Universität Karlsruhe
Fakultät für Informatik
Institut für Logik, Komplexität und Deduktionssysteme
Prof. A. Waibel

WS 2004/05

Studienarbeit

Implementierung und Evaluation eines Systems zur Erkennung von Gesichtern

Dennis Harres

Oktober 2004

Betreuer: Dipl.-Inf. K. Nickel
Dr.-Ing. R. Stiefelhagen
Prof. Dr. A. Waibel

Inhaltsverzeichnis

1 Einleitung

Schon seit Beginn des Computerzeitalters versuchen die Menschen, einer Maschine, die zuerst nur dafür gedacht war, zu rechnen, Dinge beizubringen, die einem Menschen eigen sind. Bereits Science-Fiction-Autoren und Hollywood-Regisseure sahen Roboter in menschlicher Gestalt, ausgestattet mit menschlichen Fähigkeiten, wie Sehen, Hören, Sprechen. Sie konnten sich eigenständig fortbewegen, mit Objekten umgehen und mit Menschen interagieren. Auf den ersten Blick unkompliziert, gewöhnlich und selbstverständlich.

Das menschliche Gehirn beeindruckt mit seiner Vielseitigkeit und Schnelligkeit. Seit unserer Geburt lernen wir kontinuierlich Tag für Tag. Die Instinkte, über die wir seit der Geburt verfügen, stellen einen minimalen Anteil von dem dar, was wir später dazulernen. So ist unser Sehsystem zuerst in der Lage, hell und dunkel zu unterscheiden, Farbflächen zu erkennen, danach gewisse Muster, Umrisse und Objekte. Wir lernen das Gesicht der Mutter und ihre Mimik erkennen, und lächeln zurück, wenn sie lächelt. Dies einem Computer beizubringen stellt eine große Herausforderung für die Wissenschaftler auf der ganzen Welt dar.

Aktuelle Forschung in der Gesichtserkennung ist damit beschäftigt, die unzureichenden Ergebnisse der bisherigen Verfahren auch bei uneingeschränkten Bedingungen zu verbessern und neue von den äußeren Faktoren unabhängige Verfahren zu entwickeln. Hierbei stößt man nicht nur an Grenzen der Hardwareebene, wie eine zu geringe Auflösung der für die Sehsysteme eingesetzten Kameras, Bandbreitenprobleme oder Leistungsfähigkeit aktueller Prozessoren. Auf der Software-Ebene mangelt es an Algorithmen, die in der Lage wären, Gesichter ansatzweise mit der Genauigkeit des menschlichen Gehirns zu erkennen.

Die Gesichtserkennung ist unentbehrlich für Interaktions- und Kommunikationssysteme. Denn die Identifikation des gesehenen Subjektes ist einer der wichtigsten Punkte für den Kontext des Geschehens. In unserem Alltag herrschen ungezwungene Kommunikation und Handlungsfreiheit, somit ist es wichtig, diese Kriterien auch in die Mensch-Maschine-Kommunikation zu integrieren. Man erwartet von diesen Vorgängen, dass sie ohne feste Regeln und unbemerkt erfolgen. Es wird angenommen, dass ein intelligenter Raum alle Konferenzteilnehmer während einer Diskussion erkennt, und dafür nicht ein bewusstes Blicken in die Kamera voraussetzt.

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines Systems zur Erkennung menschlicher Gesichter, unabhängig von ihrer Kopfrotation und den

im Raum herrschenden Beleuchtungsverhältnissen. Auf den von einer Videokamera gelieferten Bildern werden Gesichter gesucht und Bild für Bild verfolgt. Das gefundene Gesicht passiert die Vorverarbeitung und gelangt in den Erkenner, wo es mit den in der Datenbank gespeicherten Aufnahmen verglichen wird und als Ergebnis eine Zuordnung zu einer Person erfolgt. So ein System kann vielfältig eingesetzt werden, als Beispiele seien hier mobile Roboter, intelligente oder multimodale Räume genannt.

Die Arbeit gliedert sich in folgende Teile:

Kapitel 2 stellt die wichtigsten der in diesem System eingesetzten Algorithmen vor.

Kapitel 3 beschreibt den Aufbau und alle funktionellen Vorgänge des Systems, die notwendig sind, bevor die Erkennung durchgeführt werden kann.

Kapitel 4 beschäftigt sich mit einer Reihe von Tests, um das System auf mögliche Probleme und Schwachstellen hin zu untersuchen und Erkenntnisse für einen optimalen Einsatz zu gewinnen.

Kapitel 5 gibt eine kurze Zusammenfassung über das Erreichte und spricht die aufgetretenen Probleme an.

2 Algorithmen und Techniken

In dieser Arbeit wurde der *Eigenfaces* Ansatz [TP91] für die Gesichtserkennung implementiert. Dieser wird im Folgenden genau vorgestellt. Ausserdem werden zwei im System eingesetzte Methoden der Gesichtslokalisierung (*face tracking*) beschrieben.

2.1 Eigenfaces

Eigenfaces stellt ein Verfahren dar, welches es ermöglicht, relevante Informationen aus dem Bild eines Gesichtes zu extrahieren, möglichst effizient zu kodieren, und mit den in der Datenbank genauso kodierten Gesichtern zu vergleichen. Im mathematischen Sinne repräsentiert *Eigenfaces* eine Methode, die Hauptachsen der Verteilung der Gesichter im Raum zu finden, bzw. die Eigenvektoren einer aus Bildern bestehenden Kovarianzmatrix. Dabei wird jedes Bild als ein Punkt (oder ein Vektor) im höherdimensionalen Raum betrachtet. Jeder dieser Eigenvektoren ist ein Merkmal, welches Abweichungen zwischen Gesichtsbildern beschreibt. Da die Eigenvektoren auf Aufnahmen der Gesichter basieren, stellen sie selbst eine Art Geister*gesicht* (ghost *face*) dar, und werden somit „Eigen*faces*" genannt. Jedes einzelne Gesicht kann in Form einer Linearkombination von Eigenfaces genau dargestellt werden. Allerdings ist es meistens ausreichend, wenn es durch die „besten" Eigenfaces approximiert werden kann. Die besten Eigenfaces sind solche, die die höchsten Eigenwerte besitzen und somit für größte Unterschiede unter den Bildern stehen. In der Abbildung 2.1 ist zu sehen, wie Eigenfaces und durch sie repräsentierte Bilder aussehen können.

Sei ein Bild $I(x, y)$ ein zweidimensionales $N \times N$ Feld von 8-Bit Helligkeitswerten. Dieses Bild kann auch als ein N^2-dimensionaler Vektor betrachtet werden, dann würde eine Bildermenge eine Häufung von Punkten im höherdimensionalen Raum darstellen. Durch ihre Ähnlichkeit würden aber diese Bilder nicht willkürlich in diesem Raum liegen, und könnten somit durch einen relativ kleindimensionalen Untervektorraum beschrieben werden. Das Konzept der Hauptkomponentenanalyse (PCA: *principle component analysis*) besteht darin, jene Basisvektoren zu finden, die am besten diese Verteilung im gesamten Raum beschreiben. Diese Vektoren definieren einen Untervektorraum der Gesichtsbilder, der im Folgenden „Gesichtsraum" (*face space*) genannt wird.

Sei die Trainingsmenge[1] die Gesichtsbilder $\Gamma_1, \Gamma_2, \Gamma_3, \ldots, \Gamma_M$. Das Durchschnittsgesicht dieser Menge ist definiert als $\mathbf{\Psi} = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$. Jedes Gesicht unterscheidet sich vom Durchschnitt durch den Vektor $\mathbf{\Phi}_i = \Gamma_i - \mathbf{\Psi}$. Die PCA sucht nach M orthonormalen Vektoren $\mathbf{u}_k$, die am besten die Datenverteilung im Raum beschreiben. Der k-te Vektor, $\mathbf{u}_k$, wird so gewählt, dass

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(\mathbf{u}_k^T\mathbf{\Phi}_n)^2 \tag{2.1}$$

ein Maximum ist, wobei

$$\mathbf{u}_l^T\mathbf{u}_k = \delta_{lk} = \begin{cases} 1, & l = k \\ 0, & sonst \end{cases} \tag{2.2}$$

Die Vektoren $\mathbf{u}_k$ und Skalare λ_k sind jeweils die Eigenvektoren und die Eigenwerte der Kovarianzmatrix

$$C = \frac{1}{M}\sum_{n=1}^{M}\mathbf{\Phi}_n\mathbf{\Phi}_n^T = AA^T \tag{2.3}$$

mit der Matrix $A = (\mathbf{\Phi}_1|\mathbf{\Phi}_2|\ldots|\mathbf{\Phi}_M)$. Die Matrix C ist somit $N^2 \times N^2$. Gibt es weniger Bilder im Bildraum als dessen Dimension ($M < N^2$), existieren nur $M - 1$, statt N^2, nicht triviale Eigenvektoren. Um N^2-dimensionale Eigenvektoren zu berechnen, lösen wir zuerst die Eigenvektoren einer $M \times M$ Matrix und greifen dann auf entsprechende Linearkombinationen der Gesichtsbilder $\mathbf{\Phi}_i$. Seien die Eigenvektoren $\mathbf{v}_i$ von A^TA so, dass

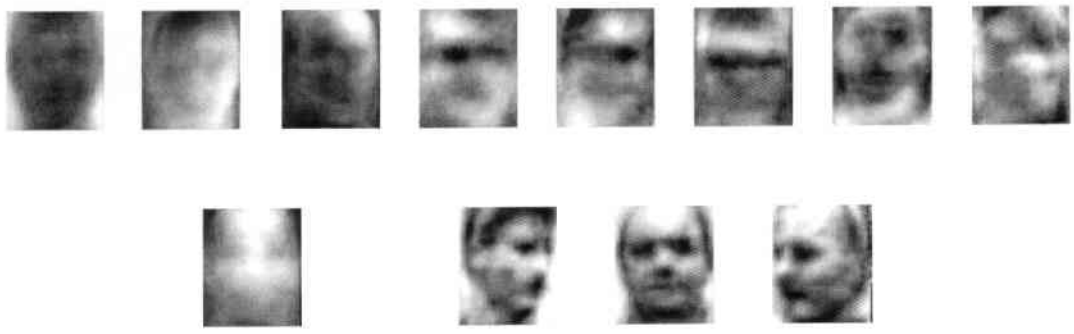
$$A^TA\mathbf{v}_i = \mu_i\mathbf{v}_i \tag{2.4}$$

und nach Multiplikation mit A

$$AA^TA\mathbf{v}_i = \mu_iA\mathbf{v}_i \tag{2.5}$$

Hier sehen wir, dass $A\mathbf{v}_i$ die Eigenvektoren von $C = AA^T$ sind. Wir bilden eine $M \times M$ Matrix $L = A^TA$, mit $L_{mn} = \mathbf{\Phi}_m^T\mathbf{\Phi}_n$ und finden M Eigenvektoren $\mathbf{v}_l$ von L. Sie bestimmen die Linearkombinationen von M Gesichtsbildern aus dem Trainingssatz für die Eigenfaces $\mathbf{u}_l$:

[1] In weiteren Kapiteln wird diese als Referenzmenge bezeichnet, da ein weiterer Trainingssatz für andere Aufgaben benötigt wird.



Obere Reihe: die besten acht Eigenfaces, absteigend nach Eigenwerten sortiert
Untere Reihe links: das zugehörige Durchschnittsgesicht
Untere Reihe rechts: drei Beispiele für approximierte Darstellung durch Linearkombination von Eigenfaces

Abbildung 2.1: Beispielbilder aus dem Eigenfaces-Ansatz

$$\mathbf{u}_l = \sum_{k=1}^{M} \mathbf{v}_{lk} \mathbf{\Phi}_k, \qquad l = 1, \ldots, M \tag{2.6}$$

Mit diesem Verfahren ist der Rechenaufwand erheblich reduziert, da in den meisten Fällen $M \ll N^2$ ist. Weiterhin reicht es meistens nur die $M' < M$ wichtigsten Eigenfaces zu nehmen. Der zugehörige Eigenwert steht für die Relevanz des Eigenvektors.

Um ein neues Gesichtsbild (Γ) in seine Eigenface-Komponenten zu zerlegen (in den Gesichtsraum zu projizieren), wird eine einfache Operation ausgeführt:

$$\omega_k = \mathbf{u}_k^T (\Gamma - \mathbf{\Psi}) \tag{2.7}$$

für $k = 1, \ldots, M'$. Dies ergibt einen Vektor $\Omega^T = |\omega_1 \omega_2 \ldots \omega_{M'}|$, der für Klassifikation sowie Erkennung verwendet werden kann.

2.2 Gesichtslokalisierung

Neben dem Erkennungsmodul benötigt das System eine Komponente, die die Daten aufbereitet, d.h. Gesichter in den bereitgestellten Aufnahmen findet und sie als für die Erkennung geeignet auswählt. In dieser Arbeit wurden zwei im Folgenden erläuterte Methoden für diese Aufgabe eingesetzt.

2.2.1 3D-Face-Tracker

Dieser Trackingansatz aus [Nic02] beschäftigt sich mit der Kombination von Tracking mit Hautfarbe und Stereobildverarbeitung.

Für die Segmentierung durch Hautfarbe wird eine Transformation der Eingangsbilder in den *chromatischen* Farbraum[2] erforderlich. Dadurch erreichen wir nicht nur eine verringerte Streuung der Farbvektoren, sondern eliminieren auch die Helligkeitskomponente, und konzentrieren uns nur auf den Farbwert.

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}. \tag{2.8}$$

Die Verteilung der rg-Farben wird in einem nicht-parametrischen Modell, einem zweidimensionalen Histogramm gespeichert. Im ersten Histogramm werden die Hautfarben (Auswahl eines Bildausschnittes z.B. durch den Benutzer) gespeichert, im anderen die „Nicht-Haut"-Farbwerte. Die dadurch ableitbaren Wahrscheinlichkeiten werden für die Klassifikation herangezogen. Ist das Verhältnis der „Haut"-Wahrscheinlichkeit eines Pixelfarbwertes zur „Nicht-Haut"-Wahrscheinlichkeit größer als ein gewisser Faktor, wird dieser Pixel als Hautfarbe klassifiziert. Mit Hilfe von morphologischen Operatoren wird die „pixelige" Struktur nach der Klassifikation zu einer flächigen Struktur. Nun wird der größte zusammenhängende Block mit Hautfarbe gesucht – der Kopf, alle anderen als Hautfarbe markierten Bereiche werden verworfen. Dieser wird in die Hautfarbmaske übernommen.

Eine Stereokamera, also zwei in einer Ebene parallel ausgerichtete Kameras, liefert zwei zum gleichen Zeitpunkt aufgenommene Bilder. Sie unterscheiden sich nur in der Perspektive der jeweiligen Kamera. Dieser Versatz macht es möglich, ein Disparitätenbild zu erstellen. Es ist ein Graustufenbild, welches die Tiefeninformation aus den korrespondierenden Punkten des linken und rechten Bildes durch Helligkeit repräsentiert.

Kombiniert man nun die Hautfarbmaske aus dem ersten Verfahren mit dem Tiefenbild, so gelingt es, viele Probleme der Farbsegmentierung zu umgehen oder zumindest zu minimieren. So werden beispielsweise hautfarbähnliche Bereiche im Hintergrund hinter der aufgenommenen Person durch Hinzunahme der Tiefenmaske verworfen, vorausgesetzt die Person hat einen ausreichenden Abstand zum Hintergrund.

Aus der Farbmaske und dem Tiefenbild resultiert ein Rechteck, das den Kopf eingrenzt. Dieses bestimmt den Ausschnitt, der an den Erkenner übergeben wird.

[2] auch *rg*-Farbraum genannt

2.2.2 Elliptical Head Tracking

Dieser zweite Ansatz für die Gesichtslokalisierung geht aus [Bir98] hervor. Hier wird eine andere Kombination analysiert: als erste Komponente wieder Hautfarbe aber in Verbindung mit einem Kantenbild. Dieser Ansatz setzt eine Bildersequenz bzw. Videoaufnahme voraus, weil er in der Umgebung des letzten Ergebnisses sucht[3]. Eine Projektion des Kopfes in einem Bild ist offensichtlich eine Ellipse, und diese wird für die Modellierung verwendet.

Zuerst wird der Helligkeitsgradient berechnet. In der Umgebung des letzten Ergebnisses wird nach einer neuen Position gesucht, sodass der Ellipsenrand eine möglichst hohe Überdeckung mit den Kanten des neuen Bildes erreicht. Die Suche geschieht iterativ im Bereich des vordefinierten Positionsoffsets $\pm\delta$ auf der x-, sowie auf der y-Achse. Für jede Position werden Änderungen der Ellipsengröße im Bereich von $[-\gamma, +\gamma]$ durchprobiert.

Ähnlich wie beim ersten Tracker wird hier ein Farbmodell aus der Startposition[4] erstellt bzw. aus dem letzten Ergebnis aktualisiert. Wie im Schritt davor wird die iterative Verschiebung und Skalierung der Ellipse durchgeführt. Nur jetzt wird eine Bewertung über Hautfarbanteil im Ellipseninneren errechnet.

Durch Kombination beider Ergebnisse wird die neue Position bestimmt. Beide Komponenten ergänzen sich der Art, dass die gefundenen Bildausschnitte ziemlich robust in der Ausrichtung und Größe ausfallen. Mit der Angabe von δ und γ kontrolliert man die maximalen Geschwindigkeiten von Verschiebung und Skalierung und verhindert somit grobe Aussetzer. Nur die schnellen Bewegungen können ein Problem darstellen und werden erst über mehrere Bilder hinweg ausgeglichen.

[3] Somit ist für den Start eine ausreichend präzise Angabe der Position des Kopfes/Gesichtes notwendig.

[4] alle Farbwerte aus dem Inneren der Ellipse

3 Aufbau und Funktionsweise des Systems

Nachdem die wichtigsten Algorithmen vorgestellt wurden, geht es in diesem Kapitel um ihren praktischen Einsatz im Gesichtserkennungsprozess und Aufbau des Systems.

3.1 Datenbank

Im Gesichtserkennungsprozess wird eine Datenbank benötigt, welche Referenzaufnahmen von jeder bekannten Person bei unterschiedlichen Verhältnissen (z.B. Blickwinkel/Beleuchtung) und/oder mit Variationen in der Gesichtsmimik enthält. Häufig eingesetzte und bekannte Beispiele dafür sind die *Yale face database* [GB98] und die *CMU PIE face database* [SBB02].

In der vorliegenden Arbeit wurde eine auf unsere Bedürfnisse angepasste Datenbank entworfen. Aus folgenden Gründen wurde entschieden, die Datenbank durch das Dateisystem und Datentrennung und -gruppierung in Form von Verzeichnissen zu repräsentieren:

- Aufbau und Struktur sind transparent
- Zugehörigkeit und Bedeutung der Daten ist leicht ableitbar
- Alle notwendigen Informationen sind direkt ohne zusätzliche Datenstrukturen lesbar
- Manuelle Eingriffe sind ohne weiteres System oder Frontend möglich
- Bekanntes und intuitives Arbeiten durch Dateioperationen

Die Datenbank wurde in Hinblick auf bevorstehende Experimente wie folgt konzipiert. Zu jeder Person werden die Daten in Trainings- und Evaluationssatz unterteilt. Für Berechnung von Klassenschwellwerten (s. Kapitel 3.6) und für automatische Ermittlung von Referenzen (s. Kapitel 3.7) wird der Trainingssatz verwendet. Alle Experimente werden ausschließlich auf den Daten aus der Evaluationsmenge durchgeführt. Ferner kann jede Person *mehrere* Referenzmengen besitzen – eine höhere Flexibilität und neue Einsatzmöglichkeiten sind gewährleistet.

3.2 Gesichtslokalisierung

Diese Komponente übernimmt das Suchen des Gesichts bzw. des Kopfes im gegebenen Bild und ist selbst kein Bestandteil des Erkenners, sondern liefert diesem den relevanten Bildausschnitt oder speichert ihn in der Datenbank.

Um die Abhängigkeit der Erkennungsraten vom Tracking später untersuchen zu können, werden die beiden im Kapitel 2.2 vorgestellten Verfahren eingesetzt. Es wird eine SVS (*Small Vision System* [Kon97][1]) basierte Stereo-Visualisierung für die Lokalisierungsalgorithmen verwendet. Der „3D-Face-Tracker" setzt die Stereoinformationen voraus, der *Elliptical-Head-Tracking*-Ansatz arbeitet mit dem rektifizierten Bild. Beide benötigen eine manuelle Initialisierung.

3.3 Bildvorverarbeitung

Der größte Nachteil aller bildbasierten Methoden, in unserem Falle *Eigenfaces*, ist die Abhängigkeit von der Normierung der Eingaben. Neben komplexen Problemen, wie Ausrichtung (*alignment*) und Skalierung der Gesichter, existieren Einfachere, die in diesem Kapitel behandelt werden: Qualität der Kamerabilder, bedingt durch Einstellungen der Kamera (z.B. Blende bei Über-/Unterbelichtung) und durch ihre Eigenschaften (Farbrauschen, unterschiedlicher Dynamikumfang des Bildes je nach Inhalt oder Beleuchtung). Um dieses Problem zu minimieren, werden folgende Verfahren der Bildnormierung und -vorverarbeitung vorgeschlagen:

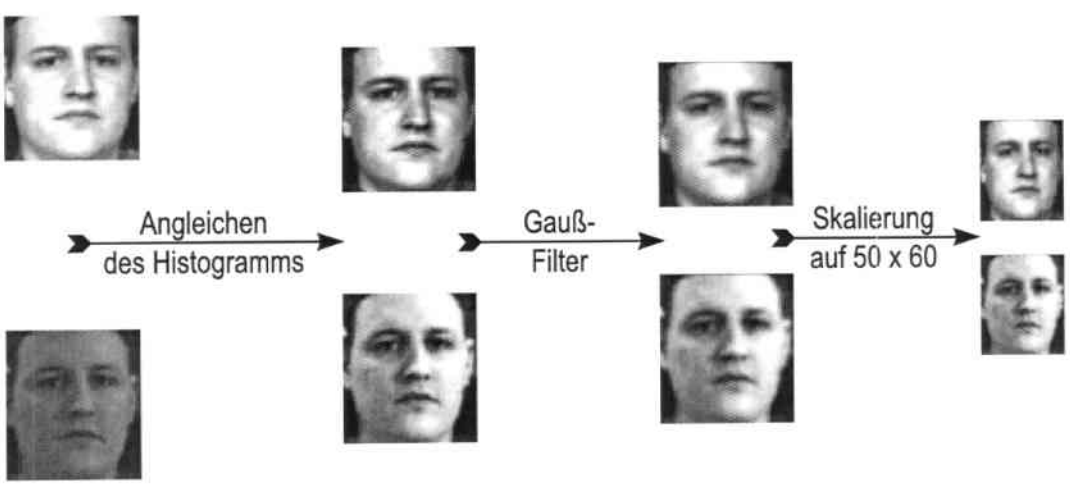


Abbildung 3.1: Bildvorverarbeitung

[1] Weitere Informationen auch unter http://www.ai.sri.com/~konolige/svs/index.htm

Zuerst wird das Histogramm des Bildes angeglichen. Mit diesem Filter wird eine Gleichverteilung der Graustufen erreicht, und somit wird Über- oder Unterbelichtung der Kamerabilder ausgeglichen[2]. Bei unterbelichteten Bildern wird nach der Filteranwendung mit dem Signal auch das Kamerarauschen verstärkt. Um diesem Effekt entgegenzuwirken, sowie generell rauscharme Bilder zu bekommen, wird ein Gauß-Glättungsfilter angewandt.

Für Eigenfaces ist eine gleiche Dimensionalität der Eingangsvektoren erforderlich. Als letzter Schritt werden somit intern alle Bilder auf eine einheitliche Größe angepasst. Die Skalierung geschieht ohne Beachtung der Seitenverhältnisse. Dies hat keine Auswirkung auf die Ergebnisse, da dieser Vorgang auf alle Bilder angewandt wird. Die Frage des Seitenverhältnisses und dessen Robustheit bleibt ausserdem die Aufgabe des Trackings.

Geschieht das Lokalisieren der Gesichter entkoppelt vom System, so ist die Normierung der gelieferten Bildausschnitte (Online-Erkennung) oder der Bilder aus der Datenbank (Initialisierung, Offline-Erkennung) ein fester Bestandteil des Erkenners. Die Vorteile liegen auf der Hand:

- Die jedes Mal notwendige Implementierung seitens des Nutzers entfällt
- Normierung ist *einheitlich* und an einer Stelle
- Änderungen wirken sich auf alle Vorgänge des System aus

3.3.1 Angleichen des Histogramms

Seien X und Y zwei n-dimensionale Vektoren, welche jeweils ein Bild mit Helligkeitswerten x_i $(0 \leq i < n)$ aus $[0, 255] \subset N_0$ repräsentieren, dabei sei X das Ausgangsbild und Y das Ergebnis. Sei $H \in N^{256}$ das zu X zugehörige Histogramm so, dass für $0 \leq k \leq 255$

$$h_k = \sum_{i=0}^{n-1} \delta(x_i, k) \tag{3.1}$$

wobei $\delta \in N \times N$ das Kronecker-Symbol ist. Anhand des Histogramms wird das Ausgangsbild Y so berechnet:

$$y_i = \frac{255}{n} \sum_{k=0}^{x_i} h_k \tag{3.2}$$

[2]Natürlich kann der schon entstandene Informationsverlust nicht kompensiert werden.

Man erhält ein Bild mit gleichmäßig verteilter Helligkeit – diese steht im direkten Zusammenhang mit der Anzahl der Pixel. Lagen 50% aller Pixel im Bereich von $[0, 80]$ und wiesen somit höhere Dichte im Histogramm auf, so werden diese auf $[0, 127]$ gestreckt[3].

Alle Schritte der Vorverarbeitung sind in der Abbildung 3.1 für besseres visuelles Verständnis dargestellt.

3.4 Initialisierung und Klassifikation

Die Initialisierung des System erfolgt mit den Bildern aus den Referenzmengen der Datenbank. Sie werden alle für die Berechnung von Eigenfaces verwendet. Die zugehörigen Eigenwerte deuten auf die Wichtigkeit des jeweiligen Eigenface. Wir entscheiden uns für die besten 98%[4], d.h. alle Eigenfaces mit Eigenwerten kleiner als $0,02 * \lambda_{max}$, wobei λ_{max} der höchste Eigenwert sei, tragen wenig zur Erkennung bei und werden nicht berücksichtigt.

Wie in Kapitel 2.1 beschrieben, benutzen wir die errechneten Eigenfaces für Ermittlung der Gewichte, bzw. für Projektion des Bildes in den Gesichtsraum. Der dadurch erhaltene Vektor (Punkt im Gesichtsraum) beschreibt eine eigenständige Klasse des zugehörigen Referenzbildes.

Als Alternative stand im Entwicklungsstadium des Systems eine auf den ersten Blick ausreichende personenweise Klassenbildung zur Debatte, also eine Zusammenfassung der Referenzvektoren einer Person beispielsweise durch Mittelwertbildung. Diese Idee wurde aus folgenden Gründen verworfen. Erstens, können *voneinander verschiedene* Referenzen *einer* Person unmöglich nebeneinander im Gesichtsvektorraum liegen. Denn für die Unterschiede sind *verschiedene* Eigenfaces zuständig, also *verschiedene* Dimensionen des Gesichtsraumes. Bildung eines Durchschnitts bedeutet im schlimmsten Fall Verlust relevanter Merkmale oder gar den Verlust der Möglichkeit, zwischen den Personen zu unterscheiden. Zweitens, wenn jedes Referenzbild Zusatzinformationen[5] über die Aufnahme der Person besitzen würde, ginge die Möglichkeit, diese Daten durch Erkennung zu gewinnen, nach der personenweisen Zusammenfassung der Referenzen verloren. Bei Vorhandensein der Daten über Kopfrotation stünde uns nach der Erkennung der Person nicht nur ihre Identität, sondern auch die Blickrichtung im Raum zur Verfügung!

Die Vorgehensweise mit Abbildung jeder Referenz als eigenständige Klasse erlaubt uns, den Einsatz des Systems je nach Bedarf des Nutzers zu modifizieren.

---

[3] nichtlinear, wiederum von der Verteilung abhängig

[4] während der Entwicklung als sinnvoll erwiesener Wert für maximale Erkennungsraten, in der Praxis sind Werte von 90% bis 95% ausreichend

[5] Kopf-/Blickrichtung, Mimik, Licht, Kopfbedeckung und Kleidung, Brille u.ä.

Mit der Wahl der Referenzen entscheidet er, mit welchem Schwerpunkt die Gesichtserkennung erfolgt. Sind es frontale Aufnahmen unter unterschiedlichen Beleuchtungsverhältnissen, wird das System auf ausreichend varianten Daten zum quasi beleuchtungsunabhängigen Erkenner. Sind die Referenzbilder Aufnahmen von Personen mit verschiedenen Kopfrotationen, wird eine rotationsunabhängige Erkennung angestrebt. Durch Vereinigung beider Referenzmengen erhält man ein beleuchtungs- *und* rotationsunabhängiges Gesichtserkennungssystem.

In der vorliegenden Arbeit wurden getrennte Datenbanken für verschiedene Beleuchtungsverhältnisse eingesetzt. Als Referenzen wurden zehn Aufnahmen rotierter Gesichter in jeder Datenbank ausgewählt. Jedes Referenzbild repräsentiert eine eigenständige Klasse.

3.5 Erkennung

Der Erkennungsprozess läuft in seinem ersten Schritt ähnlich der Initialisierung. Das zu erkennende Bild wird zuerst in den Gesichtsraum projiziert, wo es durch die Eigenfaces repräsentiert wird. Anschließend wird die Euklidische Distanz zu allen Klassen berechnet. Aus der Klasse mit dem kürzesten Abstand wird die zugehörige Person abgeleitet.

Das System bietet die Möglichkeit einer Offline-Erkennung, die für die anstehenden Experimente im Falle einer Wiederholung unverzichtbar ist. In diesem Modus ist es möglich, ganze Bildersequenzen einer Person erkennen zu lassen, mit der Angabe der potentiellen Namen (Identifikationen) und ihrer Erkennungsraten, zusammengefasst aus der Einzelerkennung.

An dieser Stelle wird auf das Problem des normierten Ausrichtens (*alignment*) der Gesichter eingegangen, welches sich nicht allein während der Gesichtslokalisierung lösen lässt. Die Erkennung des gelieferten Gesichtes wird nicht nur einmal durchgeführt, sondern das zugehörige Bild wird intern iterativ horizontal und vertikal verschoben. Jedesmal wird das neu ausgerichtete Bild in den Gesichtsraum projiziert und die neuen Abstände werden berechnet. Der minimale Abstand aus allen Durchläufen mit der zugehörigen Referenz bzw. Klasse stellt das Ergebnis dar.

3.6 Schwellwerte

Der im obigen Kapitel vorgestellte Ansatz der Erkennung durch minimale Abstände erwartet eine bereits bekannte Person und kann keine Eindringlinge erkennen. Gebraucht wird noch ein Maß – ein Schwellwert – um entscheiden zu können, wann das Ergebnis bzw. der errechnete Abstand nicht mehr relevant ist. So kann die

Falscherkennung bekannter und die Erkennung unbekannter Personen minimiert werden.

In der Entwurfsphase standen mehrere Optionen zur Verfügung: Ein globaler Schwellwert, Schwellwert für eine Person oder Klassen- bzw. Referenzenschwellwert. Der erste würde wegen der großen Verallgemeinerung kaum die Voraussetzung erfüllen. Die zweite Idee, nur personenbezogene Schwellwerte einzuführen, erschien nicht so präzise und flexibel wie der Klassenschwellwert. Dies wurde später in der Testphase des Systems bestätigt, so fielen die Schwellwerte beispielsweise für frontale Aufnahmen deutlich kleiner aus, als die für extrem rotierte Gesichter. Deren Unterscheidung wäre somit nur auf der Klassenebene möglich.

Der Schwellwert erzeugt im Sinne des Euklidischen Abstandes eine n-dimensionale Kugel um den Klassenvektor. In dieser würden sich alle zu dieser Klasse gehörenden Gesichter befinden. Das ist der nächste Schritt in der Erkennung, nachdem der kürzeste Abstand berechnet wurde: es muss geprüft werden, ob dieser kleiner ist, als der Klassenschwellwert, d.h. ob das zu erkennende Bild im Gesichtsraum innerhalb der Kugel um den Klassenvektor liegt.

Ein weiterer Differenzierungsschritt ist die Definition der Schwellwerte für jede Dimension, also eines n-dimensionalen Quaders. Es ist durchaus vorstellbar, dass sich dieser in bestimmte Richtungen mehr ausdehnen würde. Leider steigt mit diesem Ansatz der Aufwand erheblich, da nach jeder Änderung in der Referenzdatenbank alle berechneten Schwellwerte (vor allem durch Verschiebung der Dimensionen) ungültig wären.

Die Klassenschwellwerte werden durch Verifikation gebildet, dabei sind die Falschakzeptanz (FAR: *false acceptance rate*) und die fälschliche Rückweisung (FRR: *false rejection rate*) zu minimieren. Nach der Initialisierung wird für jede Person der zugehörige Trainingssatz für diesen Vorgang verwendet. In diesem Fall weiß das System a priori, um welche Personen es sich dabei handelt, und vergleicht das mit dem Ergebnis der Erkennung. FAR gibt an, welcher Anteil der Eindringlinge (*impostor*) fälschlicherweise als gegebene Person akzeptiert wurde. FRR ist das Verhältnis zwischen den falsch abgewiesenen und allen authentischen Aufnahmen der gegebenen Person. Beide Werte hängen vom gewählten Schwellwert ab und sind bei dem bestmöglichen Schwellwert gleich (s. Abbildung 3.2) und ergeben die *equal error rate* (EER).

Die Verifikationsaufgabe wird wie folgt vollzogen. Zuerst wird die Erkennung des ersten Bildes aus dem Trainingssatz der ersten Person durchgeführt. In der erkannten Klasse wird der Abstand gespeichert, sowie ob das Ergebnis korrekt oder falsch war. So wird verfahren, bis alle Trainingsdaten zur Erkennung herangezogen wurden. Für jede Klasse ergeben sich zwei Verteilungskurven der Abstände. Die erste Kurve beschreibt die Verteilung der Distanzen der korrekten Erkennung, die zweite die Verteilung der falschen. Der Schnittpunkt, projiziert auf die Abstandsachse ist der Schwellwert, bei dem die EER minimal ist.

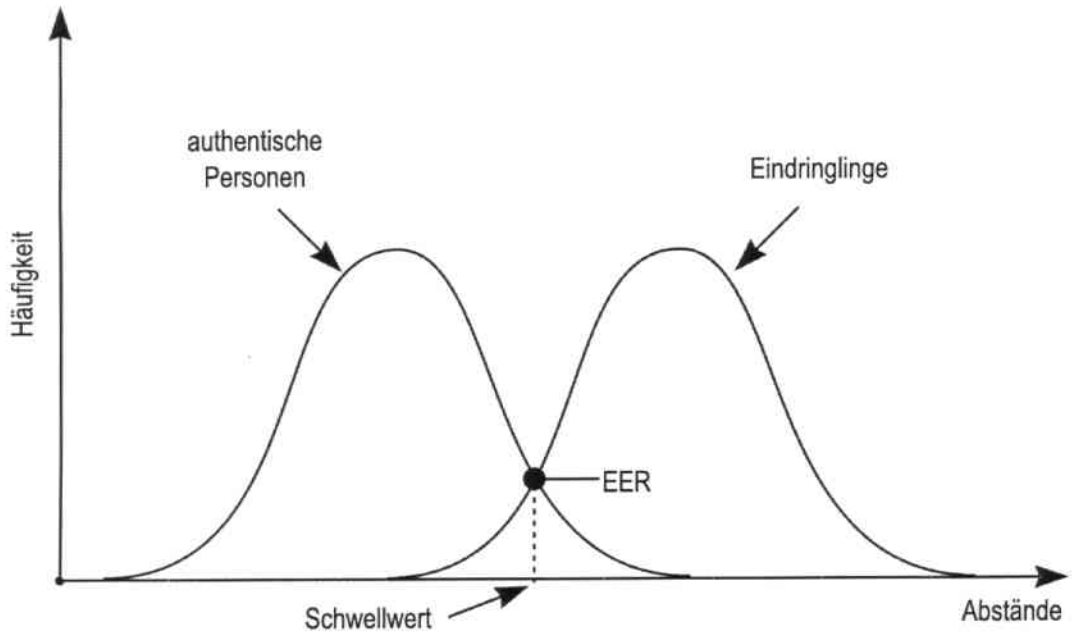


Abbildung 3.2: Ermittlung des Schwellwertes für die beste *EER*

Der errechnete Schwellwert wird für jedes Referenzbild in der Datenbank gespeichert und, wenn erwünscht, bei der Erkennung ausgelesen und verwendet. Nachdem die ID und der Abstand bekannt sind, wird dieser mit dem Schwellwert der jeweiligen Klasse verglichen und im Falle *Abstand* > *Schwellwert* wird als Ergebnis die *unknown*-Klasse zurückgegeben.

3.7 Referenzenauswahl

Der Vorgang der Referenzenauswahl spielt eine große Rolle im gesamten Erkennungsprozess. Auch ein System mit besten Algorithmen führt die Erkennung nur so gut durch, wie die zur Verfügung stehenden Referenzen es erlauben. Eine für den gegebenen Ansatz optimale Wahl zu treffen, ist eine schwierige Aufgabe. Besonders wenn es sich um die Auswahl einiger weniger Bilder handelt, die mehrere Hundert bestmöglich repräsentieren sollen. Bei geringem Wissen über den Erkennungsalgorithmus kann diese Aufgabe den Nutzer überfordern. Bei einer Entscheidung über mehrere ähnlich aussehende Bilder würde er nach Gefühl handeln, und nicht das beste Bild im mathematischen Sinne der Erkennungsmethode wählen.

Mit einem automatisierten Prozess kann es gelingen, den Zufallsfaktor zu minimieren und ein festes Entscheidungskriterium zu gewinnen. Hinzu kommt der Vorteil der Zeitersparnis und Entlastung des Nutzers.

In dieser Arbeit wurden, abgesehen von der gegebenen manuellen Auswahl, zwei automatische Ansätze implementiert. Der erste ist schnell und unkompliziert: Per Zufall wird die benötigte Anzahl der Referenzbilder aus der Trainingsmenge ausgewählt und in die Datenbank eingefügt.

Die zweite Methode ist etwas aufwendiger, basiert auf dem k-Means-Algorithmus und teilt den Trainingssatz in die gegebene Anzahl von Clustern auf. Um das Clustering durchführen zu können, wird für jedes Bild aus dem Trainingssatz seine Darstellung im Gesichtsraum benötigt. Dafür wird das bestehende System nicht wie üblich mit Referenzbildern initialisiert, sondern mit dem Trainingssatz der Person, für die die neuen Referenzen gesucht werden. Anschließend kann der k-Means-Ansatz, die ähnlichen Gesichter, d.h. die nahe beieinander liegenden Punkte im Gesichtsraum, in einem Cluster zusammenfassen. Das zur Clustermitte am nächsten liegende Bild wird die neue Referenz.

Das Ergebnis dieser Methode wird in der Abbildung 4.4 in Kapitel 4.3 veranschaulicht.

4 Experimentelle Ergebnisse

In diesem Kapitel werden die experimentellen Ergebnisse, die mit dem implementierten Ansatz erzielt wurden, besprochen. Zu jedem durchgeführten Versuch wird eine Übersicht der Bedingungen, wie Trainings- und Evaluationssätze, gegeben. Anschließend werden die Ergebnisse ausgewertet und mögliche Verbesserungen vorgeschlagen.

Eigenfaces ist einer der wichtigsten Muster- und Gesichtserkennungsansätze. In vielen Arbeiten, die sich mit diesem beschäftigen, werden meistens nur ausgewählte Experimente auf gut und häufig manuell vorbereiteten Daten durchgeführt. Diese im *best case* gewonnenen, recht optimistischen Ergebnisse finden selten so gute Bestätigung in der Praxis, besonders unter *unkontrollierten* Bedingungen. Das Ziel dieser Arbeit war, bestmöglich die potentiellen Einsatzbedingungen zu simulieren, mit manuellem Eingriff nur für Bedienung und Initialisierung des Systems. Als ein Anwendungsszenario wird eine mobile Roboterplattform mit der Verwendung einer Stereokamera ins Auge gefasst.

4.1 Beschreibung und Zusammensetzung der Datensätze

Zur Versuchsdurchführung werden drei mit einer Stereokamera aufgenommenen Videosequenzen verwendet. Die ersten zwei sind sogenannte „Portrait-View" Aufnahmen mit unterschiedlicher Beleuchtung. Die Tageslicht-Version wird im Folgenden als *P1*-Datensatz bezeichnet, die Aufnahme unter Kunstlicht-Bedingungen als *P2*. In beiden Fällen wurden mehrere Personen aus kurzem Abstand zur Kamera aufgenommen. Die sich dadurch ergebene Größe des Gesichtsausschnittes beträgt etwa 80×100 Pixel. Der dritte Datensatz beruht auf einer „Robo-View"[1] Videosequenz, in der die Personen aus einer größeren Entfernung aufgenommen wurden. Dabei schrumpfte die Gesichtsgröße auf etwa ein Viertel von Portrait-View und misst etwa 40×50 Pixel. Dieser Datensatz wird im Folgenden R genannt. In der Abbildung 4.1 ist eine Gegenüberstellung der drei Datensätze zu finden.

[1] Die Stereokamera wurde auf einer Roboterplattform montiert, um den Einsatz bestmöglich zu simulieren.

Datensatz	P1		P2		R	
Name/Tracking	A/B	C	A/B	C	A/B	C
person1	449	453	327	387		
person2	469	447				
person3	217	228	228	244		
person4	601	601	415	416	826	858
person5					866	900
person6	385	432	399	403		
person7					446	453
person8	320	346	342	342		
person9	303	372	363	371	599	755
person10					688	719
person11			176	196		
person12	522	523				
person13	429	499	417	489		
person14	341	356	316	347		
person15	262	269	373	466		
person16	419	370	278	302	739	820
Summe	4.717	4.896	3.634	3.963	4.164	4.505

Tabelle 4.1: Bilderanzahl in den Datensätzen für jede Person Die Spalten *A/B* und *C* sind die Trackingarten (s. Kapitel 4.2). A- und B-Tracking besitzen die gleiche Anzahl der Bilder.

Tabelle 4.1 zeigt die zu jedem Datensatz zugehörigen Testpersonen mit der entsprechenden Anzahl der Bilder aus der jeweiligen Aufnahme. Diese Bilder wurden im Zufallsverfahren in zwei Datensätze aufgeteilt: 25% als Trainings- und 75% als Testmenge. Insgesamt wurden über 38.000 Bilder aus den Videosequenzen extrahiert.

Bei den Videosequenzen handelt es sich um Aufnahmen von verschiedenen Personen, die nach ihrem Belieben vor der Kamera den Kopf bewegten, um eine rotationsunabhängige Erkennung zu ermöglichen.

Alle in den folgenden Experimenten genannten Erkennungsraten beziehen sich auf die Offline-Erkennung aller Testbilder einer Videosequenz und stellen das Verhältnis zwischen den *korrekt* erkannten und allen Bildern aus der Testmenge dar. Alle Experimente werden mit *zehn* Referenzen pro Person durchgeführt.

Obere Reihe: der erste „Portrait-View"-Datensatz
Mittlere Reihe: der zweite „Portrait-View"-Datensatz
Untere Reihe: Bilder des „Robo-View"-Datensatzes

Abbildung 4.1: Beispielaufnahmen aus den Datensätzen

4.2 Anfangsprobleme und Vorbereitungen

Die Aufgabe der Lokalisierung der Gesichter stellte sich als ausserordentlich schwierig heraus. Große Robustheit der Ansätze und Kontinuität der Ergebnisse waren von großer Wichtigkeit. Aus diesem Grunde kamen zwei Trackingmethoden (s. Kapitel 2.2) in diesem System zum Einsatz.

Zuerst wurde der in Kapitel 2.2.1 beschriebene 3D-Face-Tracker für das Lokalisieren der Gesichter verwendet. Mit dem Vorteil durch die Tiefeninformation der Stereobilder lieferte dieser sehr gute Ergebnisse auch bei extrem rotierten Köpfen und Problemstellen im Hintergrund. Die aufgetretenen Probleme können hier in zwei Gruppen zusammengefasst werden: Innerhalb einer Aufnahme und zwischen Aufnahmen einer Person.

Innerhalb einer Aufnahme variierten die gelieferten Ausschnitte sehr stark von Bild zu Bild. Eine Erklärung dafür findet man hauptsächlich in der Problematik der Hautfarbe:

- Bei unbedecktem Hals zählt dieser manchmal zu der Kopfregion.
- Helle Haare besitzen einen Hautfarbanteil, was den Tracker irritiert.

- Bei Lichtquellen unterschiedlicher Farbtemperatur entstehen durch Kopfrotation verschiedene Farbtonwerte – nicht erfassbar durch das Hautfarbmodell.

Diese Probleme wirken sich negativ auf die Erkennung einer Person in einer Bildfolge aus: Trotz der großen Ähnlichkeit zum vorherigen Bild kann der Erkenner durch Verschiebung des Ausschnittes und Änderung dessen Größe ein anderes Ergebnis liefern.

Es ließ sich allerdings Folgendes beobachten. Die von diesem Tracker gelieferten Bildausschnitte zeigten große Abweichungen bei der unteren Kopfregionskante, seltener in der Breite und sehr selten bei der oberen Kante. Zum einen ist das ein Hinweis auf den Vorteil durch Tiefeninformation (gute Ergebnisse in der Silhouette), zum anderen ist dieses Verhalten auf das Problem mit dem Hals zurückzuführen. Eine Definition eines festen Seitenverhältnisses für die Kopfregion würde diese Störung verringern.

Die dauerhafte Störung liegt zwischen verschiedenen Aufnahmen. So können sich die Ausschnitte aus *P1* und *P2* für eine und dieselbe Person erheblich unterscheiden. Ursachen für dauerhafte und konstante Änderungen der Gesichtsbox von Aufnahme zu Aufnahme sind:

- Veränderte Lichtverhältnisse (kaum Farbinformationen in den dunklen Schatten in einer Gesichtshälfte)
- Andere Kleidungsstücke (Rollkragenpullover statt aufgeknöpftes Hemd)
- Unterschiedlich definierte Kameraeinstellungen (z.B. Weißabgleich und Schärfe)

Um später die Rolle der Vorverarbeitung in den Ergebnissen zu sehen, wird Tracking mit 3D-Face-Tracker ohne weitere Anpassung als $\boldsymbol{A}$ bezeichnet, mit dem festen Seitenverhältnis als $\boldsymbol{B}$.

Das Gesicht ließ sich mit diesem Tracker nicht ausreichend genau und stabil ausschneiden. Um das Ausrichtungsproblem zu lösen, wären die Datensätze *P1* und *P2* für den Einsatz der Lokalisierung der Gesichtspartien, wie Augen, Nase, Mund, geeignet. Jedoch würde dies kaum bei den kleineren Gesichtern aus der „Robo-View"-Aufnahme funktionieren.

Deswegen fand ein weiterer Ansatz aus Kapitel 2.2.2 in diesem System Anwendung und wird fortan durch $\boldsymbol{C}$ referenziert (s. Abbildung 4.2 für einen Vergleich). Durch den Vorteil des Ellipsenmodells und durch die Berücksichtigung des letzten Ergebnisses ergeben sich nur geringe Abweichungen zwischen den benachbarten Bildern. Allerdings wird der Algorithmus bei raschen Kopfbewegungen gestört und kann die Bewegung erst über mehrere nachfolgende Bilder ausgleichen. Im schlimmsten Fall wird das verfolgte Gesicht verloren und die Ellipse rutscht auf ein anderes Objekt oder einen anderen Körperteil (z.B. Hals) ab.

Obere Reihe: Tracking *A* mit unkorrigertem Ausschnitt vom 3D-Face-Tracker
Mittlere Reihe: Tracking *B* mit festem Seitenverhältnis
Untere Reihe: Tracking *C* des Ansatzes „Elliptical Head Tracking"

Abbildung 4.2: Vergleich der Trackingmethoden

Alle hier genannten Probleme des Trackings werden sich in den Ergebnissen der Experimente wiederfinden und werden an entsprechenden Stellen diskutiert.

4.3 Referenzenauswahl

Im ersten Versuch wird der Einfluss der Referenzenauswahl untersucht. Die Wahl der charakteristischen Gesichter einer Person ist der wichtigste Schritt für die Erkennung und hat somit den größten Einfluss auf die Erkennungsraten. Untersucht werden jeweils manuelle, zufällige und k-Means basierte Auswahl auf *P1*. Der k-Means-Ansatz wird mit $k = 10$ initialisiert, um zehn Repräsentanten des Datensatzes zu finden.

Training:	P1
Referenzen:	manuell, zufällig, k-Means
Tracking:	C
Schwellwerte:	nein
Auswertung:	P1

Wie in der Abbildung 4.3 und der Tabelle 4.2 zu sehen ist, können wir von einem Erfolg für den k-Means basierten Ansatz für die Referenzenauswahl sprechen. Er erreicht im Schnitt die Quote der per Hand ausgewählten Daten.

Die guten Ergebnisse bei einigen Personen (p4, p6, p12 – größere Datensätze, vgl. Tabelle 4.1) weisen drauf hin, dass es auf variationsreiche Daten für die k-Means

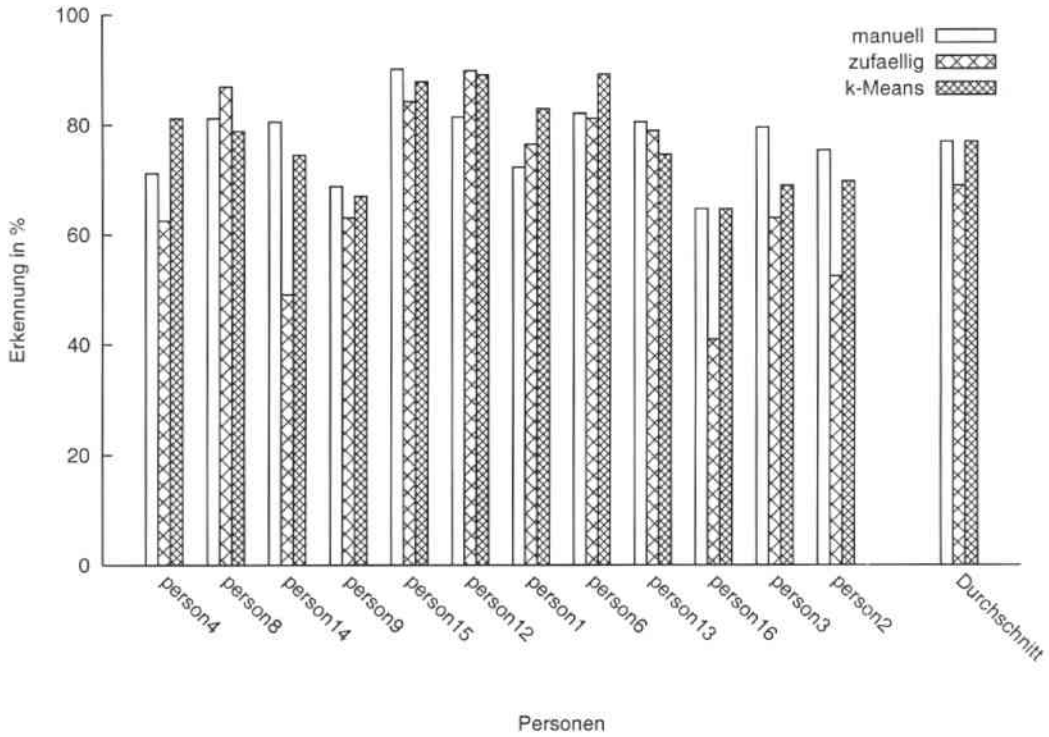


Abbildung 4.3: Vergleich der Referenzierungsmethoden mit Training und Auswertung auf *P1*, Tracking *C*

Name	*manuell*	*zufällig*	*k-Means*
person4	71,2	62,5	81,2
person8	81,2	86,9	78,9
person14	80,5	49,1	74,5
person9	68,8	63,1	67,0
person15	90,1	84,2	87,8
person12	81,4	89,8	89,1
person1	72,4	76,5	82,9
person6	82,1	81,2	89,2
person13	80,5	78,9	74,7
person16	64,8	41,0	64,8
person3	79,5	63,2	69,0
person2	75,4	52,5	69,8
Durchschnitt	77,3	69,1	77,4

Tabelle 4.2: Erkennungsraten der Referenzierungsmethoden, in %.

Abbildung 4.4: Die Auswahl des k-Means basierten Referenzierungsverfahrens

basierte Auswahl ankommt. Durch eine ungünstige Trennung der aufgenommenen Daten und durch wenig gestreute Daten kann es vorkommen, dass dieser Ansatz im Einzelnen schlechtere Ergebnisse liefert, als die Erkennung auf den manuell ausgewählten Referenzen.

Mit den im vorhergehenden Kapitel beschriebenen Problemen des Trackings entscheidet sich die k-Means-Methode für Bilder, die für einen Menschen ohne Rücksicht auf Position und Skalierung ähnlich erscheinen, aber im Sinne von Eigenfaces kein Optimum darstellen. Dieser Effekt war deutlich zu beobachten, bevor der Erkenner mit der Funktion für den Ausgleich der Bildausrichtung (s. Kapitel 3.5) ausgerüstet wurde.

Höchstinteressant ist der Punkt, dass man mit Hilfe von k-Means eine Erklärung für die Funktionsweise von Eigenfaces bekommt: Schaut man sich die ausgewählten Bilder in der Abbildung 4.4 an, ist es sofort nachvollziehbar, was für eine richtige Wahl der Referenzen entscheidend ist. Unter den vorgeschlagenen Bildern findet man in diesem Fall sofort eine Sammlung der Kopfrotationen (das Merkmal des Trainingssatzes!) sowie gelegentlich Gesichter mit ähnlichen Richtungen aber unterschiedlicher Mimik vor.

Das gute Abschneiden von k-Means-Referenzen rechtfertigt den weiteren Einsatz dieses Referenzierungsverfahrens in allen folgenden Experimenten.

4.4 Veränderung der Gesichtsgröße

Das System arbeitet intern mit der festen Bildgröße von 50×60 Pixeln. Die Bilder werden immer auf diese Größe skaliert, unabhängig davon, wie groß sie in der Datenbank vorliegen oder zur Erkennung übergeben werden. Interessant

Name	100% (50 × 60)	50% (25 × 30)	30% (15 × 18)	20% (10 × 12)
person4	77.2	73.4	59.9	55.8
person8	92.6	88.7	89.1	73.9
person14	79.3	77.0	70.5	65.1
person9	79.6	80.3	72.8	68.5
person15	64.9	60.3	43.7	24.0
person1	89.3	88.3	83.1	74.1
person6	76.9	68.7	55.8	52.5
person11	81.0	76.2	72.8	50.3
person13	83.4	82.6	77.9	70.0
person16	56.4	55.1	48.9	39.2
person3	66.1	59.6	55.7	49.7
Durchschnitt	77.0	73,6	66,4	56,7

Tabelle 4.3: Erkennungsraten in % bei verschiedenen Gesichtsgrößen. Training und Auswertung auf *P2*, Tracking *C*.

dabei sind die Auswirkungen auf die Ergebnisse, wenn intern mit kleineren Bildern gearbeitet wird.

Training: P2
Referenzen: k-Means
Tracking: C
Schwellwerte: nein
Auswertung: P2

Schauen wir uns die Ergebnisse in der Tabelle 4.3 sowie ihre visuelle Darstellung in der Abbildung 4.5.

Deutlich ist hier zu sehen, dass sich die Ergebnisse mit der Bildgröße von 15×18 Pixeln rapide verschlechtern. Ein vollkommen erwartetes Verhalten: mit kleinerer Pixelmenge stehen auch weniger Informationen für die Erkennung zur Verfügung.

Andererseits verträgt das System die Änderung von 50 × 60 auf 25 × 30 Pixel sehr souverän. Hier werden nur wenige Abstriche in der Erkennungsrate gemacht und diese gelten nur für einige Personen. Also kann diese Größe bei großen Datenbanken und für bessere Geschwindigkeit auf den kleineren Wert gesetzt werden. Es wäre aber zu früh, von einem Sättigungsverhalten zu sprechen. Denn trotz ihrer hohen Auflösung besitzen die Eingangsbilder deutlich weniger Informationen, bedingt durch begrenzte Schärfentiefe, Bayer-Interpolation, Qualität des Kameraobjektives. Durch diese „Unschärfe“ können die Bilder bis zu einem gewissen Punkt ohne großen Informationsverlust verkleinert werden. Genau aus diesem Grund kann man *nicht* darauf schließen, dass bei Verkleinerung der Eingangsbilder (nicht der Größe der internen Verarbeitung!) dieser Effekt weiterhin zu beobachten sein wird.

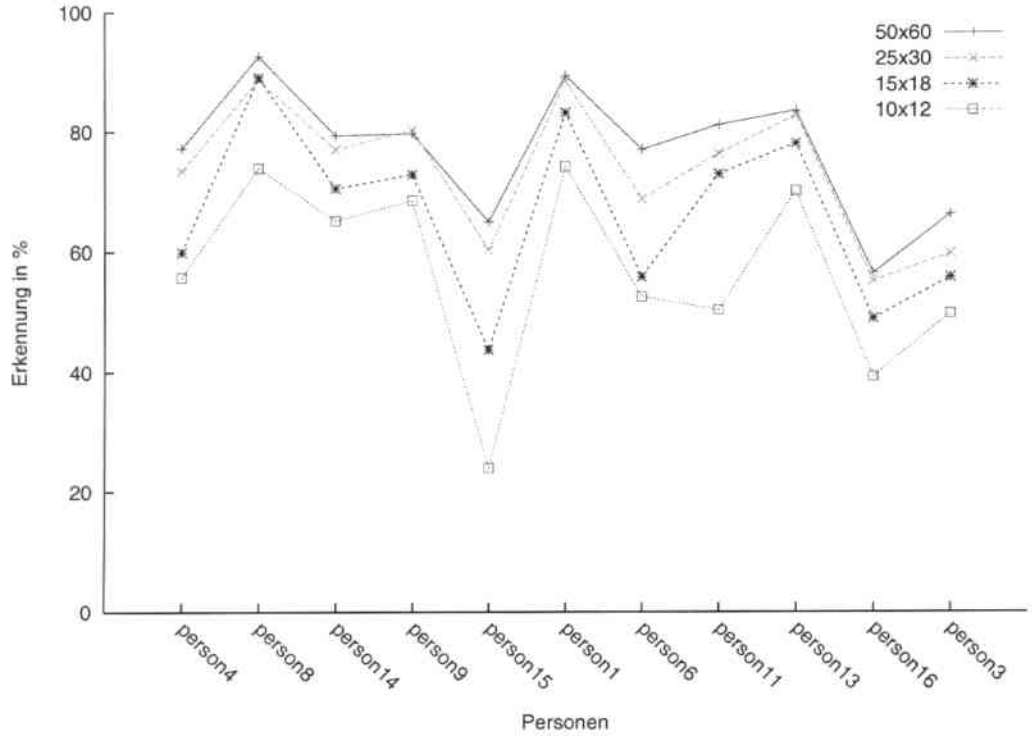


Abbildung 4.5: Erkennungsergebnisse bei verschiedenen Gesichtsgrößen

4.5 Auswirkungen der Vorverarbeitung

Mit diesem Versuch wird die real messbare Verbesserung der im Kapitel 3.3 beschriebenen Vorverarbeitungsschritte untersucht.

4.5.1 Trackingmethoden

Zuerst untersuchen wir das Verhalten der drei Methoden der Gesichtslokalisierung. Dafür wird der zweite „Portrait-View"-Datensatz für das Training sowie für die Evaluation verwendet.

Training:	P2
Referenzen:	k-Means
Tracking:	A, B, C
Schwellwerte:	nein
Auswertung:	P2

Die Ergebnisse sind in der Abbildung 4.6 dargestellt sowie genau in der Tabelle 4.4 zu finden.

	p4	p8	p14	p9	p15	p1	p6	p11	p13	p16	p3	Schnitt
A	84,9	98,4	90,3	99,6	88,9	77,1	90,7	86,4	83,4	83,7	88,3	88,3
B	78,9	98,1	83,5	97,4	80,0	85,3	75,7	87,1	87,2	61,7	81,3	83,3
C	76,9	93,4	79,7	79,9	65,4	89,0	77,2	79,6	83,9	56,8	65,0	77,0

Tabelle 4.4: Erkennungsraten der verschiedenen Trackingmethoden, in %. Training und Auswertung auf *P2*.

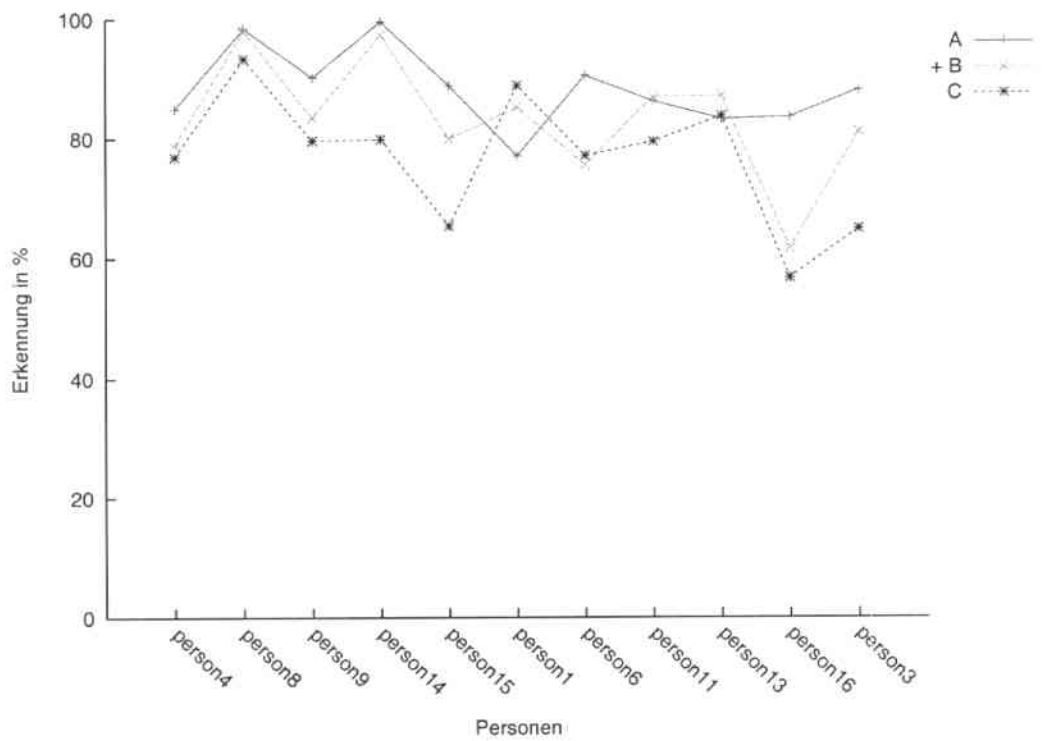


Abbildung 4.6: Erkennungsergebnisse der verschiedenen Trackingmethoden

Alle Methoden liefern sehr gute Ergebnisse und haben bei einigen Personen Vor- und Nachteile. Wider Erwarten fallen die Ergebnisse mit dem *Elliptical-Head-Tracking*-Ansatz am schlechtesten aus, gegenüber den anderen beiden Tracking-Methoden. Das gute Abschneiden der letzteren ist aber relativ einfach zu erklären.

Da keine der Vorverarbeitungsmethoden weder positions- noch skalierungsinvariant sind, wird diese Eigenschaft zum Merkmal selbst. Schneidet der Tracker die Gesichter von Person X im Vergleich zu Person Y mit 120% (bezogen auf die Kopfgröße) kleiner aus, so reicht es, für Eigenfaces nur die „Skalierungseigenschaft" zu prüfen, um diese Personen auseinanderzuhalten. Genauso funktioniert es mit der Position: wird die Gesichtsregion bei X andauernd von der Stirn bis zum Kinn und bei Y von den Augenbrauen bis zum Hals ermittelt, wird diese Verschiebung des

Gesichts zum bedeutendsten Feature für diese Person. Zum Verhängnis wird es, wenn die nächste Testaufnahme die Eigenschaft der trainierten Aufnahme nicht mehr aufweist: die starke Gewichtung der Position und Skalierung verfälscht das Ergebnis deutlich.

Aus diesen Gründen fällt das Ergebnis der Testperson „p16“ mit so einem großen Vorsprung für das Tracking *A* aus. Nach Durchsicht der Datensätze kann bestätigt werden, dass die Bilder der Trackingart *A* (kein festes Seitenverhältnis) eine eindeutig resistente Eigenschaft der Position und Skalierung haben, gegenüber allen anderen Testpersonen im Vergleich zu *B* und *C*.

Auch das schlechtere Resultat des Trackings *C*, weist darauf hin, dass größere Ähnlichkeit unter seinen Bildern gibt, d.h. die Ausrichtung der Bilder ist mit diesem Ansatz deutlich besser gelungen, als bei *A* und *B*.

Als Fazit bestehen aufgrund der Ausrichtung und Skalierung der Bilder noch Zweifel, ob sich diese Ergebnisse bestätigen werden. Die Vorteile des Trackings können erst durch Auswerten von Daten aus einer anderen Aufnahme überprüft werden. Darauf wird im Versuch 4.6 genauer eingegangen.

4.5.2 Bildnormierung

Dieser Vorverarbeitungsschritt wurde eingeführt, um Beleuchtungsprobleme soweit wie möglich zu reduzieren. Somit ist eine Auswertung auf den Testdaten aus einer anderen Aufnahme als die der Referenzen erforderlich. In der Tat wäre eine Bildnormierung für die gleiche Referenzen-/Testmenge mitnichten eine Verbesserung. Denn die Daten aus den beiden Sätzen hätten ihre größte gemeinsame Eigenschaft in der Helligkeitsverteilung – je nach Aufnahme ein vollkommen ausreichendes Merkmal für fast 100%-ige Erkennung.

In diesem Experiment wird die Datenbank mit *P1*-Referenzen initialisiert, und Erkennung auf dem Testsatz aus *P2* ausgeführt.

Training:	P1
Referenzen:	k-Means
Tracking:	C
Schwellwerte:	nein
Auswertung:	P2

Ein auf den ersten Blick ernüchterndes Ergebnis liefert die Tabelle 4.5. Besonders das Ergebnis der ersten Spalte scheint keiner Regelmäßigkeit zu unterliegen. Dies ist allerdings folgendermaßen erklärbar: ohne jegliche Normierung wird die Helligkeit der Bilder zum entscheidenden Merkmal selbst, die Gesichtserkennung wäre in so einem Fall eine Art Zuordnung der Bilder gleicher Helligkeit. Bei der Testperson „p8“ ist sofort zu sehen, dass die Helligkeit in den beiden Aufnahmen ziemlich gleich sein soll. Der absolute Gegensatz herrscht bei allen Personen

Name	*ohne*	*Histogramm*	*+ Gauß* 3×3	*+ Gauß* 5×5
person4	13,1	0,3	1,0	2,2
person8	76,3	70,4	72,4	74,3
person14	0,0	10,3	10,3	10,7
person9	0,0	7,5	6,1	5,7
person15	0,0	8,6	8,9	10,0
person1	0,0	47,9	52,1	45,2
person6	0,0	12,5	13,2	10,9
person13	0,0	18,5	19,4	18,0
person16	13,2	11,5	11,5	11,9
person3	10,9	8,7	4,9	6,6
Durchschnitt	11,4	19,6	20,0	19,6

Tabelle 4.5: Bildvorverarbeitung, Erkennungsraten in %. Training auf *P1*, Auswertung auf *P2*, Tracking *C*.

mit 0,0% korrekter Erkennung. Hier liegen die Aufnahmen in der Helligkeit so weit voneinander, dass sie eher anderen Person zugeordnet werden, als den „ähnlichen" Referenzgesichtern.

Dass diese Ähnlichkeit überhaupt existiert, zeigt uns sofort die zweite Spalte. Diese Ergebnisse werden noch mit dem Gaußfilter mit 3×3 verbessert. Der nächste Schritt 5×5 glättet etwas mehr und minimiert nur geringfügig die Erkennungsrate im Durchschnitt. Nichtsdestotrotz wird er im System eingesetzt, da er bei Untersuchungen mit anderen Datensätzen deutliche Verbesserung im Vergleich zu 3×3 zeigte.

Insgesamt sehen wir in der Praxis, was schon vorher aus der Theorie bekannt war: Beleuchtung stellt ein großes Problem für Eigenfaces dar. Dieses Problem ist nicht mit einfachen Mitteln, wie Bildnormierung, zu lösen, und bedarf komplexerer Ansätze oder anderer Trainingsdaten. Dies wird aber genauer im folgenden Kapitel untersucht. Dennoch ist die Bildvorverarbeitung samt Tracking und Bildnormierung entscheidend für die korrekte Funktionsweise des gesamten Erkennungssystems.

4.6 Beleuchtungsveränderung

In diesem Versuch untersuchen wir den Einfluss der Beleuchtung auf die Erkennungsergebnisse. Es stehen zwei Datensätze mit unterschiedlichen Beleuchtungsverhältnissen zur Verfügung. Jeder dieser Datensätze wird einmal als Ausgang für das Training verwendet und danach als Evaluationssatz.

Name	Training *P1*, Test *P2*	Training *P2*, Test *P1*
person4	47,8	6,2
person8	34,6	50,0
person14	54,9	52,7
person9	64,5	70,2
person15	27,5	6,1
person1	38,4	29,1
person6	12,7	4,5
person13	21,4	39,1
person16	19,6	13,7
person3	17,0	15,3
Durchschnitt	33,8	28,7

Tabelle 4.6: Beleuchtungsverhältnisse, Erkennungsraten in %

Training:	P1, dann P2
Referenzen:	k-Means
Tracking:	B
Schwellwerte:	nein
Auswertung:	P2, dann P1

Die Resultate dieses Versuches sind in der Tabelle 4.6 zu finden.

Sofort fällt das allgemein schlechtere Ergebnis gegenüber Erkennung bei gleichgebliebener Beleuchtung auf. Das bekannte Problem von Eigenfaces wird in diesem Versuch noch einmal bestätigt. Im Einzelnen sieht man starke Abweichungen unter den Personen. Dies ist zum größten Teil auf das andere Verhalten der Gesichtstracker bei geänderter Beleuchtung zurückzuführen, die bei manchen Personen teilweise komplett unbrauchbare Ergebnisse liefern. Eine große Rolle spielt dabei natürlich auch die Referenzenauswahl, wie gut sie aus dem Trainingssatz auch die Bilder aus dem Evaluationssatz beschreibt. Dadurch ergeben sich die Differenzen beim Tausch Trainings-/Testsatz ($P1 \rightarrow P2$, $P2 \rightarrow P1$). Hatten die beiden Videosequenzen einer Person etwa die gleiche Varianz in den Daten, so fielen auch die Referenzen gut geeignet für den alternativen Datensatz aus.

Bei Personen „p8“, „p14“ und „p9“ sehen wir positive Beispiele des stabilen Trackings und somit für diese Bedingungen sehr gute Ergebnisse. Allein durch wenige Vorverarbeitungsschritte konnte dies erreicht werden.

Wie im Versuch 4.5.1 hingewiesen, wollen wir noch einmal in diesem Experiment die Trackingmethoden untersuchen. Dazu vergleichen wir die Ergebnisse der letzten Spalte in der Tabelle 4.5 (Tracking C) mit den Ergebnissen aus der ersten Spalte der Tabelle dieses Experiments (Tracking B). Hier sehen wir einen deutlichen Gewinn für die letzte Methode. Die Erklärung für dieses unerwartete Verhalten wurde in Kapitel 4.5.1 besprochen. Die Beleuchtungsveränderung beeinflusst das Ergebnis der ähnlich ausgeschnittenen Gesichter des Tracking C am meisten,

	Tracking A				Tracking B				Tracking C			
	W	X	Y	Z	W	X	Y	Z	W	X	Y	Z
p5	99,1	91,5	8,2	99,7	97,7	92,9	6,0	98,9	99,0	98,5	1,5	100,0
p9	94,7	88,9	11,1	100,0	92,2	88,4	11,3	99,7	91,0	88,5	11,3	99,8
p7	92,8	92,8	7,2	100,0	98,2	92,8	6,3	99,1	93,8	90,0	5,9	95,6
p10	84,3	83,3	13,6	96,4	80,2	77,5	20,9	98,0	84,4	81,7	13,2	94,0
p4	86,8	84,2	12,3	96,0	87,3	82,9	11,6	93,8	84,5	80,3	16,8	96,5
p16	95,0	89,2	10,3	99,4	91,4	80,9	17,5	98,0	81,8	71,4	25,9	96,3
⌀	**92,1**	88,3	10,4	**98,6**	**91,2**	85,9	12,3	**97,9**	**89,1**	85,1	12,4	**97,0**

W Erkennungsrate **ohne** Schwellwerte
X Erkennungsrate **mit** Schwellwerten
Y Anteil der als unbekannt eingestuften Gesichter (die *unknown*-Klasse)
Z Korrektheit von **X** ($Z = \frac{X}{100 - Y} * 100\%$)

Tabelle 4.7: Einsatz der Klassenschwellwerte, Ergebnisse in %. Training und Auswertung auf R

da sie fast „nur" Gesichtsmerkmale aufweisen und kaum eine Skalierungs- oder Positionseigenschaft besitzen. In den weiteren Versuchen wird man noch sehen, dass sich das Tracking B minimal durchsetzt und die besten Ergebnisse liefert.

Nicht nur mit Problemen des Eigenface-Ansatzes mussten wir uns in diesem Versuch auseinandersetzen. Zuerst sind die Schwierigkeiten des Trackings und der Referenzenwahl zu bewältigen. Aber erst eine beleuchtungsunabhängige Enkodierung der Bilddaten wird eine weitere Anwendung von Eigenfaces vorantreiben und ihr Potenzial ausnutzen.

4.7 Schwellwerte

In diesem Versuch vergleichen wir die Ergebnisse mit und ohne Berücksichtigung der Klassenschwellwerte (siehe Kapitel 3.6). Dafür wird der bis jetzt noch nicht eingesetzte „Robo-View"-Datensatz verwendet. Gleichzeitig schauen wir uns an, wie die Erkennung auf kleineren Bildern funktioniert.

Training:	R
Referenzen:	k-Means
Tracking:	A, B, C
Schwellwerte:	mit/ohne
Auswertung:	R

Dieses Experiment verlief mit hervorragenden Ergebnissen, wie in der Tabelle 4.7 zusammengefasst. Zum einen ist das gute Resultat auf eine deutlich größere Menge der Daten zurückzuführen (vgl. Tabelle 4.1), zum anderen standen in dieser

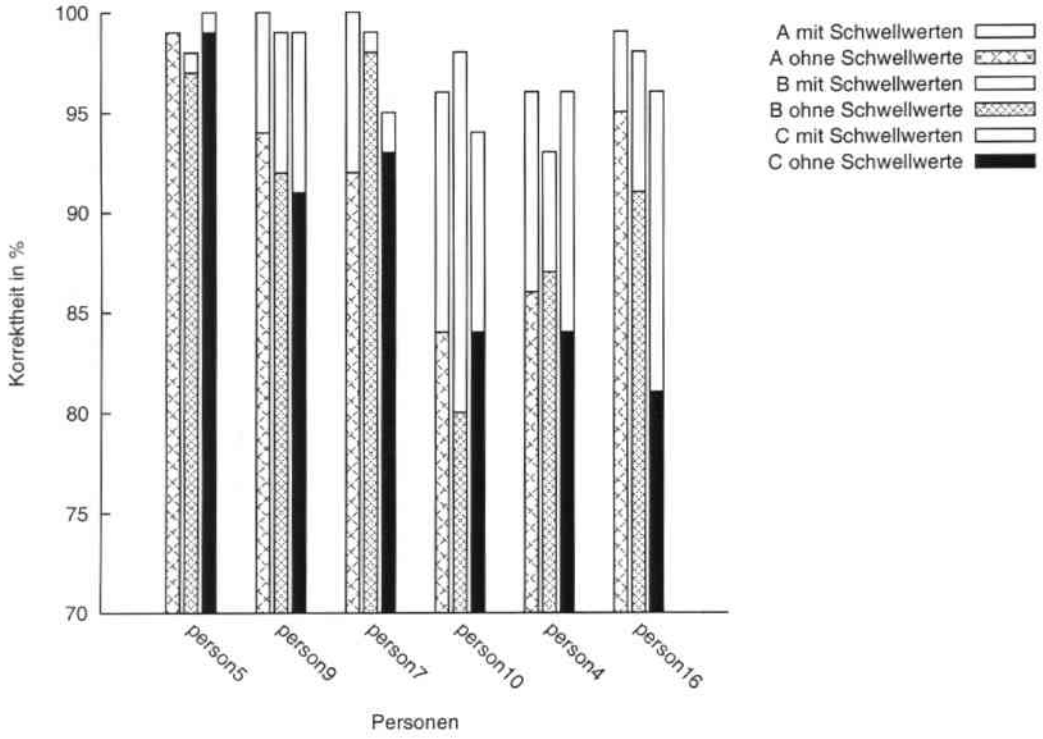


Abbildung 4.7: Korrektheit der Erkennungsergebnisse mit und ohne Schwellwerte

Aufnahme nur sechs statt elf bzw. zwölf Personen zur Verfügung. Nichtsdestotrotz beeindrucken die Ergebnisse der Spalten *W*.

Mit der Berücksichtigung der Schwellwerte geht nur ein kleiner Anteil der korrekten Erkennung verloren (siehe Spalten *X* der Tabelle 4.7). Der größte Rückgang der Erkennungsrate betrifft Personen mit Ergebnissen deutlich über 90%. Durchschnittlich fallen sie jedoch nur um 3,8%, 5,2% und 4,0% jeweils für Trackings *A*, *B* und *C*. Durch Hinzunahme der Schwellwerte gelingt es jedoch die meisten Eindringlinge zurückzuweisen. Ferner ist die korrekte Klassifikation deutlich höher als ohne Berücksichtigung der Schwellwerte, wie in den Spalten *Z* zu sehen.

Eine grafische Gegenüberstellung von Erkennung mit und ohne Schwellwerte ist in der Abbildung 4.7 zu finden.

Eine weitere interessante Beobachtung ist, dass trotz der kleineren Auflösung der Bilder im Vergleich zu „Portrait-View"-Aufnahmen das Ergebnis dieses Experiments deutlich über dem mit *P1* und *P2* Datensätzen liegt. Der Grund dafür könnte die „unscharfe" Repräsentation der Gesichter sein. Der Informationsgehalt ist ausreichend groß, so dass der Algorithmus noch genug aussagekräftige Unterschiede in den Bildern finden kann, ohne sich auf die Kleinigkeiten und viele Kanten zu konzentrieren. Dies erklärt auch den Verbesserungssprung des Glättungsfilters in der Vorverarbeitung – die Ausrichtung der Bilder fällt weniger ins Gewicht, z.B.

	P1			P2		
Name	A	B	C	A	B	C
person4	0,8	0,2	0,0	3,1	0,2	2,3
person9	8,2	11,8	5,3	42,0	63,6	36,0
person16	7,4	7,2	4,1	0,2	15,7	8,1
Durchschnitt	5,5	6,4	3,1	15,1	26,5	15,5

Tabelle 4.8: Erkennungsraten in % der Robo-View-Gesichter mit Training auf *P1* und *P2*. Ein Problem für den implementierten Eigenfaces-Ansatz.

haben die nur um wenige Prozent gedrehten Gesichter nach der Anwendung des Filters geringere Abweichungen zu der zuständigen Referenz.

4.8 Der Datensatz „Robo-View“

In diesem Versuch ist der Umstand zu untersuchen, ob eine auf „Portrait-View“ basierte Datenbank ausreichend ist, um Erkennung auf Gesichtern aus der „Robo-View“-Aufnahme durchzuführen. Dieser Punkt ist von großem Interesse, da nicht nur andere Beleuchtungsverhältnisse herrschen, sondern zusätzlich eine Änderung des Detailgrades der Gesichtsbilder vorliegt.

Training: P1, dann P2
Referenzen: k-Means
Tracking: A, B, C
Schwellwerte: nein
Auswertung: R

Die Aussagekraft der in Tabelle 4.8 dargestellten Ergebnisse ist durch die kleine Anzahl der übereinstimmenden Personen zwischen den Aufnahmen relativ schwach. Wir können nur darauf schließen, dass die Robo-View-Aufnahme für Person „p4“ auf der Pixelebene grundverschieden zu den Bildersätzen *P1* und *P2* war. Desweiteren fällt sofort auf, dass *P2* besser für die Erkennung geeignet war. Neben Zufall darf an dieser Stelle vermutet werden, dass eine ähnliche Beleuchtung (Kunstlicht in beiden Fällen) hier eine gewissen Bedeutung hat.

Insgesamt sieht man, dass der Einsatz des Systems unter diesen Umständen keinen bis wenig Erfolg verspricht und nicht empfehlenswert ist. Die Ergebnisse weisen darauf hin, dass es bei dem hier gewählten Eigenfaces-Ansatz notwendig ist, Trainingsdaten aus Aufnahmen unter den dem Systemeinsatz *ähnlichen* Bedingungen zu verwenden.

Satz/Tracking	A	B	C	Durchschnitt
P1	64,8	67,5	61,9	64,8
P2	74,2	65,4	55,9	65,2
R	83,0	78,1	81,2	80,8
Durchschnitt	74,0	70,3	66,3	

Tabelle 4.9: Kombination von Datensätzen, Erkennungsraten in %: Training auf allen Datensätzen mit Neuermittlung von zehn Referenzbildern.

4.9 Kombination von Datensätzen

Bis jetzt wurden immer alle Datensätze untereinander verglichen, um Einflussfaktoren für die Erkennung bei Änderung der Bedingungen zu ermitteln. Dabei wurde immer auf eine Gesichterdatenbank mit verschiedenen Kopfrotationen zugegriffen und somit eine rotationsunabhängige Erkennung angestrebt. Im praktischen Einsatz versucht man natürlich alle möglichen Abhängigkeiten auszuschließen, um bestmögliche Ergebnisse zu erreichen. Dies versuchen wir in diesem letzten Experiment: durch personenweise Vereinigung aller Trainingsdatensätze und Neuermittlung der Referenzen wird die Erkennung auf immernoch in *P1*, *P2* und *R* getrennten Testdaten durchgeführt.

Training: P1 + P2 + R
Referenzen: k-Means
Tracking: A, B, C
Schwellwerte: nein
Auswertung: getrennt in P1, P2, R

Zusammengefasst nach Datensatz und Tracking finden sich die Ergebnisse in der Tabelle 4.9. Wie erwartet, fallen die Ergebnisse durchschnittlich schlechter aus, als bei Trainings- und Testdatensätzen unter gleichen Aufnahmebedingungen. Allerdings darf man nicht vergessen, dass wir nun in der Lage sind, alle Datensätze P1, P2, R mit der so zusammengestellten Datenbank zu erkennen. Das gelingt durch Auswahl solcher Referenzbilder, die nicht nur ihren zugehörigen Datensatz gut beschreiben können, sondern auch noch große Ähnlichkeit mit den anderen Aufnahmen haben. Diese Aufgabe übernimmt das k-Means basierte Referenzierungsverfahren. Die Datenbank wurde in diesem Experiment auch nur mit zehn Referenzbildern initialisiert, und unter diesem Aspekt erreichen wir sehr gute Ergebnisse im Vergleich zu drei Datenbanken mit jeweils zehn Referenzbildern.

Insgesamt können wir sagen, dass bei gleicher Referenzenanzahl mehrere Datenbanken, nach einem bestimmten Kriterium getrennt, wie Beleuchtungsverhältnisse, durchaus sinnvoller sind, als eine große gemischte Datenbank, wenn es um Erkennungen von einzelnen Bildern geht. Mit Verlusten um 10-15% in der Erkennungsrate ist die große Datenbank vollkommen für die Erkennung von Bildsequenzen geeignet. In einem solchen Fall geht man davon aus, dass nur *eine* Person über

Name	ohne Schwellwerte Erkennung	Korrektheit	mit Schwellwerten Erkennung	Korrektheit
person5	95,7		91,2	98,0
person4	58,8		58,1	85,7
person8	81,1		70,0	95,8
person14	66,5		48,1	80,9
person7	94,3		90,2	99,3
person9-1	65,1		59,8	87,5
person10	83,5		81,0	95,7
person15	58,9		51,4	89,1
person12	89,5		60,0	90,0
person1	75,3		63,8	93,9
person6	70,8		61,3	90,0
person11	89,4		86,4	93,4
person13	85,2		69,3	93,0
person16	44,5		39,8	74,9
person3	62,3		54,5	74,9
person2	77,3		67,3	89,4
Durchschnitt	74,9		65,8	89,5

Tabelle 4.10: Kombination von Datensätzen und Schwellwerte, Angaben in %. Training und Auswertung auf personenweise vereinigten Datensätzen nach Neuermittlung der Referenzen.

einen gewissen Zeitraum lokalisiert und erkannt wird. In der Auswertung des Erkennungsergebnisses des nächsten Bildes werden die schon bekannten Ergebnisse der letzten Bilder berücksichtigt.

Interessant bleibt noch die Auswirkung auf die Ergebnisse, wenn wir wieder Schwellwerte berücksichtigen und somit Verdächtige ausschließen. Der Übersichtlichkeit halber werden auch die Testmengen aller Datensätze kombiniert.

Training: P1 + P2 + R
Referenzen: k-Means
Tracking: B
Schwellwerte: ja/nein
Auswertung: P1 + P2 + R

Wie aus der Tabelle 4.10 zu entnehmen ist, gelingt es in diesem Experiment einen Großteil der Falscherkennungen zu eliminieren. Denn die Korrektheit liegt im Schnitt bei fast 90%, mit jeder Person über fast 75%. Dies geschieht auf Kosten der Erkennungsrate, die somit um 9% niedriger ausfällt. Geht es um *präzisere* Ergebnisse, statt hohe Erkennungsraten, ist der Einsatz von Klassenschwellwerten notwendig.

5 Zusammenfassung

In der vorliegenden Arbeit wurde ein System zur Erkennung von Gesichtern mit dem Aspekt des Einsatzes auf mobilen Robotern entworfen und auf seine Schwachstellen hin durch geeignete Experimente untersucht. Als Grundlage für die Erkennung diente der *Eigenfaces*-Ansatz (vorgestellt in [TP91]). Er findet unter den Gesichtern Unterschiede, benutzt sie als Merkmale in Kombination für die Darstellung jedes einzelnen Gesichts. Bei Erkennung vergleicht er die Merkmale des Eingabegesichts mit den ihm bekannten Personen und gibt als Ergebnis die Person aus, die am nächsten liegt.

Als Vorverarbeitung kamen zum einen Lokalisierung von Gesichtern und zum anderen Bildnormierung zum Einsatz. Es wurden zwei Gesichtstracker (siehe [Nic02] und [Bir98]) verwendet, um ihren Einfluss auf die Erkennung untersuchen zu können. Für die Bildnormierung wurden nur einfache und schnelle Methoden, wie Helligkeitsanpassung und Weichzeichner verwendet.

Das System wurde um eine Offline-Funktionalität erweitert, um viele Experimente durchführen zu können. Um Einsätze auf Robotern zu simulieren wurde die Abarbeitung einer Bildersequenz eingeführt. Als Ergebnis erhält man in diesem Fall bildweise zusammengefasste Erkennungsraten für jede in Frage kommende Person.

Im System wurde eine Möglichkeit für Abweisung unbekannter Personen eingebaut. Dafür werden auf den vorhandenen Daten Klassenschwellwerte ermittelt, so dass sie einen Kompromiss zwischen der Falscherkennung und Zurückweisung der authentischen Personen darstellen.

Mit der im Rahmen dieser Arbeit entwickelten Software gelingt eine sehr gute Erkennung der frei rotierten Gesichter auf den Datensätzen aus gleichen Beleuchtungsverhältnissen. Die Erkennungsraten lagen dabei durchschnittlich im Bereich von 77% bis über 83%. Durch Vereinigung aller Datensätze ist die Erkennung, mit kleineren Abstrichen in der Erkennungsrate, auch unter unterschiedlichen Beleuchtungsverhältnissen möglich, mit einem gutem Ergebnis von fast 75%. Durch Hinzunahme der Schwellwerte wurde eine korrekte Klassifikation von fast 90% möglich. Wider Erwarten lieferte die Erkennung von den kleineren Gesichtern aus der „Robo-View"-Aufnahme erstaunliche 90% Erkennungsrate mit bis 100% und im Schnitt etwa 98% korrekter Klassifikation.

Durch die Versuche konnte Folgendes in Erfahrung gebracht bzw. aus anderen Arbeiten bestätigt werden. Beleuchtungsänderung beeinflusst die Erkennung drastisch. Mit weiteren Ansätzen, die eine beleuchtungsunabhängige Enkodierung der Bilddaten ermöglichen, schafft es der *Eigenfaces*-Ansatz auf diesem Gebiet robuster zu werden. Als vielversprechender Ansatz sei das SQI (self quotient image) erwähnt, welches in [WLW04] diskutiert wird. Durch Division eines Bildes durch sein geglättetes[1] Abbild, welches niederfrequent ist und somit fast ausschließlich Beleuchtung darstellt, erhält man ein Bild ohne überbelichtete Flächen und ohne Schattenbereiche.

Ein weiteres Problem im *Eigenfaces*-Verfahren stellten Ausrichtung, Positionierung sowie Skalierung der Gesichter dar. Hier sind die Trackingmethoden noch nicht robust genug, um pixelgenaues Lokalisieren zu ermöglichen. Es ist möglich, die Auswirkungen dieses Problems zu minimieren, indem man eine Umgebungssuche und zusätzlich Skalierungsschleifen in den Erkennungsprozess einbaut, jedoch auf Kosten der Geschwindigkeit – mit bis zu $O(n^3)$. Die im diesem System eingesetzte Umgebungssuche verlangsamen die Erkennung um das 49fache.

Auf einem Pentium M mit 1,6 GHz benötigte die Erkennung von 12.515 Bildern mit 160 Referenzen und 40 Eigenfaces 356 Sekunden. Dies entspricht 35,2 Bildern pro Sekunde. Für 4.505 Bilder auf einer mit 60 Referenzen und mit 40 Eigenfaces initialisierten Datenbank wurden 97 Sekunden benötigt, also eine Geschwindigkeit von 46,4 Bildern pro Sekunde. Ohne den Iterationsfaktor von 49 bekommen wir eine Roherkennungsleistung von jeweils 1.722 und 2.275 Gesichtern pro Sekunde. Die gesamte Initialisierung der Datenbank dauerte in beiden Fällen etwa maximal fünf Sekunden.

[1] z.B. Gaußscher Weichzeichner

Literaturverzeichnis

[Bir98] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.

[GB98] Athinodoros S. Georghiades and Peter N. Belhumeur. Illumination cone models for faces recognition under variable lighting. In *IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, 1998.

[Kon97] K. Konolige. Small vision systems: Hardware and implementation. In *Eighth International Symposium on Robotics Research*, Hayama, Japan, 1997.

[Nic02] Kai Nickel. 3D-Tracking von Gesicht und Händen mittels Farb- und Tiefeninformation. Studienarbeit, Fakultät für Informatik, Universität Karlsruhe, 2002.

[SBB02] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *IEEE Conference on Automatic Face and Gesture Recognition*, May 2002.

[TP91] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, March 1991.

[WLW04] Haitao Wang, Stan Z Li, and Yangsheng Wang. Face recognition under varying lighting conditions using self quotient image. In *IEEE Conference on Automatic Face and Gesture Recognition*, Beijing, China, 2004.