ($Y = 1$) und „Ereignis findet nicht statt" ($Y = 0$). Nun interessiert der Einfluss der jeweiligen unabhängigen Variablen auf diese Eintrittswahrscheinlichkeit. Die logistische Regression löst diese Aufgabe durch eine geeignete Transformation der abhängigen Variablen. Sie geht aus von der Idee der Odds, d.h. dem Verhältnis von $P(Y = 1)$ zur Gegenwahrscheinlichkeit $1 - P(Y = 1)$ bzw. $P(Y = 0)$

$$Odds(Y = 1) = \frac{P(Y = 1)}{1 - P(Y = 1)} = \frac{P(Y = 1)}{P(Y = 0)} \tag{2.8}$$

Die Odds können zwar Werte größer 1 annehmen, doch ist ihr Wertbereich nach unten beschränkt, da er sich asymptotisch 0 annähert. Ein unbeschränkter Wertbereich wird durch die Transformation der Odds in die sogenannten Logits erzielt.

$$Logit(Y = 1) = \ln \frac{P(Y = 1)}{1 - P(Y = 1)} \tag{2.9}$$

Diese können nun Werte zwischen minus und plus unendlich annehmen.

In der logistischen Regression wird dann die Regressionsgleichung

$$Logit(Y = 1|X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n \tag{2.10}$$

geschätzt. Es werden also die Regressionkoeffizienten β_i (auch Logit-Koeffizienten) für jede unabhängige Variable X_i bestimmt, nach denen die Logits für ein gegebens X berechnet werden können. Da diese Regressionskoeffizienten der logistischen Regression nicht einfach zu interpretieren sind, bildet man die sogenannten Effektkoeffizienten mittels Verwendung des Antilogarithmus. Die Regressionsgleichung bezieht sich dadurch wiederum auf die Odds.

$$Odds(Y = 1|X) = \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n) \tag{2.11}$$

Durch eine weitere Transformation lassen sich die Einflüsse der logistischen Regression auch als Einflüsse auf die Eintrittswahrscheinlichkeit $P(Y = 1)$ ausdrücken:

$$P(Y = 1|X) = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)} \tag{2.12}$$

Abbildung 2.6 zeigt eine simple logistische Funktion für eine binäre, abhängige Variable und eine unabhängige Variable mit den Logit-Koeffizienten $b_0 = 0$ und $b_1 = 1$. b_0 hat dabei keinen Einfluss auf die Gestalt der logistische Regressionskurve, sondern lediglich auf die Lage der Kurve in der Horizontalen.

2.6 Klassifikatorkombination

Ein Standardverfahren zur Kombination verschiedener Klassifiaktoren basiert auf der Summenregel von Kittler et al. [9], die verschiedene Klassifikationsergebnisse gemäß der Funktion

$$c = \text{argmax}_{h_j \in H} \sum_{i}^{n} P(h_j|x_i) \tag{2.13}$$

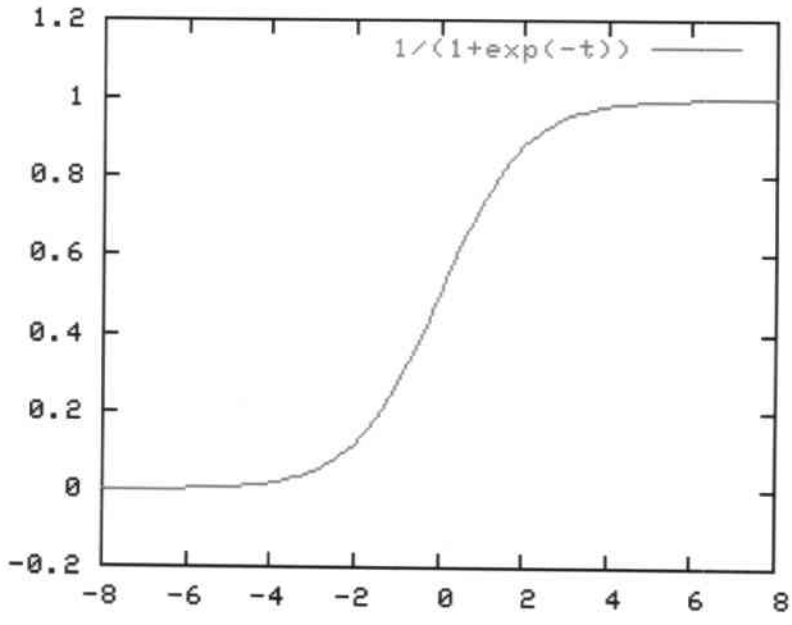

Abbildung 2.6: Einfache logistische Regressionskurve für zwei Variablen mit $b_0 = 0$ und $b_1 = 1$

verbindet, wobei H der gesamten Hypothesenmenge, n der Anzahl an Hypothesen der Benutzersession und $P(h_j|x_i)$ der a posteriori Wahrscheinlichkeit der Einzelbildhypothese h_j unter gegebenem Merkmalsvektor x_i zum Zeitpunkt i entsprechen.

Zur Berechnung der a posteriori Wahrscheinlichkeit einer Einzelbildhypothese lässt sich ein k-Nearest-Neighbour-Schätzer verwenden, welcher $\hat{P}(h_j|x_i)$ schätzt, indem für jede mögliche Hypothese h_j die Anzahl der zugehörigen Trainingsinstanzen unter den k nächsten Nachbarn gezählt und durch k dividiert wird.

Eine andere Möglichkeit besteht darin, ebenfalls unter Verwendung eines k-Nearest-Neighbour-Klassifikators, die sog. Min-Max-Normalisierung [1] zu verwenden, die zunächst den Abstand unter den k nächsten Nachbarn gemäß der Formel

$$\mathrm{nd}(y_i) = 1 - \frac{d(x, y_i) - \min(D)}{\max(D) - \min(D)} \tag{2.14}$$

normalisiert, wobei $d(x, y_i)$ der Distanz von Testinstanz x zur Trainingsinstanz y_i gemäß Gleichung 2.5 entspricht und D analog einer Menge aus Distanzen von Testinstanz zu den k nächsten Nachbarn. Bei diesem normalisierten Abstand handelt es sich jedoch noch nicht um eine Wahrscheinlichkeit, da

$$\mathrm{sum} = \sum_{i=1}^{k} \mathrm{nd}(y_i) \tag{2.15}$$

nicht 1 ergibt. Stattdessen wird diese Bedingung erfüllt, sofern man die a pos-

teriori Wahrscheinlichkeit wie in der Gleichung

$$\hat{P}(h_j|x) = \max_{i=1}^{k} \frac{\mathrm{nd}(y_i) \cdot \gamma(y_i, h_j)}{\mathrm{sum}} \tag{2.16}$$

approximiert, wobei $\gamma(y_i, h_j) = 1$ enspricht, sofern das Label von y_i mit der Hypothese h_j übereinstimmt. Anderenfalls entspricht $\gamma(y_i, h_j) = 0$.

2.7 k-means Clustering

Unter dem Begriff des Clusterings[6] versteht man ein multivariates Analyseverfahren zur Ermittlung von Clustern von Objekten, deren Eigenschaften oder Ausprägungen bestimmte Ähnlichkeiten vorweisen [15]. Clustering kommt bspw. dann zum Einsatz, um eine vorhandene Datenmenge zu kategorisieren, um von ihr zu abstrahieren oder um ihre Komplexität zu reduzieren.

Der k-means Algorithmus stellt hierbei ein Clusteringverfahren dar, dass unüberwacht k Cluster aus den gegebenen Datenpunkten bildet. Der Algorithmus setzt dabei eine Distanzfunktion zur Bestimmung des Abstands zweier Elemente und eine Methode zur Berechnung des Mittelpunkts, auch Zentroid genannt, voraus. Der in dieser Arbeit eingesetzte k-means Algorithmus verwendet als Distanzfunktion die Euklidische Distanz, wie sie auch schon in Gleichung 2.5 definiert wurde. Zur Bestimmung des Zentroiden kommt die Gleichung

$$\hat{x} = (x_1, \ldots, x_n)^T = (\frac{1}{m}\sum_{i=1}^{m} y_{i_1}, \ldots, \frac{1}{m}\sum_{i=1}^{m} y_{i_n})^T = \frac{1}{m}\sum_{i=1}^{m} \hat{y}_i \tag{2.17}$$

zum Einsatz, wobei $\hat{x}$ einem Zentroid im n-dimensionalen Vektorraum entspricht, dem die m Datenpunkte $\hat{y}_1$ bis $\hat{y}_m$ zugeordnet wurden.

Der Algorithmus läuft nach folgendem Schema ab:

1. Initialisierung: Zufällige Generierung von k Zentroiden oder zufällige Auswahl von k Punkten des Datenbestands.
2. Zuordung: Jeder Punkt der Datenmenge wird gemäß Gleichung 2.5 dem nächsten Zentroiden zugeordnet.
3. Neuberechnung: Für jedes Cluster werden die Zentroiden gemäß Gleichung 2.17 neu berechnet.
4. Überprüfung: Falls sich die Zentroiden verschoben haben, wird der Algorithmus bei Schritt 2 fortgesetzte. Ansonsten bricht das Verfahren ab.

Obwohl gezeigt werden kann, dass der Algorithmus immer terminiert, hängen die resultierenden Cluster stark von den anfänglichen Zentroiden ab.

[6] cluster, zu dt. Ballung, Anhäufung

Kapitel 3

Klassifikation und Konfidenzberechnung auf Bildfolgen

Dieses Kapitel erklärt den Ablauf und die einzelnen Schritte, die durchgeführt werden müssen bevor ein Gesicht in einer Bildsequenz identifiziert werden kann. Dazu verschafft der erste Teil zunächst einen allgemeinen Überblick über das komplette System und die Systemarchitektur. Nachfolgend wird näher auf einzelne Komponenten eingegangen. Teil zwei beschreibt daher das Personentracking, Teil drei die Vorgehensweise der Bildvorverarbeitung und Bildung eines Merkmalsvektors. Abschnitt vier und fünf befassen sich mit der Bestimmung einer Hypothese auf einem einzelnen Bild und der Berechnung einer zugehörigen Konfidenz als Maß der Klassifikationsgüte. Abschnitt sechs erläutert die durchgeführte Segmentierung, bevor in den folgenden beiden Teilen die Gesichteridentifikation auf Bildfolgen projiziert wird. Abschließend wird der Aspekt des Lernens und der Eingliederung neuer Personen beleuchtet.

3.1 Überblick

Auf dem Weg zur Erstellung einer Annahme über die sich vor dem Roboter befindliche Person werden verschiedenste Prozesse durchlaufen, die auf vielfältige Weise miteinander interagieren und voneinander abhängen. Abbildung 3.1 verschafft einen schematischen Überblick über das Gesamtsystem und das Zusammenspiel der verschiedenen Komponenten.

Am Anfang des Prozesses steht die Roboterkamera, die in regelmäßigen Abständen Aufnahmen macht. Sofern das Gesicht einer Person auf dem Bild zu erkennen ist, erfolgen anschließend diverse bildvorverarbeitende Schritte, die als Ergebnis einen Merkmalsvektor liefern, der das Gesicht des Individuums repräsentiert. Parallel dazu erfolgt das Personentracking und die Segmentierung in Benutzersessions. Aufgrund des extrahierten Merkmalsvektors und den bereits bestehenden Daten erfolgt eine Klassifikation, die zur Bestimmung einer Einzelbildhypothese führt. Zu dieser Hypothese wird eine Konfidenz berechnet, d.h. eine Wahrscheinlichkeit inwieweit man dieser Identifikation vertrauen kann.

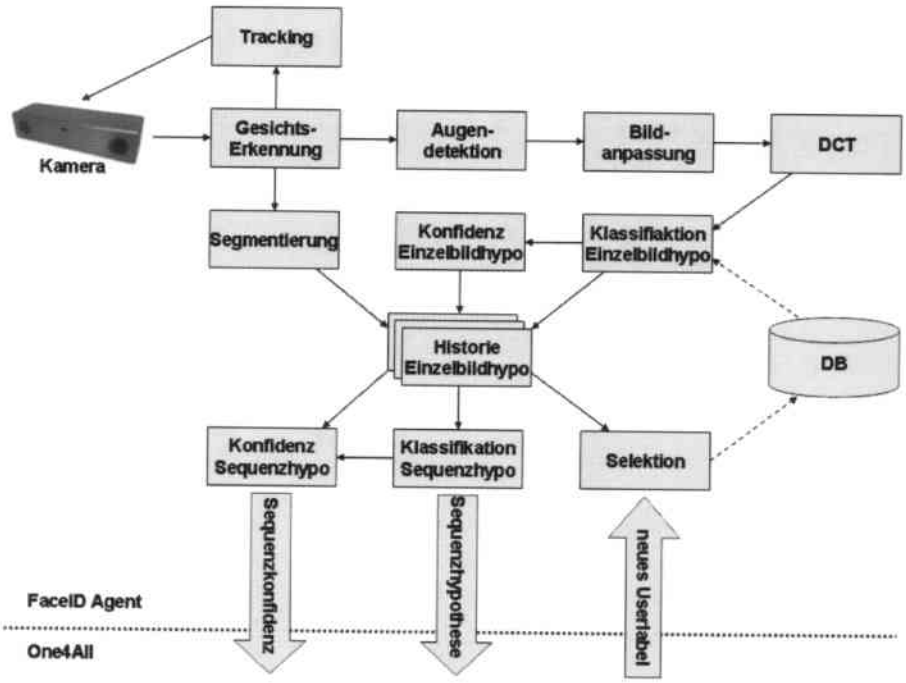

Abbildung 3.1: Überblick über das System und die Verbindung zwischen den einzelnen Komponenten

Dies ermöglicht es, bestimmte Hypothesen mit geringer Konfidenz fallen zu lassen und umgekehrt vertrauenswürdige Annahmen weiter zu verfolgen. Zu jeder Benutzersession werden die Einzelbildhypothesen und deren zugehörige Konfidenzen gesammelt. Diese bilden die Historie auf deren Basis eine weitere Klassifikation zur Gesichteridentifikation auf Bildsequenzen erfolgt. Anschließend wird auch zu dieser Sequenzhypothese eine Konfidenz berechnet, die analog das Vertrauen in die getroffene Annahme widerspiegelt. Abhängig von diesem Konfidenzwert sind nun andere Systeme auf dem Roboter, wie bspw. ein Dialogsystem [8], in der Lage, die Sequenzhypothese zu verwerfen, zu verifizieren oder gar als gegeben zu betrachten. Die Kommunikation zu diesen externen Komponenten erfolgt über das ONE4ALL Kommunikationsprotokoll, welches am ISL der Universität Karlsruhe entwickelt wurde und eine FIPA/ACL[1] Nachrichtenstruktur verwendet.

Die zuvor aufgeführten Schritte wiederholen sich für jeden neuen Frame, so dass fortlaufend neue Einzelbildhypothesen erstellt werden, die zu einer Aktualisierung der Sequenzhypothese führen. Nach Ende der Benutzersession und Erhalt eines entsprechenden Befehls über ONE4ALL können neu extrahierte, repräsentative Merkmalsvektoren zu den bereits bestehenden Daten hinzugefügt werden, um die vorhandene Datenbasis zu erweitern.

3.2 Tracking

Die Kamera des Roboters ruht auf einer sog. PTU[2], einem Bauteil, das Drehungen in horizontaler und vertikaler Richtung zulässt. Somit ist es möglich, die Bewegungen einer Person zu verfolgen. Dadurch wird einerseits die Aufmerksamkeit der Person erregt [17] und andererseits die Dauer des Benutzers im Blickfeld des Roboters verlängert.

Das Tracking erfolgt durch Approximation des Winkels zwischen Benutzer und Bildmittelpunkt. Dazu wird die Pixeldifferenz zwischen Gesichtsmittelpunkt und Bildmitte in horizontaler Richtung berechnet. Mit Hilfe trigonometrischer Funktionen lässt sich diese Differenz in einen entsprechenden Winkel umrechnen, um den die PTU gedreht wird. Liegt der Winkel unterhalb eines zuvor festgelegten Grenzwertes, erfolgt keine Drehung, um ein ständiges Anpassen zu vermeiden und die Bildqualität durch verschwommene Aufnahmen nicht zu verschlechtern.

3.3 Bildvorverarbeitung und Merkmalsextraktion

Um eine Identifikation eines Gesichtes auf Bildern überhaupt zu ermöglichen, müssen zuvor diverse bildvorverarbeitende Arbeitsschritte ausgeführt werden. Herzstück der Gesichteridentifikation ist der sogenannte Face Recognizer [3] von Hazim K. Ekenel der Universität Karlsruhe (TH). Diese Software wurde in C++ geschrieben und verwendet in vielfältiger Weise die Bibliotheken von openCV. Zunächst wird das Bild (Abbildung 3.2 (a)) mittels der oben beschriebenen

[1] http://www.fipa.org
[2] Pan-Tilt-Unit
[3] Gesichtserkenner

Haarklassifikatoren auf Gesichter untersucht. Der Gesichtsbereich ist in Abbildung 3.2 (b) durch ein rotes Rechteck hervorgehoben. Sofern ein Gesicht erkannt wurde, fährt man mit der Augendetektion im Gesichtsfeld fort. Dies erfolgt ebenfalls mittels Haarklassifikatoren, die trainiert wurden, rechte bzw. linke Augen zu erkennen. Augen- und Gesichtsbereich sind in Abbildung 3.2 (c) grün markiert. Die Detektion des Augenpaars ist erforderlich, da anhand des Augenabstands

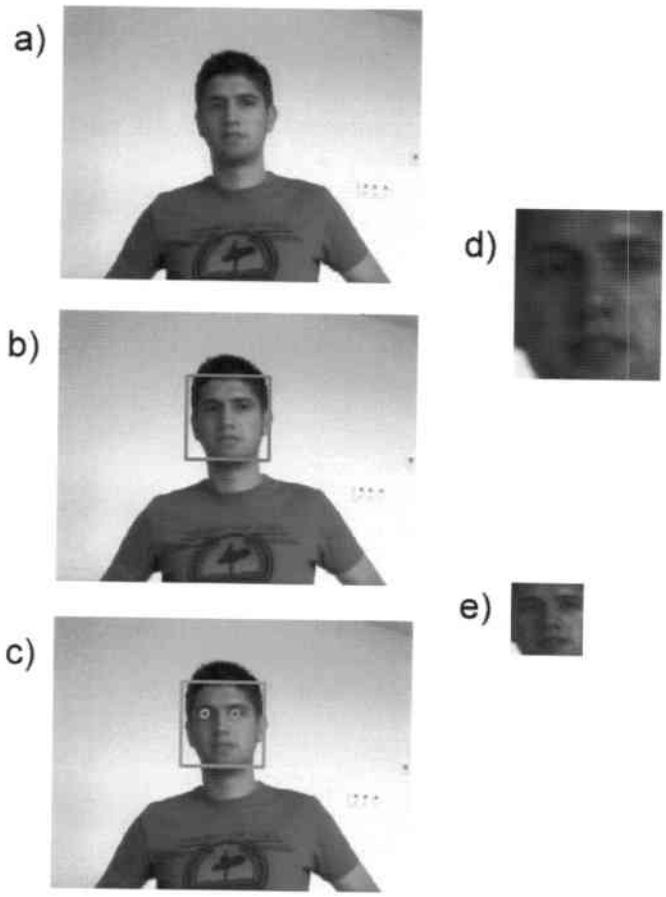

Abbildung 3.2: Vorverarbeitung eine Bildes durch den Face Recognizer: a) Ursprungsbild, b) Detektion des Gesichts, c) Augendetektion, d) Skalierung, Grauwertkonvertierung und Extraktion des Gesichts und e) Skalierung auf einen quadratischen Gesichtsbereich

und der Augenkoordinaten das Bild rotiert und auf eine fixe Größe skaliert wird. Daraus resultiert ein neues 130×150-Pixel großes Bild, das lediglich das Gesicht zeigt. Die Augenzentren liegen nun symmetrisch zur senkrechten Mittelachse und haben einen normierten Abstand von 70 Pixeln zueinander. Zudem sind die Augenmittelpunkte 45 Pixel vom oberen Bildrand entfernt. Durch diese „Normalisierung" des Gesichtsbereich erzeugt man eine gemeinsame Basis auf der Gesichter unterschiedlicher Personen sinnvoll miteinander verglichen werden können, ohne die jeweiligen Differenzen, die durch verschiedene Neigungs- und Blickwinkel des Gesichts entstehen, betrachten zu müssen. Desweiteren werden

die enthaltenen Farbinformationen verworfen, indem eine Konvertierung in ein Graubild durchgeführt wird. Das Resultat zeigt Abbildung 3.2 (d). Letztendlich erfolgt eine weitere Skalierung auf ein 64×64-Pixel großes, quadratisches Gesichtsbild wie in Abbildung 3.2 (e), bevor die Vorverarbeitung durch den Face Recognizer abgeschlossen ist.

Das resultierende Bild wird in 64 8×8-Pixel große Blöcke unterteilt. Auf jedem einzelnen wird wie oben beschrieben die diskrete Kosinustransformation ausgeführt. Pro Block werden somit 64 Koeffizienen erzeugt, die mittels Zig-Zag-Scan in einen eindimensionalen Vektor geschrieben werden. Wie bereits erwähnt befinden sich die Koeffizienten der niedrigen Frequenzen nun am Anfang des Vektors. Diese enthalten die zur Beschreibung des Pixelblocks wesentlichen Informationen. Unter Auslassung des ersten Koeffizienten (dieser beschreibt nur den durchschnittlichen Grauwert des Blocks) beschränkte man sich in der vorliegenden Arbeit auf die fünf folgenden Koeffizienten. Dadurch reduziert sich die zu betrachtende Gesamtzahl der Koeffizienten pro Gesicht von 4096[4] auf 320[5].

Die Konkatenation dieser verbleibenden Koeffzienten über das ganze 64 × 64-Pixel große Gesichtsfeld wird auch „feature fusion" genannt [3], womit ein Merkmalsvektor gebildet wird, der das komplette Gesicht repräsentiert. Dieser stellt die Grundlage dar, anhand derer im weiteren Verlauf Gesichter klassifiziert werden.

3.4 Einzelbildhypothese

Zur Erstellung einer Einzelbildhypothese, d.h. die Erstellung einer Hypothese über die Person auf einem einzelnen Bild anhand ihres Merkmalsverktors, wird in dieser Arbeit der k-Nearest-Neighbour-Klassifikator verwendet. Da es sich bei dem Merkmalsvektor um einen reellwertige Zahlenvektor handelt, kann eine Instanz als Punkt im 320-dimensionalen Euklidischen Vektorraum verstanden werden. Die Distanz einer neu zu klassifizierenden Instanz zu bereits trainierten Instanzen berechnet sich nun wie in Gleichung 2.5 gegeben. Wählt man $k = 1$ entspricht die Hypothese der Klasse derjenigen Trainingsinstanz, die der neuen Instanz am nächsten liegt. Desweiteren wurden in dieser Arbeit auch Klassifikationen für $k = 3$ und $k = 5$ durchgeführt. Dabei wird die Einzelbildhypothese gemäß den Gleichungen 2.6 und 2.7 bestimmt.

3.5 Konfidenz der Einzelbildhypothese

Neben der eigentlichen Hypothese, welche Person sich auf einem Bild befindet, ist es auch von großem Interesse, die Konfidenz in diese Hypothese zu ermitteln. Konfidenz bedeutet Zuversicht und Vertrauen und drückt in diesem Zusammenhang nichts anderes aus als eine Art Wahrscheinlichkeit, mit der eine Hypothese der Realität entspricht. Mit Hilfe der logistischen Regression wird in dieser Arbeit die Konfidenz der Einzelbildhypothese geschätzt. Die binäre, abhängigen Variable Y erfasst das Ereignis „Hypothese richtig" ($Y = 1$) bzw. „Hypothese falsch" ($Y = 0$). Die Wahl der unabhängigen Variablen X_i gestaltet sich ungleich schwieriger, da sich eine große Anzahl an möglichen Merkmalen

[4] $64 \cdot 64 = 4096$
[5] $5 \cdot 64 = 320$

als Variablen anbieten. Nach Einsicht der Daten, die in Kapitel 4 näher erläutert werden, haben sich letztendlich vier Merkmale herauskristallisiert, die eine sinnvolle Modellierung mittels logistischer Regression zulassen. Diese sind im Einzelnen:

- X_1: Distanzabweichung
- X_2: Grauwertmittel
- X_3: Erstklassifikation
- X_4: Nachbardistanz

Bei der Variablen „Distanzabweichung" handelt es sich um die absolute Differenz zwischen aktuellem Abstand der Person auf dem Bild zur Roboterkamera und der durchschnittlichen Distanz. Die aktuelle Distanz wird hierbei anhand der Pixelbreite des Gesichtsfeldes approximiert, da sich diese zur tatsächlichen Distanz antiproportional verhält.

Die Variable „Grauwertmittel" beschreibt den durchschnittlichen Grauwert aller Pixel auf dem finalen 64×64-Pixel großen Gesichtsbild. Mit dieser Variablen sollen die vorherrschenden Lichtverhältnisse erfasst werden. Sind diese ausreichend hell, so treten die Kontraste zwischen verschiedenen Elementen eines Gesichts (wie bspw. Stirn und Augenbrauen) in den Vordergrund, wodurch personenindividuelle Differenzen stark betont werden.

Bei der Variablen „Erstklassifikation" handelt es sich um eine binäre Variable. Kann der Gesichterkenner nach einer Sequenz von mindestens einem Frame, in der keine Augen oder kein Gesicht detektiert werden konnte, zum ersten Mal wieder eine Identifikation durchführen, so hat diese Variable den Wert 1.

Desweitern kommt die Variable „Nachbardistanz" zum Tragen, die den Abstand zum nächsten Nachbarn der k-Nearest-Neighbour-Klassifikation widerspiegelt.

Der Vollständigkeit halber sei erwähnt, dass das Merkmal der Distanz nicht direkt als Vorhersagevariable verwendet werden konnte, da der Verlauf dieser Variablen nicht monoton steigt, sich also nicht zur Modellierung eines logistischen Zusammenhangs eignet. Der Grund hierfür liegt in der Funktionsweise der Gesichtsidentifiaktion, die eine Detektion der Augen voraussetzt. Falls sich die Person zu nah am oder zu weit entfernt vom Roboter befindet, funktioniert diese Erkennung leider nur unzureichend aufgrund verschlechterter Bildqualität. Andere Merkmale wie bspw. der Winkel, der sich zwischen Benutzer, Roboterkamera und Bildmittelachse bildet, wurden nicht zur Berechnung der Konfidenz hinzugezogen, da kein nennenswerter statistischer Zusammenhang zwischen diesen und dem Klassifikationsergebnis festgestellt werden konnte.

Nach Festlegung auf diese vier unabhängigen Variabeln wurde mit Hilfe der Software WEKA 3 [5] eine logistische Regressionsfunktion trainiert und somit die zugehörigen Logit-Koeffizienten β_0, β_1, β_2, β_3 und β_4 bestimmt. So kann nun für jede neue Einzelbildhypothese auch eine zugehörgie Konfidenz gemäß Gleichung 2.12 bestimmt werden.

3.6 Segmentierung

Die Segmentierung beschäftigt sich damit, Beginn und Ende einer Benutzersession zu definieren. Sequenzhypothesen basieren auf dem Klassifikationsergebnis

einer Sequenz von Einzelbildhypothesen. Das bedeutet, dass die Sequenzhypothese zum Zeitpunkt t_i stark von den vorherigen Hypothesen der Zeitpunkte t_0 bis t_{i-1} abhängt. Zur Vermeidung von Hypothesenbildung über verschiedene Benutzer hinweg, ist es eminent von Bedeutung, den Beginn einer Sequenz, also denn Beginn der Benutzersession zu erkennen. Abbildung 3.3 zeigt einen Zustandsautomaten der das Sessionmodell repräsentiert.

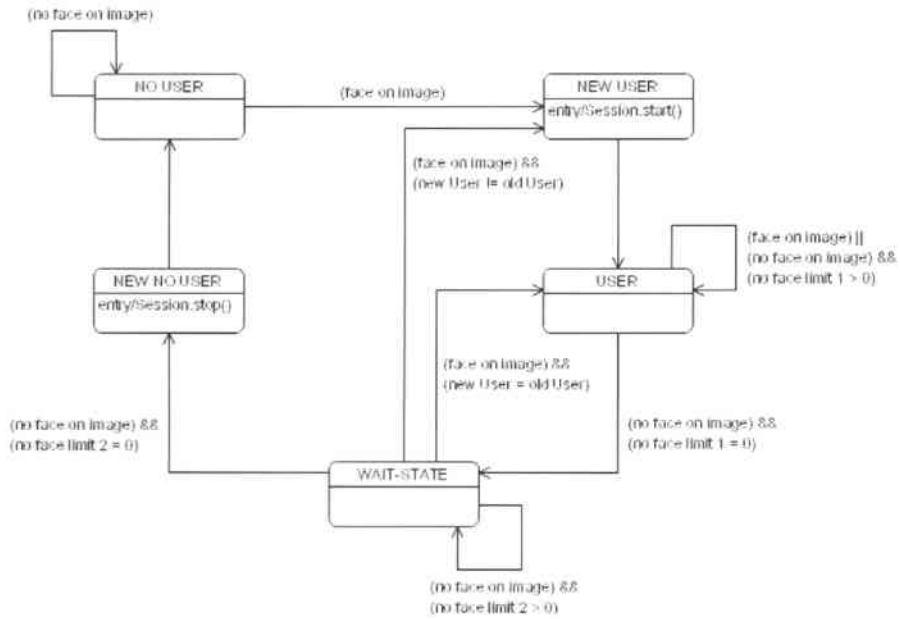

Abbildung 3.3: Der momentane Zustand des Zustandsautomat des Sessionmodells ist in erster Linie davon abhängig, ob ein Gesicht detektiert werden kann.

In der vorliegenden Arbeit wird eine neue Session initiiert, sobald der Gesichtsdetektor das erste Mal ein Gesicht auf einem Bild erkennt. Bei jedem neuen Frame wird das Modell aktualisiert. Sobald drei Sekunden lang kein Gesicht mehr detektiert werden konnte, wechselt das Modell in einen Wartezustand. Wird in den folgenden sieben Sekunden wiederum kein Gesicht erfasst, so gilt die Benutzersession als beendet und der Automat kehrt in den ursprünglichen Zustand zurück. Andernfalls wird überprüft, ob die neue Sequenzhypothese mit der letzten Sequenzhypothese übereinstimmt. Ist dies der Fall, so wird die alte Benutzersession wieder aufgenommen. Dementsprechend wird eine neue Session gestartet, sofern keine Übereinstimmung vorliegt.

3.7 Sequenzhypothese

Unter dem Begriff der Sequenzhypothese versteht man analog zur Einzelbildhypothese eine Annahme über die sich vor dem Roboter befindliche Person. Im Gegensatz dazu wird bei deren Bildung nicht nur ein einzelnes Bild, sondern die vielen Einzelbildhypothesen der aktuellen Benutzersession betrachtet. Es besteht allerdings die Möglichkeit sich auf eine vordefinierte Fensterbreite

zu beschränken, d.h. in die Bestimmung der Sequenzhypothese fließen nur die Einzelbildhypothesen mit ein, die innerhalb der letzten zehn, 20 oder 30 Frames erstellt wurden. Schließlich wird auch innerhalb dieses Fensters weiter gefiltert. Ein vordefinierter Prozentsatz p gibt an, dass nur die besten p Prozent weiter betrachtet werden. Als Qualtiätsmaß dient hierbei die oben erwähnte Konfidenz der Einzelbildhypothese.

Die Konfidenz spielt aber noch eine weitere entscheidende Rolle, da die Konfidenz identischer Einzelbildhypothesen gemäß der Gleichung

$$\text{sum}(h_j) = \sum_{h_i \in H} \text{konf}(h_i) \cdot \delta(h_i, hj) \tag{3.1}$$

aufsummiert wird. h_i ist hierbei eine Einzelbildhypothese der verbleibenden Hypothesenmenge H, $\text{konf}(h_i)$ entspricht der zugehörigen Konfidenz und $\delta(h_i, hj) = 1$ sofern h_i und h_j gleich sind, ansonsten 0.

Die letztendliche Sequenzhypothese wird dann gemäß der Gleichung

$$c = \text{argmax}_{h_i \in H} \, \text{sum}(h_i) \tag{3.2}$$

bestimmt mit $sum(h_i)$ als der zuvor berechneten Konfidenzsumme der Hypothese h_i.

Die Sequenzhypothese wird während des laufenden Betriebs mit jedem neuen Frame und neuer Einzelbildhypothese innerhalb einer Benutzersession aktualisiert und erneut bestimmt. Sie kann somit nicht nur zu einem bestimmten Zeitpunkt der Sitzung erzeugt werden, sondern stattdessen an jeder beliebigen Stelle der Benutzersession.

3.8 Konfidenz der Sequenzhypothese

Desweiteren ist auch für die Sequenzhypothese die Bestimmung einer Konfidenz als Ausdruck ihrer Zuverlässigkeit von großem Interesse. Das hier beschriebene System ist nur eine Komponente auf einem Roboter und interagiert dort mit anderen Komponenten, die eine solche Sequenzhypothese benötigen. Ein Qualitätsmaß in Form einer Konfidenz ist also auch hier von großem Nutzen.

Auch bei der Sequenzhypothese wird die zugehörige Konfidenz mit Hilfe der logistischen Regression bestimmt. Erneut erfasst die binäre, abhängige Variable Y das Ereignis „Hypothese richtig“ ($Y = 1$) bzw. „Hypothese falsch“ ($Y = 0$). Bei der Suche nach geeigneten Merkmalen als unabhängige Variablen wurden besonders diejenigen betrachtet, die Sequenzeigenschaften modellieren. Schließlich wurden die Merkmale „Übereinstimmung“ und „Stabilität“ zur Konfidenzberechnung ausgewählt. Die Variable „Übereinstimmung“ erfasst dabei die Anzahl der Einzelbildhypothesen, die mit der akutellen Sequenzhypothese äquivalent sind, geteilt durch die Anzahl der Sessionframes. „Stabilität“ hingegen erfasst Anzahl der Einzelbildhypothesenwechsel pro Sessionframe. Beide Variablen können lediglich Werte zwischen 0 und 1 annehmen; sie gelten daher als normalisiert. Nicht betrachtet wurden Merkmale wie bspw. die reine Anzahl der Idenfikationen pro Sessionframes, oder die Anzahl der Identifikation geteilt durch die Fensterbreite, da diese sich nicht als passend erwiesen haben, den logistischen Zusammenhang zum korrekten Klassifikationsergebnis zu modellieren.

Wie zuvor kommt bei der Bestimmung der Logit-Koeffizienten β_0, β_1 und β_2 die Software WEKA 3 zum Einsatz, um eine passende logistische Regressionsfunktion zu trainieren. Auch hier erfolgt die Berechnung der Konfidenz nach Gleichung 2.12.

3.9 Lernen und Erweitern der Datenbasis

Das hier vorgestellte System ist nicht nur in der Lage von „Hand" trainierte Personen zu identifizieren, stattdessen können iterativ neue Personen online erlernt werden. Prinzipiell werden während einer Benutzersession alle extrahierten Merkmalsvektoren der Person gespeichert. Um diese in den Datenbestand eingliedern zu können, ist auch ein Personenlabel vonnöten, das die Klasse der neuen Trainingsdaten repräsentiert. Dieses neue, unbekannte Personenlabel kann bspw. vom Dialogsystem [8] bereitgestellt werden. Zusammen mit der Aufforderung, die neue Person in die Datenbasis aufzunehmen, wird der Lernprozess des Gesichteridentifikationssystem initiiert. Dazu werden zunächst die gespeicherten Merkmalsvektoren mit Hilfe des k-means Algorithmus geclustert. Die Anzahl k der Cluster hängt dabei von der Anzahl m der Merkmalsvektoren gemäß der Relation

$$k = \lfloor \sqrt{m} \rfloor \tag{3.3}$$

ab. Die durch den Algorithmus bestimmten Zentroiden werden dann anschließend zusammen mit dem Personenlabel in die vorhandene Datenbank eingegliedert. Durch diese Technik verhindert man ein zu rasantes Anwachsen der Trainingsdaten, da nur eine verhältnismäßige geringe Anzahl an Merkmalsvektoren hingezügt wird.

Kapitel 4

Experimente

In diesem Kapitel wird zunächst der Aufbau der durchgeführten Experimente beschrieben, bevor im folgenden Abschnitt die dadurch erzeugten Daten erläutert werden. Im Anschluss daran erfolgt eine Beschreibung des Trainings und der durchgeführten Tests zur Bestimmung der benötigten Parameter für die Konfidenzberechnung. Abschließend werden im vierten Abschnitt die Evaluationsergebnisse vorgestellt und mit Standardverfahren verglichen.

4.1 Wizard of Oz-Aufnahmen

Zur Erzeugung der Daten wurde ein Kennenlern-Dialog-Szenario geschaffen. Dieses Szenario sieht vor, dass der Proband den Robotor mit dem zuvor vorgestellten Gesichtsidentifikationssystem passiert. Daraufhin versucht die Maschine die Aufmerksamkeit des Probanden zu erregen, indem einerseits die Roboterkamera den Benutzer trackt und andererseits der Robotor über ein ebenfalls vorhandenes Dialogsystem den Benutzer anspricht. Daraufhin wird der Proband in ein kurzes Gespräch verwickelt, bei dem der Name des Probanden erfragt wird. Nachdem diese Information erfolgreich ausgetauscht wurde, verabschiedet sich der Roboter vom Probanden woraufhin dieser seinen Weg fortsetzt.

Der Dialog wurde nach der Art eines Wizard-of-Oz-Experiments durchgeführt. Nach [10] handelt es sich dabei um ein Experiment, bei dem der Proband annimmt, mit einem im Sinne der Künstlichen Intelligenz autonomen System zu kommunizieren. In Wirklichkeit aber erzeugt ein anderer Mensch im Verborgenen die Reaktionen des Systems. Im konkreten Fall bedeutet dies, dass die Äußerungen des Probanden nicht durch das Dialogsystem verarbeitet wurden. Stattdessen wurden die Antworten und Äußerungen des Roboters von einem Menschen ausgewählt und über die TTS[1]-Komponente des Dialogsystems ausgegeben.

4.2 Datensatz

Die durch das oben beschriebene Experiment erzeugten Daten bestehen aus einer Sammlung von Einzelbildern. Sobald der Roboter den Probanden im Blick-

[1] text to speech

feld detektiert, wird die Aufnahme mit zwei Frames pro Sekunden gestartet. Da es sich bei der verfügbaren Kamera, um ein Stereogerät handelte, wurden somit vier Bilder pro Sekunde gespeichert. Das Bildformat entspricht einer Bitmap in der Auflösung 640×480 Pixel mit 24-bit Farbtiefe. Der Dateiname eines Bildes besteht aus dem Zeitstempel des Aufnahmebeginns, gefolgt von einer fortlaufenden Sequenznummer und einem Indikator mit welcher Kamera das Bild aufgenommen wurde. Somit lassen sich alle Bilder einer Benutzersession zuordnen und in einen zeitlichen Zusammenhang bringen. Zusätzlich wird nach dem Ende der Session eine Datei angelegt, die Metainformationen, wie das Personenlabel sowie Zeitstempel von Anfang und Ende der Aufnahme enthalten.

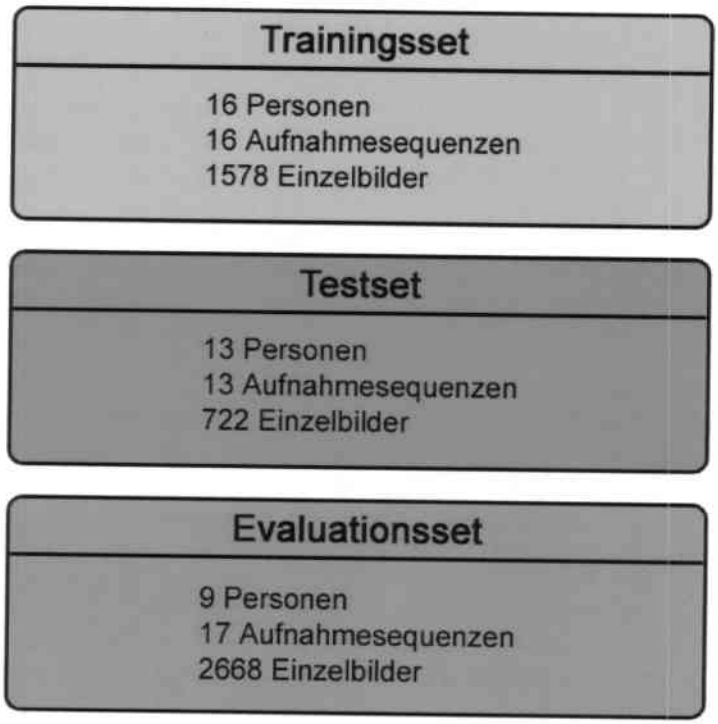

Abbildung 4.1: Der gesamte Datenbestande wird in drei Datensets aufgeteilt: Trainingsset, Testset und Evaluationsset

Auf diese Art und Weise wurden von insgesamt 16 Personen Aufnahmen gemacht. Das Wizard-of-Oz-Experiment wurde dabei pro Person mehrfach durchgeführt, so dass pro Benutzer zwei oder drei Aufnahmesequenzen vorliegen. Zwischen den individuellen Aufnahmen liegt ein zeitlicher Abstand von etwa einem bis vier Monaten. Eine Aufnahmesequenz besteht aus 20 bis 150 Einzelbildern und einer Datei mit Metainformationen. Der gesamte Datenbestand wurde wie in Abbildung 4.1 zu sehen in drei Sets unterteilt.

Das Trainingsset enthält dabei von jeder Person genau eine Aufnahmesequenz. Weitere Aufnahmesequenzen des Probanden wurden dann entweder für das Test- oder das Evaluationsset verwendet.

4.3 Durchführung der Experimente

4.3.1 Training

Wie der Name schon andeutet, wurden die Daten aus dem Trainingsset dazu verwendet, den k-Nearest-Neighbour-Klassifikator zu trainieren. Dazu wurde jedes Bild, wie in Kapitel 3 beschrieben, nach Gesichtern und Augen durchsucht und ggf. ein Merkmalsvektor des Gesichts extrahiert. Dieser wurde dann zusammen mit dem Personenlabel als Trainingsinstanz in der Datenbasis hinterlegt. Von den vorhandenen 1578 Trainingsbildern konnten auf insgesamt 711 Frames Merkmalsvektoren extrahiert werden.

4.3.2 Bestimmung der Logit-Koeffizienten zur Berechnung der Einzelbildhypothesenkonfidenz

Mit Hilfe des Testsets wurde zunächst das Klassifikationsergebnis der Einzelbildhypothese überprüft. Dazu wurden diverse Testläufe vorgenommen mit unterschiedlicher Wahl des Parameters k des k-Nearest-Neighbour-Klassifikators. Die folgenden Tabelle 4.1 zeigt die Resultate die auf dem Testset erzielt werden konnten:

k	Hypothesen	korrekt klassifiziert	Erfolgsquote
1	394	313	79,44%
3	394	303	76,90%
5	394	302	76,40%

Tabelle 4.1: Erfolgsquote der Einzelbildhypothese auf dem Testset für unterschiedliche k

Für $k = 1$ wurden von den 394 Bildern, auf denen sowohl Gesicht als auch Augenpaar detektiert werden konnten, 313 korrekte Einzelbildhypothesen aufgestellt. Dies ist geringfügig besser als bei den anderen Konfigurationen dieses Klassifikationsverfahrens. Aus diesem Grund wurde im weiteren Verlauf stets der einfache Nearest-Neighbour-Algorithmus zur Bestimmung der Einzelbildhypothesen verwendet. Unter anderem wurde das hiermit erzielte Klassifikationsergebnis benutzt, um geeignete Merkmale zur Approximation der Logit-Koeffizienten auszuwählen und zu bestimmen. Abbildung 4.2 zeigt in vier Histogrammen den quantitativen Zusammenhang der selektierten Merkmale und den Hypothesen. Die blauen Bereiche kennzeichnen fehlerhafte Hypothesen des jeweiligen Intervalls, wohingegen rot korrekte Annahmen markiert. Über den Balken ist die jeweilige Gesamtzahl an Einzelbildhypothesen pro Intervall aufgetragen. Histogramm 4.2 a) bezieht sich auf die Variable „Distanzabweichung", Darstellung 4.2 b) nimmt Bezug auf das Merkmal „Grauwertmittel". Abbildung 4.2 c) visualisiert den Zusammenhang zwischen der binären Variable „Erstklassifikation" und dem Klassifikationsergebnis. Dabei ist die Variable mit 0 für „keine Erstklassifikation" und 1 für „Erstklassifikation" kodiert. Das Histogramm 4.2 d) zeigt letztendlich das Verhältnis von Merkmal „Nachbardistanz" zum Klassifikationsergebnis.

Anhand dieser Ergebnisse erfolgte daraufhin mit Hilfe der bereits erwähnten Software WEKA 3 die Bestimmung der Logit-Koeffizienten $\beta_0, \ldots, \beta_5$. Tabelle 4.2 zeigt das berechnete Ergebnis.

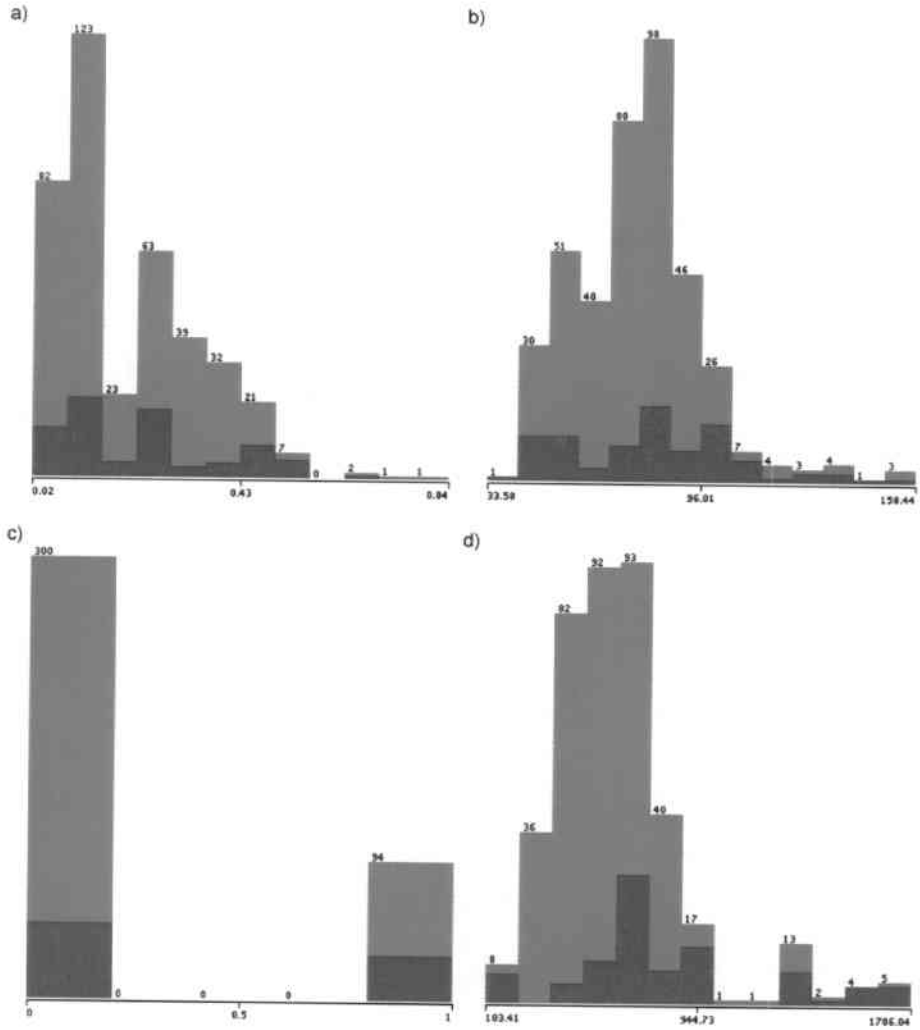

Abbildung 4.2: Anzahl der Hypothesen in Abhängigkeit der Merkmalsausprägungen: a) Distanzabweichung, b) Grauwertmittel, c) Erstklassifikation und d) Nachbardistanz. Die blauen Bereiche kennzeichnen fehlerhafte, die roten korrekte Hypothesen.

Konfidenzmerkmal	i	Koeffizient β_i	Ø-Merkmalswert d_i	$\beta_i \cdot d_i$
	0	3,0146		
Distanzabweichung	1	−0,6879	0,212	−0,146
Grauwertmittel	2	0,0131	77,510	1,015
Erstklassifikation	3	−0,6259	0,239	−0,150
Nachbardistanz	4	−0,0035	629,975	−2,205

Tabelle 4.2: Logit-Koeffizienten $\beta_0, \ldots, \beta_5$ zur Berechnung der logistischen Regressionskurve

Die Resultate zeigen, dass mit steigender Ausprägung der Merkmale „Distanzabweichung", „Erstklassifikation" und „Nachbardistanz" die berechnete Konfidenz gemäß Gleichung 2.12 abnimmt. Im Gegensatz dazu steigt die Konfidenz mit steigendem Wert der Variable „Grauwertmittel". Dies deckt sich mit den Erwartungen in diese Merkmale. Der Einfluß eines Merkmals auf die Konfidenz lässt sich anhand des absoluten Wertes eines Logit-Koeffizienten nicht bewerten, da die zugehörigen Merkmalsausprägungen nicht normiert sind. Multipliziert man hingegen den Regressionskoeffizienten mit dem Mittelwert der entsprechenden Variable, so sagt dieses Produkt sehr wohl etwas über den relativen Einfluss aus. Laut den vorliegenden Daten hat die Variable „Nachbardistanz" den stärksten Effekt, gefolgt von der Variable „Grauwertmittel". Die Merkmale „Distanzabweichung" und „Erstklassifikation" haben hingegen verhältnismäßig geringen Einfluss auf die berechnete Konfidenz der Einzelbildhypothese.

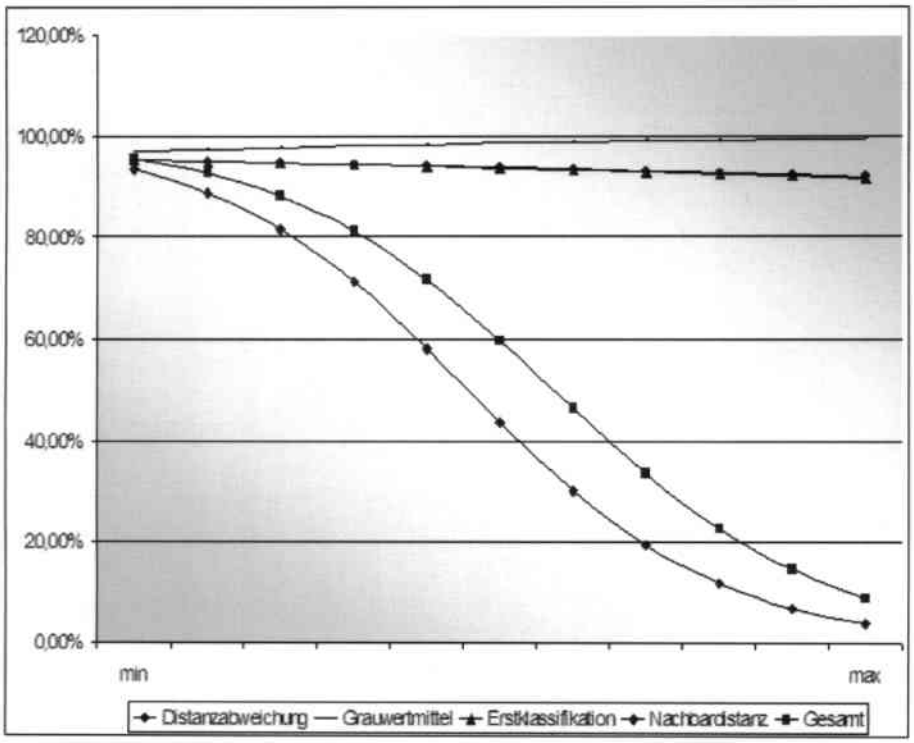

Abbildung 4.3: Logistische Funktion der Einzelbildkonfidenz für berechnete β_i. Jede Variable wurde dabei für sich allein ohne Berücksichtigung der übrigen Variablen betrachtet. Die Gesamtkurve zeigt den Verlauf der logistischen Regressionskurve für alle Variablen mit linearem Anstieg

Abbildung 4.3 zeigt für jedes Merkmal die zugehörige logistische Funktion. Die jeweils übrigen Variablen wurden nicht berücksichtigt. Der Wert des betrachteten Merkmals reicht dabei vom jeweiligen Minimum bis zum Maximum und steigt linear an. Desweiteren zeigt die Grafik auch den Verlauf der logistischen Funktion unter Berücksichtigung aller vier Variablen. Die Werte der Variablen steigen in diesem Fall für alle linear vom Minimal- zum Maximalwert.

4.3.3 Bestimmung der Logit-Koeffizienten zur Berechnung der Sequenzhypothesenkonfidenz

Aus Gründen der Vollständigkeit sei an dieser Stelle auch die Bestimmung und Beschreibung der Logit-Koeffizienten zur Berechnung der Sequenzhypothesenkonfidenz erwähnt, obwohl dazu das noch nicht erwähnte Evaluationsergebnis der Sequenzhypothese benötigt wird. Jedoch wurden nur 50 % des Evaluationsatensatzes verwendet, um die Regressionskoeffizienten zu ermitteln. Die andere Hälfte findet wiederum bei der Evaluation der Sequenzhypothesekonfidenz Verwendung. Abbildung 4.4 zeigt die Verteilung der Sequenzhypothsen in Abhängigkeit der beiden betrachteten Merkmale. Auf der X-Achse ist die Variable „Übereinstimmung", auf der Y-Achse die Variable „Stabilität" aufgetragen. Die Kreuze entsprechen den erstellten Sequenzhypothesen, wobei blau fehlerhafte und rot korrekte Klassifikationen kennzeichnet. Man erkennt deutlich eine Konzentration korrekter Annahmen in den Bereichen, die hohe Ausprägungen der betrachteten Merkmale voraussetzen.

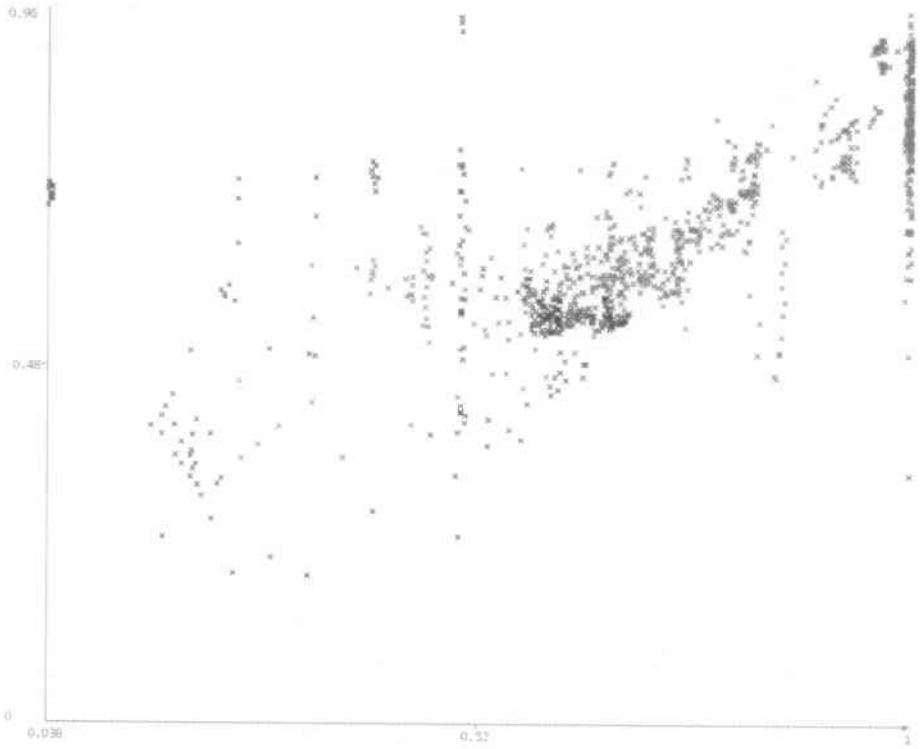

Abbildung 4.4: Verteilung der Sequenzhypothesen in Abhängigkeit der betrachteten Merkmale „Übereinstimmung" und „Stabilität"

Die Ermittlung der Logit-Koeffizienten erfolgt wiederum wie zuvor. Tabelle 4.3 zeigt das Ergebniss für β_0, β_1 und β_2.

Da beide Variablen normalisiert sind, also nur Werte zwischen 0 und 1 annehmen, kann der Einfluss auf die Konfidenz direkt anhand des Logit-Koeffizienten abgelesen werden. Beide Merkmale weisen einen starken positiven Zusammenhang auf, d.h. mit steigenden Variablenwerten erhöht sich auch die Eintritts-

Konfidenzmerkmal	i	Koeffizient β_i
	0	−9,291
Übereinstimmung	1	7,8473
Stabilität	2	8,6639

Tabelle 4.3: Logit-Koeffizienten β_0, β_1 und β_2 zur Berechnung der logistischen Regressionskurve für die Bestimmung der Sequenzhypothesenkonfidenz

wahrscheinlichkeit einer korrekten Sequenzhypothese. Abbildung 4.5 zeigt den Verlauf der logistischen Regression in Abhängigkeit der Variablen „Stabilität" (X) und „Übereinstimmung" (Y).

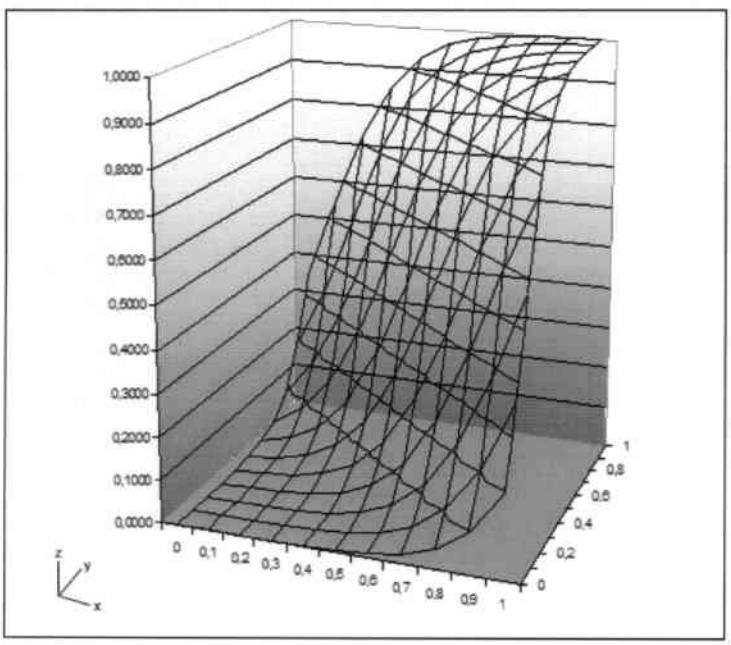

Abbildung 4.5: Verlauf der logistischen Regressionfunktion zur Berechnung der Sequenzhypothesenkonfidenz in Abhängigkeit der Variablen „Stabilität" (X) und „Übereinstimmung" (Y).

4.4 Evaluation

Auf dem dritten Datenset wurden letztendlich die zuvor vorgestellten Methoden zur Bestimmung der Sequenzhypothese evaluiert und mit etablierten Standardverfahren verglichen. Bei diesen Standardverfahren handelt es sich um die im Kapitel 2.6 vorgestellten Methoden zur Schätzung der a posteriori Wahrscheinlichkeit. Die Ergebnisse sind in Abbildung 4.6 dargestellt. Der erste Balken bezieht sich auf die Erfolgsquote der Einzelbildhypothese unter Verwendung eines 10-Nearest-Neighbour-Klassifikators. Dieser ist notwendigerweise für die Referenzverfahren einzusetzen, da diese zur Berechnung der a posteriori Wahrscheinlichkeit die Klassen bzw. Distanzen der $k = 10$ nächsten Nachbarn benö-

tigen. Die Erfolgsquoten der Standardverfahren zeigen die folgenden Balken von Darstellung 4.6.

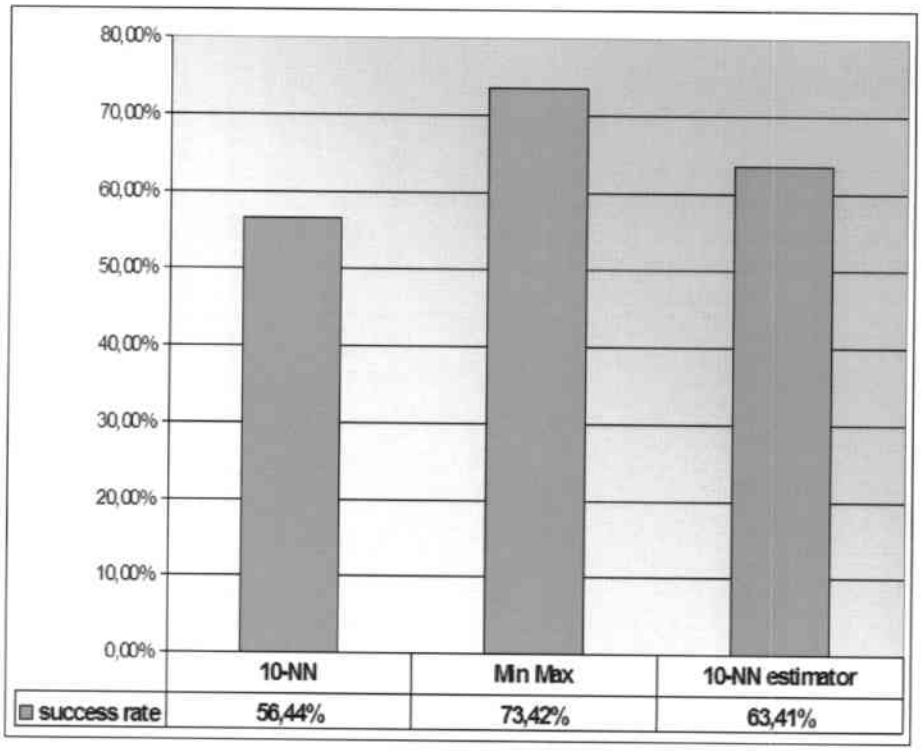

Abbildung 4.6: Erfolgsquoten des 10-Nearest-Neighbour-Klassifikators bei der Erstellung der Einzelbildhypothese, sowie Erfolgsquoten für die Verwendung der Summenregel mit normalisierten Distanzen bzw. 10-Nearest-Neighbour-Schätzer.

Die Ergebnisse, die mit dem Verfahren zur Bildung einer Sequenzhypothese aus Kapitel 3 erreicht wurden, sind in Abbildung 4.7 dargestellt. Der erste Balken bezieht sich auf die Erfolgsquote der Einzelbildhypothese unter Verwendung eines einfachen Nearest-Neighbour-Klassifikators. Die folgenden Balken zeigen die Erfolgsquote der Sequenzhypothese mit unterschiedlichen Konfigurationen, die unterhalb des Balkens verzeichnet sind. Dabei gibt die erste Zahl die Fensterbreite an, die Prozentangabe bezieht sich auf den einbezogenen besten Anteil an Einzelbildhypothesen.

Vergleicht man die Resultate miteinander, so kommt man zu dem Schluß, dass Einzelbildidentifikation weitaus schlechtere Ergebnisse (61,96%) erzielt als Gesichteridentifikation auf Bildsequenzen. Auch das beste Referenzverfahren - Min-Max-Normalisierung mit 73,42% Erfolgsrate - weist schlechtere Ergebnisse auf als die hier vorgestellten Verfahren. Das beste Ergebnis (77,11%) wird mit einer Fensterbreite von 30 Frames unter Berücksichtigung der besten 3 Einzelbildhypothesen erzielt. Allerdings lässt sich kein signifikanter Unterschied zu den Ergebnissen mit Fensterbreite 20 und Einbeziehung der 10 % besten Einzelbildhypothesen (76,78%) bzw. Einbeziehung aller Hypothesen der gesamten Benutzersession (76,62%) feststellen.

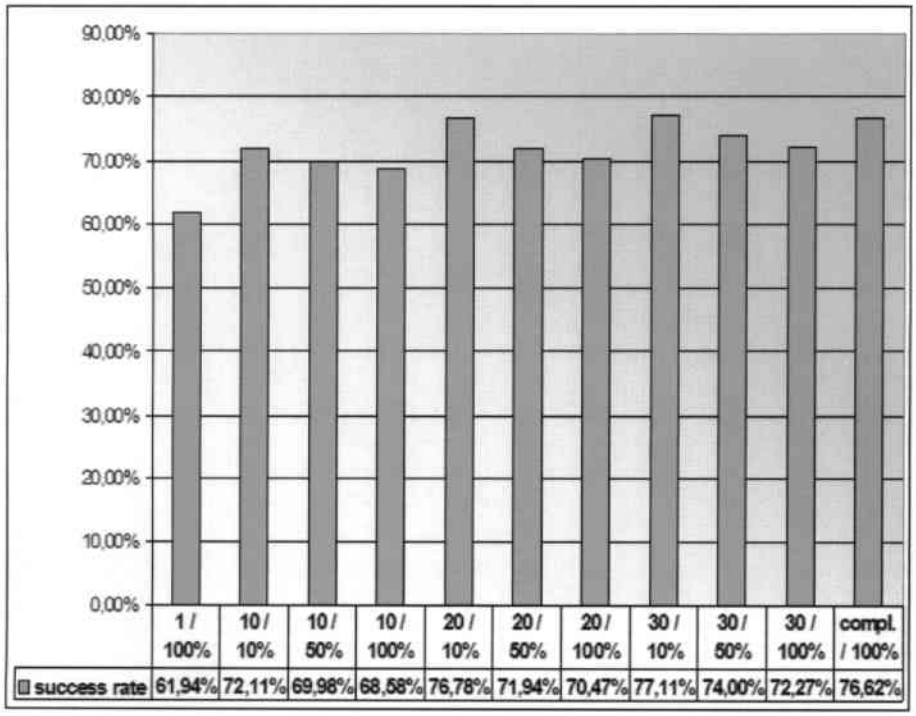

Abbildung 4.7: Erfolgsquoten unterschiedlicher Konfigurationen bei der Bestimmung der Sequenzhypothese. Die erste Zahl entspricht der Fensterbreite, die zweite bezieht sich auf den Anteil der einbezogenen besten Einzelbildhypothesen.

Um die Aussagekraft der Sequenzhypothesenkonfidenz zu evaluieren, wurde wie bereits erwähnt die Hälfte des Evaluationssets verwendet. Dazu wurden sowohl für falsche als auch für korrekte Sequenzhypothesen Mittelwert und Standardabweichung der Sequenzhypothesenkonfidenz ermittelt. Das Ergebnis kann Tabelle 4.4 entnommen werden und zeigt, dass die erstellte Konfidenz ein vernünftiges und verlässliches Maß darstellt, das Vertrauen in eine Hypothese zu modellieren.

Sequenzhypothese	Mittelwert	Standardabweichung
korrekt	0,76	0,32
falsch	0,43	0,31

Tabelle 4.4: Mittelwert und Standardabweichung der Sequenzhypothesenkonfidenz für korrekte respektive falsche Sequenzhypothesen

Kapitel 5

Diskussion und Ausblick

In diesem Kapitel werden zunächst die erzielten Ergebnisse diskutiert, bevor diverse Ansatzpunkte für zukünftige Arbeiten die Gesichteridentifikation betreffend aufgezeigt werden.

5.1 Diskussion der Ergebnisse

Ein große Herausforderung in der vorliegenden Arbeit bestand darin, mit den unterschiedlichen Bedingungen, die während der Aufnahmen der Bilddaten vorherrschten, zurecht zu kommen. Zum einen entstanden Probleme durch stark differierende Lichtverhältnisse, da die Aufnahmen in einem Institutsflur bei künstlichem Licht und Tageslicht gemacht wurden. Insbesondere da die ersten Aufnahmen im Juni, die letzen im November stattfanden. Desweitern wurden die Daten einem Wizard-of-Oz Experiment entnommen, bei dem die Probanden den Roboter passieren, mit ihm sprechen und wieder verlassen mussten. Die Konsequenz daraus ist eine große Anzahl an verschwommenen Bildern, die entweder durch sich bewegende Probanden oder gar von einer sich drehenden Roboterkamera erzeugt wurden. Auch wechselnde Hintergründe und rasche Veränderungen des Lichteinfalls sind die Folge. Zuguterletzt kann das Sprechen an sich einen negativen Einfluss auf die Einzelbilderkennung haben, die während der hier durchgeführten Tests im Vergleich zu vorherigen Evaluationen (z.B. FRGC[1]) [2, 4] verhältnismäßig schlecht abschneidet. Nichtsdestotrotz belegen die zuvor vorgestellten Evaluationsergebnisse, dass die Gesichteridentifikation auf Bildsequenzen in der Mensch-Roboter-Interaktion durchaus beachtliche Verbesserungen zur Einzelbildidentifikation erzielt.

Im Ablauf der Gesichtsidentifikation auf Bildsequenzen stellt die Berechnung der Einzelbildkonfidenz einen entscheidenden Vorgang dar. Insbesondere die Prädiktorvariablen haben einen signifikanten Einfluss auf das letztliche Klassifikationsergebnis der Sequenzhypothese. Die Beschränkung auf eine kleine Auswahl dieser Merkmale fällt allerdings nicht leicht, da eine sehr große Anzahl an denkbaren Variablen zur Verfügung steht. Eine Prämisse bei der Selektion der Merkmale zur Berechnung der Einzelbildkonfidenz war stets, nur Eigenschaften des einzelnen Bildes widerzuspiegeln, wohingegen die Merkmale der Sequenzkonfidenz möglichst Eigenschaften einer Sequenz von Bildern erfassen

[1] face recognition grand challenge

sollten. Schließlich zeigen die Ergebnisse, dass die Wahl der unabhängigen Variable für die Konfidenzbestimmung sinnvoll getroffen wurde, da die mitsamt dem eigentlichen Klassifikationsergebnis berechnete Konfidenz eine vernünftige Einschätzung des Vertrauens in die erstellte Annahme wiedergibt.

Zuguterletzt sollte erwähnt werden, dass bei der Evaluation der Bildsequenzen immer wieder Frames oder sogar Framesequenzen enthalten waren, auf denen keine Einzelbildidentifikation durchgeführt werden konnte, obwohl sich tatsächlich eine Person vor dem Roboter befand. Die Gründe hierfür sind vielfältig und können so profan sein, wie ein zur Seite gedrehtes Gesicht oder eine durch Bewegung verursachte schlechte Bildqualität. Bei allen Tests wurden diese Frames nicht beachtet. Würde man diese in die Evaluation mit einbeziehen, hätte das natürlich Auswirkungen auf das Klassifikationsergbnis. Die Einzelbildidentifikation würde schlechtere Resultate liefern, da eine Nichtidentifikation natürlich auf keinen Fall mit der Realität übereinstimmt. Die Gesichteridentifikation auf Bildsequenzen wird aber tendenziell gleich bleiben, da die Sequenzhypothese im Falle eines „leeren" Frames weiterhin bestehen bleibt. Die relative Verbesserung gegenüber der Einzelbildidentifikation würde also zunehmen und die Aussage, dass Gesichteridentifikation auf Bildsequenzen bessere Ergebnisse liefern, sogar unterstrichen werden.

5.2 Ausblick

Für die Zukunft sind zunächst weitere Experimente und Datensammlungen unter unterschiedlichen Bedingungen geplant, welche die Richtigkeit der vorgestellten Ergebnisse bestätigen und die Integration der beschriebenen Gesichteridentifikation in das Robotersystem erfolgreich belegen sollen. Insbesondere die Auswahl der Merkmale zur Konfidenzberechnung könnten durch weitere Betrachtungen und Evaluationen fundiertere Gültigkeit erlangen.

Desweitern ist es vonnöten, sich weitere Gedanken um die Eingliederung neuer Personen in die Datenbank und die dadurch immer größer anwachsenden Datenbestände zu machen. Auf der einen Seite ermöglicht es uns der durch die DCT erzeugte Merkmalsvektor zusammen mit dem instanzbasierten Lernverfahren, leicht und unkompliziert neue Instanzen in die Datenbasis aufzunehmen. Auf der anderen Seite entsteht so schnell eine außerordentlich große Menge an Trainingsinstanzen, die bei jeder Klassifikation des k-Nearest-Neighbour-Klassifikators komplett verarbeitet werden muss. Ein möglicher Ansatz ist, klassenintern auf höherer Hierarchieebene ein zweites Mal zu clustern. Allerdings besteht die Gefahr, dass die Merkmalsvektoren einer Person einer multivariate Verteilung entsprechen, z.B. durch unterschiedliche Lichtverhältnisse bei den Aufnahmen. Bei einer schlecht gewählten Anzahl an Clustern würde ein Zentroid die eigentlichen Ausprägungen des Vektors nicht gut repräsentieren und als Folge die Klassifikationsgüte verschlechtern.

Schließlich ist die Erstellung einer Verifikationskomponente von großem Interesse. Mit deren Hilfe könnte das Gesichteridentifkationssystem selbständig entscheiden, ob die sich vor dem Roboter befindliche Person dem System unbekannt ist oder nicht. Ein Ansatz, dieses Problem mit Hilfe der Einführung eines Schwellwertes zu lösen, wurde bereits in [7] verfolgt. Diese Vorgehensweise verlangt aber noch weitere Evaluationen.

Kapitel 6

Zusammenfassung

Im Verlauf dieser Studienarbeit wurde ein Gesichteridentifikationsmodul für einen mobilen Roboter entwickelt, welches während der Mensch-Roboter-Interaktion Hypothesen über die beteiligte Person generiert. Zur Erstellung dieser Annahmen werden nicht nur einzelne Bilder in Betracht gezogen, stattdessen erfahren ganze Bildsequenzen der Interaktion Beachtung. Das in dieser Arbeit vorgestellte Verfahren zur Bestimmung einer Sequenzhypothese basiert dabei auf der Kombination der Einzelbildklassifikationsergebnisse, die mit einem individuellen Konfidenzfaktor gewichtet werden. Die Konfidenzen werden dabei für jedes Bild mit Hilfe eines logistischen Regressionsmodells bestimmt. Diese Vorgehensweise erweist sich als sinnvoll, da die durchgeführte Evaluation der Gesichteridentifikation auf Bildsequenzen signifikant bessere Klassifikationsergebnisse gegenüber der Gesichtserkennung auf Einzelbildern liefert.

Darüberhinaus berechnet das entwickelte Modul Konfidenzen für die aufgestellten Sequenzhypothesen, die erneut mit Hilfe der logistischen Regression bestimmt werden. Die Evaluation zeigt, dass die Sequenzhypothesenkonfidenz ein verlässliches Maß für die Korrektheit dieser Annahmen darstellt. Andere Komponenten auf dem Robotersystem, die mit dem Gesichteridentifikationsmodul über den ONE4ALL-Server kommunizieren, werden dadurch in die Lage versetzt, die generierten Sequenzhypothesen zu verwerfen oder zu übernehmen.

Ferner ist das Modul, mit Unterstützung des auf dem Roboter vorhandenen Dialogsystems, in der Lage, neue Benutzer in die bestehende Datenbank einzufügen.

Literaturverzeichnis

[1] H.K. Ekenel and Q. Jin. Isl Person Identification Systems in the clear Evaluations. 2006.

[2] H.K. Ekenel and R. Stiefelhagen. A Generic Face Representation Approach for Local Appearance based Face Verification. *Proceedings of the CVPR IEEE Workshop on FRGC Experiments, San Diego, CA, USA*, 2005.

[3] H.K. Ekenel and R. Stiefelhagen. Local appearance based face recognition using discrete cosine transform. In *Proceedings of the 13th European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, 2005.

[4] H.K. Ekenel and R. Stiefelhagen. Analysis of Local Appearance-based Face Recognition on frgc 2.0 Database. *Face Recognition Grand Challenge Workshop (FRGC), Arlington, VA, USA*, 2006.

[5] I. Witten et.al. http://www.cs.waikato.ac.nz/~ml/weka/index.html, Stand: Oktober 2006.

[6] Y. Freund and R.E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Computational Learning Theory: Eurocolt 1995*, pages 23–27, 1995.

[7] H.K. Ekenel C. Schaa H. Holzapfel, T. Schaaf and A. Waibel. A Robot learns to know people - First Contacts of a Robot. *KI 2006, 29th annual German Conference on Artificial Intelligence*, 2006.

[8] H. Holzapfel. Building multilingual spoken dialogue systems. *Special issue of Archives of Control Sciences, G.eds. Z. Vetulani*, 4, 2005.

[9] R.P.W. Duin J. Kittler, M. Hatef and J. Matas. On Combining Classifiers. *IEEE Transactions on Pattern Analyses and Machine Intelligence*, Vol. 20, March 1998.

[10] J.F. Kelley. An Iterative Design Methodology for User-Friendly Natural Language Office Information Applications. *ACM Transactions on Office Information Systems*, Vol. 2:26–41, March 1984.

[11] T.M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[12] N.N. http://www.lrz-muenchen.de/~wlm/ilm_l11.htm, Stand: 14. Juni 2004.

[13] N.N. http://www.cs.cf.ac.uk/Dave/Multimedia/node238.html, Stand: 2005.

[14] N.N. http://de.wikipedia.org/wiki/Diskrete_Kosinustransformation, Stand: 2006.

[15] N.N. http://http://de.wikipedia.org/wiki/K-means, Stand: 2006.

[16] N.N. http://www.sonydigital-link.com/aibo/index.asp, Stand: 2006.

[17] Christoph Schaa. Proaktive Initiierung von Dialogen für humanoide Roboter. Master's thesis, Universität Karlsruhe (TH), 2005.

[18] P. Viola and M.J. Jones. Robust real-time object detection. *Cambridge Research Laboratory, Technical Report Series*, 2001.