Technical Report TR-05-008, Mathematics and Computer Science Department, University of Mannheim, September 2005
by Stephan Kopf, Alexander Oertel

REIHE INFORMATIK
TR-05-008

Gesichtserkennung in Bildern und Videos mit Hilfe von Eigenfaces

Stephan Kopf, Alexander Oertel
University of Mannheim
– Fakultät für Mathematik und Informatik –
Praktische Informatik IV
A5, 6
D-68131 Mannheim, Germany

Gesichtserkennung in Bildern und Videos mit Hilfe von Eigenfaces

Stephan Kopf, Alexander Oertel
Praktische Informatik IV
University of Mannheim
Germany
kopf@informatik.uni-mannheim.de

ZUSAMMENFASSUNG

In dieser Arbeit wird ein System vorgestellt, um Gesichter von Personen in Graustufenbildern zu lokalisieren und die Gesichter dann mit gespeicherten Charakteristika anderer Gesichter zu vergleichen. Der Ansatz, der hierzu verwendet wurde, behandelt die Gesichtserkennung als echtes zweidimensionales Erkennungsproblem, und greift nicht auf dreidimensionale Geometrie zurück. Außerdem wurde ausgenutzt, dass Gesichter in Bildern normalerweise aufrecht auftreten und somit durch eine kleine Anzahl von zweidimensionalen Charakteristika beschrieben werden können.
Das System funktioniert, indem Gesichter in einen Eigenschaftenraum projiziert werden, der aus den wichtigsten Bestandteilen bekannter Gesichter aufgespannt wird. Diese aussagekräftigen Bestandteile werden *Eigenfaces* genannt, weil sie die Eigenvektoren (Hauptbestandteile) der Menge von Gesichtern sind, die als Lernmenge angegeben wurde. Die Eigenvektoren entsprechen jedoch nicht unbedingt den, aus dem menschlichen Blickwinkel, wichtigen Eigenschaften eines Gesichtes, wie Augen, Ohren und Nasen. Eine Projektion des Ursprungsbildes in den Gesichtsraum wird durch die gewichtete Summe der Eigenschaften der Eigenfaces charakterisiert. Eine Erkennung eines bestimmten Gesichtes erfolgt durch einen Vergleich der Gewichtsvektoren mit Gewichtsvektoren bereits bekannter Gesichter.

KEYWORDS

Gesichtserkennung, Bildanalyse, Videoanalyse

1. MOTIVATION

Nicht erst seit den Anschlägen am 11.September 2001 sind biometrische Verfahren ein Thema, das in der Öffentlichkeit heiß diskutiert wird. Schon früher wurde das Potential, aber auch die Gefahr erkannt, mithilfe von Computerprogrammen biometrische Merkmale von bestimmten Personen zu erfassen und zu vergleichen. Da diese Systeme auch in kritischen Einsatzgebieten angewendet werden sollen, müssen sie möglichst exakt funktionieren. Diese biometrischen Systeme werden vorwiegend zur Identifikation oder zur Verifikation eingesetzt.

Bei der Identifikation wird ein unbekanntes Muster mit Referenzmustern in einer Datenbank verglichen. Das System weiß nicht im Voraus, um welche Person es sich handelt. Ist die Ähnlichkeit mit einem gespeicherten Referenzmuster genügend groß, so war die Identifikation für das System erfolgreich, die Person wurde erkannt. Das System muss hier ein unbekanntes Muster mit Mustern einer Datenbank vergleichen (1:n Vergleich).

Bei der Verifikation läuft der Vorgang etwas anders ab. Hier weiß das System, um welche Person es sich handeln soll, und überprüft lediglich, ob die Person auch diejenige ist, für die sie sich ausgibt. Ist die Ähnlichkeit des Eingabemusters mit dem Referenzmuster ausreichend groß, so ist die Verifikation erfolgreich. Das System muss hier nur einen Vergleich durchführen (1:1 Vergleich).

Verfahren hierzu sind längst nicht mehr nur Theorie, sondern schon in der Praxis implementiert. Es gibt viele verschiedene Möglichkeiten eine Person anhand biometrischer Merkmale zu überprüfen. Moderne Systeme können Menschen innerhalb weniger Sekunden anhand der Netzhaut, der Iris, des Fingerabdrucks, des Gesichts oder der Sprache erkennen. Nichts desto trotz ist es natürlich das Gesicht, welches für Menschen bei der Erkennung von Personen die wichtigste Rolle spielt. Daher ist den Menschen die automatische Gesichtserkennung auch das vertrauteste biometrische Verfahren. Dabei spielt es keine Rolle, ob uns das Gesicht auf einem Foto, in einem Video oder in der Realität begegnet. Wir können tausende Gesichter, die wir in Laufe unseres Lebens kennen gelernt haben, innerhalb von Sekunden als bekannt oder unbekannt einstufen. Diese Fähigkeit ist soweit entwickelt, dass wir Gesichter trotz Veränderungen der äußeren Umstände, des Gesichtsausdruckes, des Alters und anderer Veränderungen (Brille, Frisur, Bart ...) immer noch erkennen können. Sogar unscharfe, gedrehte oder verkleinerte Fotos sind keine große Herausforderung für unser Gehirn. Und selbst auf größere Entfernung bleibt die Erkennungsleistung bemerkenswert.

Mit fortschreitender Technik und dem Versuch das Leben für die Menschen einfacher, sicherer und bequemer zu machen, hält auch die Technik der Gesichtserkennung immer mehr Einzug in das alltägliche Leben. Öffentliche Plätze, wie Flughäfen, Bahnhöfe oder Sportstadien werden überwacht, um frühzeitig kriminelle Subjekte erkennen und die eventuell daraus entstehenden Gefahren beseitigen zu können. Aber diese Systeme können natürlich nicht nur für solche Zwecke verwendet werden. Es ist angedacht, Häuser mit solchen Systemen auszustatten, damit man zum Beispiel weiß, wer sich wo im Haus befindet [17]. In Kaufhäusern oder Supermärkten kann man das Kaufverhalten bestimmter Altersgruppen oder sogar bestimmter Personen genau beobachten und die gewonnenen Informationen für sich nutzen [13]. Auch für die automatische Farbfilmentwicklung ist die Gesichtserkennung ein interessanter Aspekt, weil der Effekt vieler Bildverbesserungs- und Rauschreduktionstechniken vom Bildinhalt abhängt.

Ziel war es, ein computergestütztes Modell der Gesichtserkennung zu implementieren, das schnell, einfach und in einer eingeschränkten Umgebung, wie zum Beispiel in einem Büro oder in einem

Haushalt, effektiv arbeitet. Das ist jedoch nicht so einfach, da Gesichter eine natürliche Klasse von Dingen sind, und damit (im Gegensatz zu elektronischen Signalen) sehr komplex sind. Schließlich sind sie multidimensional und ihr Aussehen ändert sich ständig mit der Mimik. Viele Erkennungsverfahren schränken daher die Erkennung auf eine wohldefinierte zweidimensionale Form (Foto) ein. Wohldefiniert bedeutet in diesem Sinne, dass das Gesicht nur frontal, ohne störenden Hintergrund gezeigt wird.

Bei der Eigenfacemethode hingegen, müssen diese Bilder nicht unbedingt wohldefiniert sein. Einzige Voraussetzung für das Verfahren sind genügend unterschiedliche Variationen der Gesichter in der Lernmenge. Damit ist die Eigenfacemethode auch fähig Gesichter unter suboptimalen Bedingungen zu erkennen. Das wiederum erlaubt der Methode, auch Gesichter in Videos zu erkennen, die meistens unter nicht optimalen Bedinungen gefunden werden.

Das dieser Arbeit zugrunde liegende Schema basiert auf einem Informationstheorieansatz, der die Gesichter in kleine Mengen von charakteristischen Eigenschaftsbildern zerlegt, die Eigenfaces genannt werden und die man sich als die Hauptbestandteile der ursprünglichen Lernmenge vorstellen kann. Die Erkennung wird durchgeführt, indem man ein neues Bild in den Unterraum, den so genannten "face space", der von den Eigenfaces aufgespannt wird, projiziert und das Gesicht klassifiziert, indem man seine Position im "face space" mit der Position von bekannten Personen vergleicht.

Die Erkennung unter variierenden Umständen wird durch das Training mit einer begrenzten Anzahl von charakteristischen Sichten (zum Beispiel: Frontansicht, eine 45° Sicht und eine Profilansicht) erreicht. Der Ansatz ist schneller und einfacher als andere Gesichtserkennungsschemata, lernt besser und ist unempfindlich gegenüber kleinen Veränderungen der Gesichtsbilder.

2. VERFAHREN ZUR KLASSIFIKATION VON GESICHTERN

Der große Vorteil der Gesichtserkennung gegenüber anderen biometrischen Verfahren liegt darin, dass Personen identifiziert werden können, ohne es zu wissen. Das heißt, sie müssen keine besonderen Aktionen ausüben, um erkannt zu werden, wie es zum Beispiel bei der Iriserkennung oder beim Vergleich von Fingerabdrücken vonnöten ist.

Für die Gesichtserkennung gibt es viele verschiedene Methoden. Grundsätzlich lassen sich diese Methoden in zwei Hauptgruppen aufteilen. Auf der einen Seite gibt es die merkmalsbasierte Gesichtserkennung. Dieser Ansatz hat sich darauf konzentriert, einzelne Eigenschaften wie Augen, Nase, Mund und Kopfkonturen zu entdecken, und ein Gesichtsmodell anhand der Position, der Größe und der Beziehungen zwischen diesen Eigenschaften zu definieren.

Es hat sich jedoch herausgestellt, dass diese Ansätze schwer auf mehrere Sichten zu erweitern sind. Außerdem sind sie sehr instabil. Daher bedarf es einer guten Anfangschätzung, damit diese Methoden gut funktionieren. Zusätzlich haben Forschungen im Bereich der menschlichen Vorgehensweise zur Gesichtserkennung gezeigt, dass die einzelnen Eigenschaften und ihre unmittelbaren Beziehungen nur eine unzulängliche Qualität der Gesichtserkennung ermöglichen [6]. Trotzdem ist dieser Ansatz in der Literatur sehr beliebt und weit verbreitet. Zu diesem Ansatz zählen zum Beispiel Methoden wie die Gesichtserkennung anhand der Gesichtsmetrik, das "Elastic Bunch Graph Matching" [27] und die Gesichtserkennung anhand geometrischer Merkmale [5].

Auf der anderen Seite gibt es den holistischen Ansatz, der sich nicht auf einzelne Merkmale spezialisiert, sondern bei dem das ganze Gesicht betrachtet wird und auch die Klassifikation anhand des ganzen Gesichtes durchgeführt wird. Hierzu zählen Methoden wie das Template Matching [5], die Gesichtserkennung mithilfe der Fourier-Transformation [23], die Fisherface-Methode [3] und die Eigenface-Methode [25].

Natürlich lässt sich die Grenze zwischen diesen Methoden nicht klar ziehen. In der Praxis kommt es meistens zu einer Vermischung beider Ansätze. So kann man zum Beispiel beim Template Matching, bei dem normalerweise das gesamte Gesicht zur Klassifizierung verwendet wird, auch mehrere Templates verwenden, um dann einzelne Merkmale, wie Augen, Mund oder Nase auf Ähnlichkeiten zu untersuchen.

Im Folgenden werden aus beiden Gruppen unterschiedliche Verfahren zur Gesichtserkennung vorgestellt. Im nächsten Kapitel wird auf das Verfahren der Eigenfaces eingegangen.

2.1 Gesichtsmetrik

Die Gesichtsmetrik konzentriert sich auf spezifische Gesichtscharakteristika wie Augen, Nase und Mund. Es werden die Position und Größe sowie die Verhältnisse der einzelnen Kennzeichen zueinander vermessen und daraus ein Modell des Gesichts errechnet. Beim Vergleich zweier Gesichter werden die Abstände und Winkel zwischen geometrischen Punkten (zum Beispiel Augen- oder Mundwinkel) herangezogen. Nachteilig an diesem Verfahren erweist sich die geringe Zuverlässigkeit der Vermessungen und die geringe Anzahl an messbaren Merkmalen.

2.2 Elastic Bunch Graph Matching

Bei diesem Verfahren wird dem Gesicht ein Gitternetz, der "Labeled Graph", zugeordnet. An den Knoten dieses Graphes berechnet ein Algorithmus wichtige Merkmale des Gesichtes, so genannte "Landmarks" (Abbildung 1).

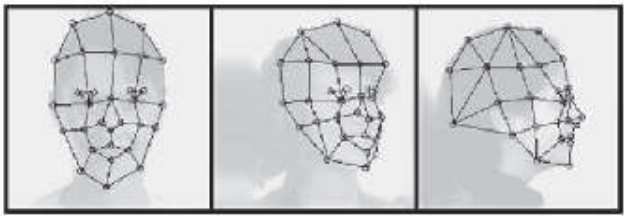

Abbildung 1: Gitternetz des Elastic Bunch Graph Matching

Die lokalen Merkmale eines Bildes werden mit Gabor Wavelets [18] extrahiert. Das Gabor Wavelet ist eine ebene Welle mit der Frequenz k, die von einem Gauß-Fenster begrenzt wird (Abbildungen 2, 3).

Gabor Wavelets werden verwendet, weil sie sehr robust gegenüber Helligkeitsänderungen und Kontraständerungen sind. Das Wavelet ist definiert durch:

$$\psi_k(x) = \frac{k^2}{\sigma^2} e^{(-\frac{k^2 x^2}{2\sigma^2})} [e^{ikx} - e^{(-\frac{\sigma^2}{2})}] \quad (1)$$

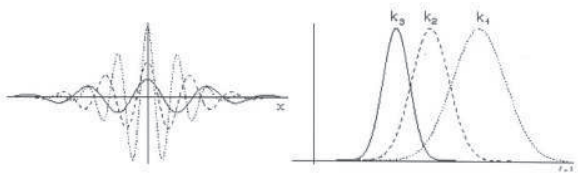

Abbildung 2: **2D Gabor-Wavelets, links im Ortsbereich, rechts im Frequenzbereich [18]**

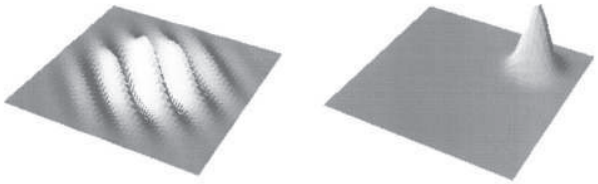

Abbildung 3: **3D Gabor-Wavelet, links im Ortsbereich, rechts im Frequenzbereich [18]**

Für die Bildposition x kann man nun anhand dieses Wavelets die Wavelet - Komponente J_k mit der Grauwertverteilung I(x) errechnen.

$$J_k(x) = \int dx' I(x') \psi_k(x - x') \tag{2}$$

Für jedes Pixel werden 40 komplexe Werte berechnet, die 40 unterschiedlichen Gabor-Wavelets entsprechen. Diese 40 Koeffizienten, bilden im Zentrum eines Bildpunkts x einen Vektor, den so genannten Jet (Abbildung 4).

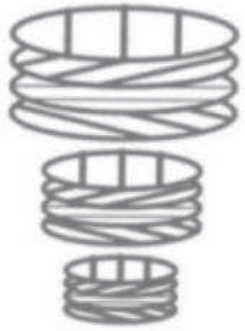

Abbildung 4: **Jet, bestehend aus 12 Gabor-Wavelets (4 Orientierungen bei 3 unterschiedlichen Größen) (Quelle [15])**

Dieser Vektor beschreibt die lokale Grauwertverteilung bzw. die lokalen Merkmale des Gebiets um die Stelle x herum. Zur Beschreibung eines Gesichts wird jetzt der Labeled Graph über das Gesicht gelegt (Abbildung 5 links). Die Knoten sind mit Jets beschriftet. Die Kanten des Graphen sind mit Vektoren, die den Abstand zwischen den einzelnen Knoten repräsentieren, beschriftet.

Um nun ein unbekanntes Gesicht durch einen Graphen darzustellen, wird der Prozess des "Elastic Bunch Graph Matching" benutzt. Der so genannte "Face Bunch Graph" enthält an bestimmten Knotenpositionen mehrere verschiedene Jets die unabhängig voneinander ausgewählt werden können und zu einem neuen Graphen zusammengesetzt werden. Die Jets des unbekannten Bildes werden mit allen Jets des Bunch Graphs verglichen und der jeweils am besten passende wird ausgewählt (Abbildung 5 rechts). Die Länge und der Winkel jeder Kante des Graphen werden gespeichert.

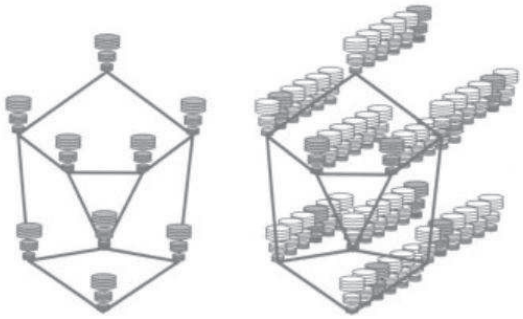

Abbildung 5: **links: Labeled Graph; rechts: Face Bunch Graph (Quelle [15])**

Das Elastic Bunch Graph Matching läuft nach dem folgenden Schema ab:

- Lokalisierung des Gesichts
- Lokalisierung der Landmarks
- Vergleich der Graphen (Vergleich der Längen und Winkel der Kanten)

Mit dieser Methode können sehr gute Erkennungsraten erreicht werden. Das Verfahren ist robust gegenüber Helligkeits- und Kontraständerungen, so dass die Fehlerraten relativ gering sind.

2.3 Gesichtserkennung anhand geometrischer Merkmale

Ein weiteres merkmalsbasiertes Verfahren ist die Gesichtserkennung mithilfe geometrischer Merkmale. Bei dieser Methode werden geometrische Merkmale des Gesichts aus einem Bild extrahiert und als Zahlenwerte in einem Vektor gespeichert. Als geometrische Merkmale gelten unter anderem die Position der Nase, der Augen oder des Mundes und ihre relative Position zueinander. Im Folgenden wird nur die von Brunelli und Poggio [5][4] entwickelte Methode näher betrachtet.

Als erstes müssen die Anforderungen an den Vektor, beziehungsweise an die einzelnen Merkmale des Vektors, festgelegt werden:

- Die Extraktion der Merkmale muss möglichst einfach sein.
- Die Abhängigkeit von der Umgebungsbeleuchtung muss relativ gering sein.
- Die Abhängigkeit von kleinen Änderungen im Gesicht muss möglichst gering sein.

- Der Vektor muss einen möglichst hohen Informationsgehalt besitzen.

Bevor die eigentliche Gesichtserkennung durchgeführt wird, sollte noch eine Normierung des Gesichts durchgeführt werden. Beispielsweise wird hierbei eine seitliche Neigung des Kopfes durch den Einsatz von Winkeln ausgeglichen.

Bei der anschließenden Merkmalsextraktion, die schrittweise erfolgt, hilft folgende Beobachtung weiter: Jedes Gesicht erfüllt bestimmte Bedingungen und die Merkmale eines Gesicht sind immer gleich angeordnet (die Augen liegen nebeneinander, die Nase liegt unter den Augen, ...). Die Tatsache, dass der Aufbau von Gesichtern immer gleich ist, erleichtert die Merkmalsextraktion.

Die Merkmale werden mit dem Verfahren der integralen Projektion extrahiert. Bei diesen Projektionen werden die Grauwerte einer Spalte bzw. einer Zeile eines Bildes aufsummiert und in einem Histogramm dargestellt (Abbildung 6). Die exakte Position und Größe eines Merkmales kann sehr einfach mit diesen Projektionen extrahiert werden, wenn das verwendete Fenster genügend klein ist. Vor der Projektion muss aber die Lage der verschiedenen Merkmale abgeschätzt werden.

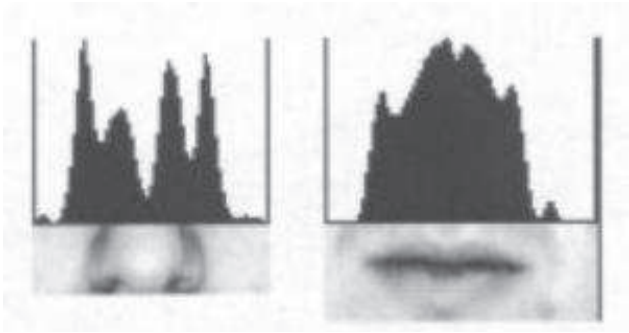

Abbildung 6: **Vertikale Projektion von Mund und Nase (Quelle [5])**

Als erstes werden anhand eines Gradientenoperators die Kanten des Gesichts in ein vertikales und ein horizontales Gradientenbild aufgeteilt (Abbildung 7).

Aus diesem Kantenbild wird durch Template Matching die Lage der Augen ermittelt, danach erfolgt die Extraktion der anderen Merkmale mithilfe des Aufbaus des Gesichts:

- Die ungefähre vertikale Position von Nase und Mund wird aus Durchschnittswerten für menschliche Gesichter abgeschätzt.
- Um die wirkliche Position von Nase und Mund zu ermitteln, wird nach einem Maximum (Nasenspitze) im horizontalen Kantenbild im Bereich der geschätzten Position der Nase gesucht. Genauso wird an der geschätzten Position des Mundes nach einem Minimum (helle Linie zwischen den Lippen im Kantenbild) gesucht.
- Danach wird der Abstand der Maxima und Minima von ihrer geschätzten Position gewichtet (Gewichtung mit Gauß-Faktor). Die Werte mit der höchsten Gewichtung werden als die vertikale Position von Nase und Mund angenommen. Das Suchfenster wird danach verkleinert.

Abbildung 7: **Vertikales und horizontales Kantenbild (Quelle [5])**

- Die horizontale Begrenzung (Breite) der Nase ergibt sich aus den beiden äußersten Maxima in der vertikalen Projektion des horizontalen Kantenbildes, deren Betrag über dem durchschnittlichen Wert im betrachteten Fenster liegt.
- Auf dieselbe Weise wird die Höhe des Mundes aus dem vertikalen Gradientenbild ermittelt.
- Die Breite des Mundes wird aus der vertikalen Projektion des horizontalen Kantenbildes extrahiert.
- Auf ähnliche Weise wie beim Mund und der Nase werden nun die Position und die Dicke der Augenbrauen ermittelt.
- Am Ende schließlich wird der Umriß des Gesichtes berechnet.

Diese 22 Merkmale sind graphisch dargestellt (Abbildung 8). Es besteht aber durchaus die Möglichkeit andere beziehungsweise zusätzliche Merkmale für die Gesichtserkennnung zu extrahieren.

Abbildung 8: **22 Merkmale im Gesicht, die zur Identifikation einer Person verwendet werden (Quelle [5])**

Die Ähnlichkeit zweier Personen wird durch den Vergleich des Vektors der zu identifizierenden Person mit allen anderen Vektoren, die in einer Datenbank gespeichert wurden, festgestellt. Die beiden Vektoren, die am Besten übereinstimmen werden dann derselben Person zugeordnet. Für die Berechnung der Ähnlichkeit wird die "Nearest Neighbour" Methode verwendet (Berechnung über die Summe der Fehlerquadrate).

Vorteil dieser Methode ist die relativ einfache Berechnung der Merkmale aus einem gegebenen Gesicht. Probleme treten jedoch dann auf, wenn Gesichter unter verschiedenen Winkeln aufgenommen werden, denn dann verändern sich die extrahierten geometrischen Daten. Ein wesentlicher Punkt ist die Bestimmung von Normierungsfaktoren, mit denen aus einem gedrehten Gesicht auf die Daten des ursprünglichen Gesichts geschlossen werden kann.

2.4 Korrelation

Bei der Korrelation handelt es sich vielleicht um das einfachste Klassifizierungsschema. Hier wird im Bildraum nach dem *nächsten Nachbarn* sortiert [5]. Ein Bild in der Testmenge wird erkannt (klassifiziert), indem man ihm den nächsten Punkt in der Lernmenge, an dem die Entfernungen im Bildraum gemessen wird, als Kennzeichen zuweist. Wenn alle Bilder normalisiert sind, das heißt, einen Mittelwert von Null und eine einheitliche Varianz haben, dann entspricht diese Vorgehensweise dem Wählen des Bildes aus der Lernmenge, das am Besten mit dem Testbild korreliert. Wegen der Normalisierung sind die Ergebnisse unabhängig von der Intensität der Lichtquelle und den Auswirkungen der automatischen Verstärkungsregelung von Videokameras.

Diese Vorgehensweise hat aber etliche bekannte Nachteile:

- Wenn die Bilder in der Lernmenge und in der Testmenge unter verschiedenen Belichtungszuständen gesammelt werden, kann es sein, dass die entsprechenden Punkte im Bildraum weit auseinander liegen.
- Die Methode ist rechnerisch sehr aufwendig. Man muss das Testbild mit jedem einzelnen Bild aus der Lernmenge korrelieren.
- Diese Methode braucht viel Speicherplatz, da die Lernmenge zahlreiche Bilder jeder Person enthalten muss.

2.5 Lineare Unterräume

Diese Methode verwendet die Beobachtung für Lambertsche Oberflächen ohne Schatten, dass die Bilder eines bestimmten Gesichtes in einem dreidimensionalen linearen Unterraum liegen.

Sei p ein Punkt auf einer Lambertschen Oberfläche die durch eine Punktlichtquelle in der Unendlichkeit erleuchtet wird. Sei $s \in R^3$ ein Spaltenvektor der das Produkt aus der Lichtquellenstärke und dem Einheitsvektor der Lichtquellenrichtung darstellt. Wenn die Oberfläche von einer Kamera betrachtet wird, ist die resultierende Bildintensität des Punkte p gegeben durch

$$E(p) = a(p)n(p)^T s \tag{3}$$

wobei n(p) der innere Einheitsnormalvektor der Oberfläche des Punktes p und a(p) das Rückstrahlvermögen der Oberfläche an der Stelle des Punktes p ist [12]. Daraus ist ersichtlich, dass die Bildintensität des Punktes p linear auf $s \in R^3$ ist. Somit können, soweit keine Schatten vorhanden sind, das Rückstrahlvermögen und die Oberflächennormale wieder hergestellt werden, wenn drei Bilder der Lambertschen Oberfläche aus demselben Blickwinkel unter drei bekannten, linear unabhängigen Lichtquellenrichtungen gegeben sind. Diese Methode ist auch als Photometrisches Stereo bekannt [21][28]. Andererseits kann man die Oberfläche des Bildes unter zufälligen Belichtungsrichtungen durch die Linearkombination der drei originalen Bilder rekonstruieren [20].

Für die Einordnung ist diese Tatsache von größter Wichtigkeit. Es zeigt, dass die Bilder der Lambertschen Oberfläche, für einen fixierten Blickwinkel, in einem dreidimensionalen linearen Unterraum des hochdimensionalen Bildraumes liegen. Um Lambertsche Oberflächen zu erkennen, legt diese Beobachtung einen einfachen Klassifizierungsalgorithmus nahe, der unempfindlich gegenüber einer großen Veränderung der Belichtungsbedingungen ist.

Für jedes Gesicht sollte man drei oder mehr Bilder unter verschiedenen Belichtungsrichtungen verwenden, um ein dreidimensionales Grundgerüst für den linearen Unterraum zu schaffen. Man beachte, dass die drei Basisvektoren dieselbe Dimension wie die Trainingsbilder haben, und dass man sie sich als Basisbilder vorstellen kann. Um die Erkennung durchzuführen wird einfach die Entfernung des neuen Bildes zu jedem linearen Unterraum berechnet und das Gesicht gewählt, das der kürzesten Entfernung entspricht.

Solange es kein Rauschen oder keine Schattierungen gibt, würde der Algorithmus, der lineare Unterräume verwendet, unter allen Belichtungsbedingungen fehlerfreie Einordnungen erreichen, solange die Oberflächen dem Lambertschen Reflektierungsmodell folgen würden. Nichtsdestoweniger gibt es einige Gründe, anderweitig weiterzuforschen:

- Wegen Schattierungen, Spiegelungen und Gesichtsausdrücken gibt es in einigen Regionen von Bildern von Gesichtern Veränderungen, die nicht mit dem linearen Unterraummodell übereinstimmen.
- Um ein Testbild zu erkennen, muss man die Entfernung zum linearen Unterraum für jede Person bestimmen.
- Der Algorithmus für die linearen Unterräume muss für jede Person drei Bilder im Speicher halten.

2.6 Template Matching

Ein häufig angewandtes Verfahren zur Gesichtserkennung ist das Template Matching [1]. Templates sind vorgegebene Masken die mit dem zu erkennenden Gesicht oder mit Teilen des zu erkennenden Gesichtes verglichen werden. Beim einfachen Template Matching wird die Ähnlichkeit des Bildes mit einem einzigen Template berechnet. Es handelt sich hier entweder um ein Template, dass das ganze Gesicht, oder das nur einen Teil des Gesichtes repräsentiert. (Abbildung 9).

Brunelli und Poggio beschreiben in ihrem Artikel [5] eine Verbesserung des Ansatzes. Sie verwenden nicht mehr nur ein einzelnes Template, sondern eine Frontalaufnahme des Gesichtes und vier Templates, die die Augen, den Mund, die Nase und das gesamte Gesicht (unterhalb der Augenbrauen) darstellen (Abbildung 10).

Für die Erkennung eines unbekannten Gesichtes wird nun ein Vergleich dieses Gesichtes mit allen in der Datenbank gespeicherten Bildern durchgeführt. Bei einer Gegenüberstellung mit n Referenzmustern erhält man einen n-dimensionalen Vektor, der die Ähn-

Abbildung 9: **Referenzbild und ein Template, das mit der Augenregion verglichen wird (ähnlich Quelle [1])**

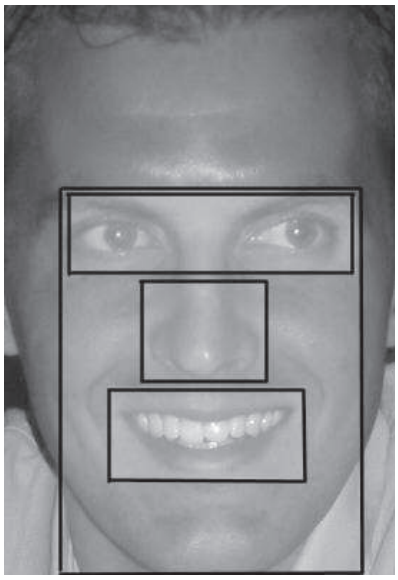

Abbildung 10: **Vier Templates für Augen, Nase, Mund und das ganze Gesicht (ähnlich Quelle [5])**

lichkeit jedes Referenzmusters mit dem getesteten Bild angibt. Das gesuchte Gesicht wird dann als das Gesicht erkannt, das die höchste kumulierte Ähnlichkeit aller Merkmale hat. Das Template Matching führt einen direkten Vergleich von Bildsegmenten durch. Allerdings werden zu einem effizienten Einsatz dieses Verfahrens die gleiche Größe, Orientierung und Beleuchtung der zu vergleichenden Bilder vorausgesetzt. Die Ergebnisse hängen stark von der Qualität der verwendeten Maske ab. Problematisch erweist sich nämlich bei dieser Methode der Sachverhalt, dass die Referenzmuster bei möglichst vielen unterschiedlichen Personen anwendbar sein müssen und möglichst unabhängig von Helligkeits- oder Kontraständerungen sein sollten. Das größte Problem des Template Matching jedoch ist der enorm hohe Rechenaufwand.

Eine verbesserte Form des Template Matching ist das Deformable Template Matching. Diese Variante des Template Matching vergleicht nicht nur die Bildsegmente, sondern dreht, verschiebt, und verformt diese innerhalb gewisser Grenzen. So kann eine möglichst hohe Ähnlichkeit mit dem Referenzbild erreicht werden. In einer weiteren Abwandlung dieser Methode wird ein Gitter über das zu identifizierende Gesicht gelegt. Auch hier ist als großer Nachteil der enorme Berechnungsaufwand zu nennen.

2.7 Fourier Transformation

Die Idee der Fourier-Transformation [23] besteht darin, das Originalbild und das Vergleichsbild in den Frequenzbereich zu transformieren, um dort die Spektren der beiden Bilder einfacher vergleichen zu können.

Bei der Fourier-Transformation wird jedes Pixel im Ausgangsbild anhand aller Pixel im Eingangsbild berechnet (globaler Operator). Für die Verarbeitung von Bildern wird die zweidimensionale diskrete Fourier - Transformation (DFT) verwendet. Die Transformation erfolgt also vom Ortsbereich in den Frequenzbereich (Abbildung 11).

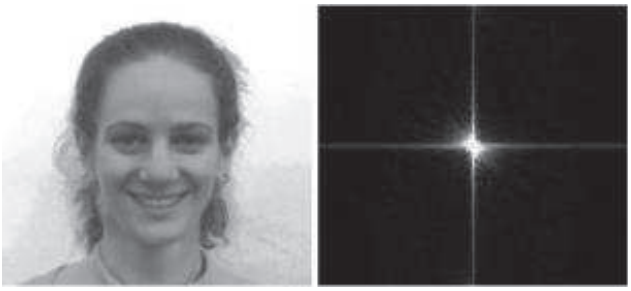

Abbildung 11: **Gesicht und seine Transformation in den Frequenzbereich (Quelle [23])**

Der größte Teil der Bildinformationen ist in den niederen Frequenzen enthalten. Mithilfe der Varianz werden die Frequenzen berechnet, die Informationen über die Unterschiede zwischen den Bildern enthalten. Man erkennt, dass auch die Informationen über Bildunterschiede zum großen Teil in den Fourierkoeffizienten niedriger Ordnung enthalten sind. Diese Informationen sind letztendlich die relevanten Informationen für die Gesichtserkennung. Die Fourierkoeffizienten werden nun entsprechend ihres Informationsgehaltes angeordnet und nur die ersten N Koeffizienten für die Gesichtserkennung verwendet.

Nachdem die Anzahl der Koeffizienten festgelegt wurde, erfolgt

die Klassifikation der Gesichter. Zuerst muss eine Datenbank mit den Referenzbildern erzeugt werden, die in Form von Vektoren, die die verschiedenen Fourierkoeffizienten enthalten, gespeichert sind. Beim Vergleich zweier Bilder wird die euklidische Norm zwischen dem Vektor des zu testenden Bildes und allen Vektoren, die sich in der Datenbank befinden, berechnet. Die beiden Vektoren, die den geringsten Abstand besitzen, werden als zum selben Gesicht gehörig eingestuft. Zusätzlich kann noch ein Schwellwert eingeführt werden, damit Bilder, die keine Gesichter enthalten, nicht fälschlicherweise klassifiziert werden.

2.8 Fisherfaces

Die Gesichtserkennung mit Fisherfaces [3] ist empfindlich gegenüber großen Veränderungen in der Belichtung und im Gesichtsausdruck. Dabei sollte man bedenken, dass Unterschiede in der Belichtung nicht nur die Stärke des Lichtes umfasst, sondern auch die Richtung und die Anzahl der Lichtquellen. Wie aus Abbildung 12 deutlich wird, kann eine Person aus demselben Blickwinkel, mit demselben Gesichtsausdruck extrem unterschiedlich aussehen, wenn die Lichtquellen, die das Gesicht anleuchten, aus unterschiedlichen Richtungen kommen.

Abbildung 12: Dieselbe Person kann unter verschiedenen Belichtungsbedingungen unterschiedlich aussehen. Links ist die vorherrschende Lichtquelle fast direkt vor dem Gesicht, rechts ist die vorherrschende Lichtquelle oben rechts (Quelle [3]).

Dieser Ansatz für die Gesichtserkennung verwendet zwei Beobachtungen:

1. Alle Bilder einer Lambertschen Oberfläche, die von einem fixen Standpunkt unter veränderlichen Belichtungsbedingungen gemacht wurden, liegen in einem dreidimensionalen Unterraum des hochdimensionalen Bildraumes [20].

2. Die obige Beobachtung gilt bei Regionen mit Schattierungen, Spiegelungen oder Änderungen des Gesichtsausdruckes nicht vollständig. In der Praxis zeigen bestimmte Regionen im Gesicht Unterschiede von Bild zu Bild, die oft sehr weit von dem linearen Unterraum abweichen und folglich für die Erkennung weniger zuverlässig sind.

Diese Beobachtungen werden verwendet, um eine lineare Projektion des Gesichtes aus dem hochdimensionalen Bildraum in einen deutlich niedrigdimensionaleren Eigenschaftenraum zu finden, die sowohl unempfindlich gegen Veränderungen in der Beleuchtungsrichtung, als auch gegenüber dem Gesichtsausdruck ist.

Der vorherige Algorithmus der linearen Unterräume nutzt, zugegebenermaßen unter idealisierten Bedingungen, die Tatsache, dass die Unterschiede in der Klasse in einem linearen Unterraum des Bildraumes liegen. Infolgedessen sind die Klassen konvex und somit linear trennbar. Man kann die Reduzierung der Dimensionen mit der linearen Projektion durchführen und immer noch die lineare Trennbarkeit beibehalten. Das ist ein starkes Argument, um lineare Methoden für die Reduzierung der Dimensionen bei der Gesichtserkennung zu verwenden, zumindest, wenn man dabei eine Unempfindlichkeit gegenüber Belichtungsbedingungen sucht.

Da die Lernmenge gekennzeichnet ist, macht es Sinn, diese Information zu verwenden, um eine zuverlässigere Methode für die Reduzierung der Dimensionalität des Eigenschaftenraumes zu entwickeln. Es wird argumentiert, dass die Verwendung von klassenspezifischen linearen Methoden für die Reduzierung der Dimensionen und von einfachen Klassifizierern im reduzierten Eigenschaftenraum bessere Erkennungsraten bringt, als die lineare Unterraummethode. Fishers Lineare Diskriminante (FLD) [10] ist in dieser Hinsicht ein Beispiel für eine klassenspezifische Methode, da sie versucht, die Streuung zu "formen", um sie für die Klassifikation zuverlässiger zu machen. Diese Methode wählt die Matrix $W \in R^{n \times m}$ mit orthonormalen Spalten in [7] so, dass das Verhältnis von Streuung zwischen den Klassen zur Streuung innerhalb der Klassen maximiert wird. Hierbei ist n die Dimension des Bildraumes und m die Dimension des Eigenschaftenraumes mit $m < n$. $\mu \in R^n$ ist das Durchschnittsbild aller Trainingsbilder und c die Anzahl der Klassen in die die Bilder fallen können.

Sei die Streuung zwischen den Klassen definiert als

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{4}$$

und die Streuung innerhalb der Klassen definiert als

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \tag{5}$$

wobei μ_i das Durchschnittsbild der Klasse X_i und N_i die Anzahl der Elemente in der Klasse X_i ist. Wenn S_W nicht singulär ist, dann wird die optimale Projektion W_{opt} als Matrix mit den orthonormalen Spalten gewählt, die das Verhältnis zwischen der Determinante der Matrix der Streuung zwischen den Klassen der projizierten Beispiele zu der Determinante der Matrix der Streuung innerhalb der Klassen der projizierten Beispiele maximiert, das heißt

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1\ w_2\ ...\ w_m] \tag{6}$$

wobei $\{w_i | i = 1, 2, ..., m\}$ die Menge der verallgemeinerten Eigenvektoren von S_B und S_W ist, die den m größten generalisierten Eigenwerten $\{\lambda_i | i = 1, 2, ..., m\}$ entsprechen, das heißt

$$S_B w_i = \lambda_i S_W w_i \quad i = 1, 2, ..., m \tag{7}$$

Man beachte, dass es mindestens c-1 generalisierte Eigenwerte gibt, die nicht Null sind, wodurch eine obere Grenze für m durch c-1 definiert ist und c die Anzahl der Klassen ist, siehe [8].

Beim Problem der Gesichtserkennung ist man mit der Schwierigkeit konfrontiert, dass die Matrix $S_W \in R^{n \times n}$ der Streuung innerhalb einer Klasse immer singulär ist. Das kommt daher, dass der Rang von S_W mindestens N-c beträgt und im Allgemeinen die Anzahl der Bilder in der Lernmenge N wesentlich kleiner ist, als die Anzahl der Pixel in jedem Bild n. Das bedeutet, dass es möglich ist, die Matrix W so zu wählen, dass die Streuung innerhalb der Klasse der projizierten Bilder genau auf Null gesetzt werden kann.

Die Methode, die sich *Fisherfaces* nennt, vermeidet das Problem einer singulären Matrix S_W, indem man die Bildermenge in einen niedriger dimensionalen Raum projiziert, damit die daraus resultierende Matrix S_W der Streuung innerhalb der Klassen nichtsingulär ist. Dies wird dadurch erreicht, dass man die Hauptkomponentenanalyse, auch Principal Component Analysis (PCA) genannt, zur Reduzierung der Dimensionen des Eigenschaftenraumes auf N-c Dimensionen verwendet und die standardisierte FLD, die durch (6) definiert ist, anwendet, um die Dimension auf c-1 zu reduzieren.

Formal ausgedrückt, ist W_{opt} durch

$$W_{opt}^T = W_{fld}^T W_{pca}^T \tag{8}$$

gegeben, wobei W_{pca} und W_{fld} definiert sind als

$$W_{pca} = \arg\max_W |W^T S_T W| \tag{9}$$

und

$$W_{fld} = \arg\max_W \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \tag{10}$$

wobei S_T durch

$$S_T = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T \tag{11}$$

gegeben und N die Anzahl der Beispielbilder ist.

Man beachte, dass die Optimierung von W_{pca} über $n \times (N - c)$ Matrizen mit orthonormalen Spalten durchgeführt wird, während die Optimierung für W_{fld} über $(N - c) \times m$ Matrizen mit orthonormalen Spalten durchgeführt wird. Beim Berechnen von W_{pca} werden nur die c-1 kleinsten Hauptkomponenten verworfen.

Sicherlich gibt es noch andere Wege der Reduzierung der Streuung innerhalb der Klassen während man die Streuung zwischen den Klassen beibehält. Zum Beispiel kann man W so wählen, dass man die Streuung zwischen den Klassen der projizierten Beispiele maximiert, nachdem man zuerst die Streuung innerhalb der Klassen vermindert hat. Ein Extremfall wäre es, die Streuung zwischen den Klassen der projizierten Beispiele unter der Voraussetzung zu maximieren, dass die Streuung innerhalb der Klassen Null wäre, das heißt

$$W_{opt} = \arg\max_{w \in W} |W^T S_B W| \tag{12}$$

wobei W eine Menge von $n \times m$ Matrizen mit orthonormalen Spalten ist, die im Kern von S_W enthalten sind.

2.9 Neuronale Netze

Neuronale Netzwerke haben das menschliche Nervensystem als Vorbild. Das Netzwerk ist aus einzelnen Elementen, den so genannten Neuronen, aufgebaut. Diese Neuronen reagieren auf Eingangsimpulse, verarbeiten die Information und können einen Ausgangsimpuls weiterleiten. Das gesamte Netzwerk ist lernfähig, indem Gewichte zwischen den Neuronen verändert werden. Somit kann ein System nach einer Lernphase Gesichter erkennen.

Da es viele verschiedene Ansätze der Gesichtserkennung mithilfe von neuronalen Netzwerken gibt, wird hier nur ein allgemeiner Überblick über die Funktionsweise gegeben. Die kleinste Einheit eines neuronalen Netzes ist ein Neuron (Abbildung 13).

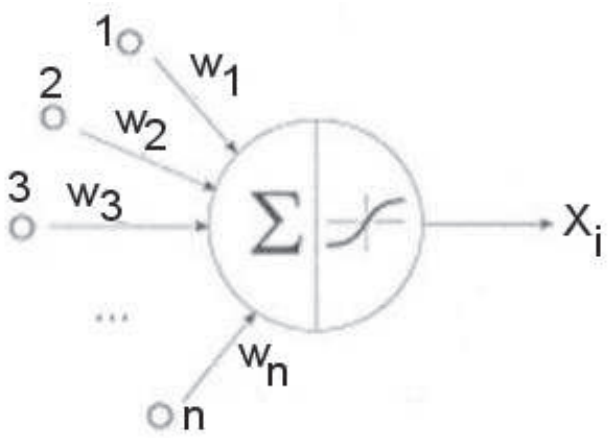

Abbildung 13: **Neuron (Quelle [2])**

Mehrere Neuronen sind in einer Schicht zusammengefasst und mit den Neuronen der nachfolgenden Schicht verbunden, wobei die Verbindungen gewichtet sind (Abbildung 14). Neuronen derselben Schicht sind untereinander nicht verbunden. Die Eingangsimpulse werden nach einer Gewichtung aufsummiert und nach der Addition eines Bias-Wertes einer nichtlinearen Funktion zur Berechnung des Ausgangsimpulses übergeben.

Bevor mit dem eigentlichen Training des Netzes begonnen werden kann, müssen für die Trainingsmenge noch einige Vorbereitungen getroffen werden. Die Gesichter werden auf gleiche Größe und Orientierung normiert und Unterschiede in der Beleuchtung werden herausgerechnet. Nach der Normierung werden die vorhandenen

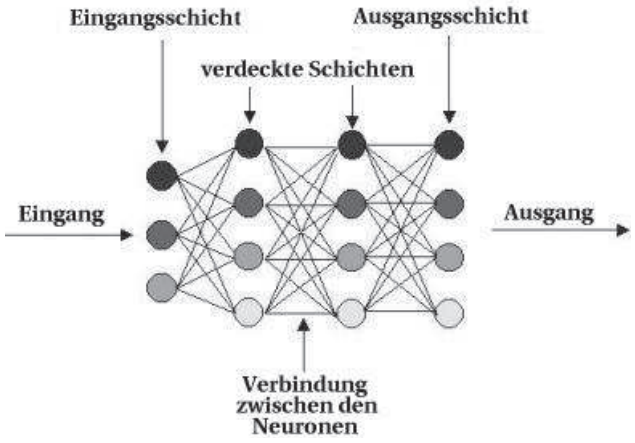

Abbildung 14: **mehrschichtiges neuronales Netzwerk**

Bilder in Trainingsmengen unterteilt. Dies hängt davon ab, wie das neuronale Netz trainiert werden soll. Die Trainingsmenge könnte zum Beispiel in Brillenträger und Nichtbrillenträger unterteilt werden, um das Netz darauf zu trainieren Brillenträger zu erkennen. Für die Identifikation einzelner Personen muss die Trainingsmenge natürlich viel genauer unterteilt werden.

Der eigentliche Lernvorgang des Netzes erfolgt durch den rückgekoppelten Lernalgorithmus. Wird ein Trainingsmuster an die Neuronen der Eingabeschicht angelegt, erfolgt nach der Verarbeitung durch das Netz ein Vergleich des Sollwertes mit dem Istwert. Der Fehler, der aus der Differenz zwischen Soll- und Istwert entsteht, wird dazu benutzt die Gewichte der Verbindungen anzupassen und somit den Fehler zu verkleinern. Man muss darauf achten, dass sich das Netz während des Trainings nicht auf die Trainingsmenge spezialisiert.

Natürlich haben neuronale Netze auch einige Nachteile. Es entsteht ein hoher Trainingsaufwand, bis das Netz soweit ist, bestimmte Gesichter zu erkennen. Außerdem braucht man eine große Anzahl von Bildern für die Trainingsmenge, damit der Erkennungsprozess danach überhaupt zufriedenstellend abläuft.

Desweiteren muss man in der Lernphase aufpassen, dass das neuronale Netz nicht *zu gut* trainiert wird und sich dadurch zu sehr auf bestimmte Muster spezialisiert. Daher werden neuronale Netze häufig anderen Verfahren zur Gesichtserkennung nachgeschaltet. Dies verbessert meist die Ergebnisse, die das vorhergehende Verfahren geliefert hat.

Für die bisher vorgestellten Verfahren ergeben sich somit große Nachteile, die man eben gerne vermeiden würde. Man möchte den Rechenaufwand so gering wie möglich halten, auch bei großen Mengen von Bilder. Die Abhängigkeit von kleinen Änderungen in der Gesichtsmimik oder der Beleuchtung soll möglichst nur einen kleinen oder gar keinen Einfluss auf das Verfahren haben. Dennoch soll natürlich eine niedrige Fehlerquote erreicht werden.

Im nächsten Kapitel wird nun auf die Eigenfacemethode eingegangen, die einen Teil dieser Nachteile reduzieren kann. Änderungen in der Beleuchtung und der Mimik können einfach durch entsprechende Bilder in der Lernmenge reduziert werden. Durch mathematische Umrechnungen kann man auch den Rechenaufwand sehr vermindern und kommt trotzdem zu guten Resultaten.

3. EIGENFACES

3.1 Die Eigenfacemethode

Die meisten bisher bekannten Arbeiten zur automatischen Gesichtserkennung gehen nicht auf die Frage ein, welche Merkmale eines Gesichtes überhaupt zu dessen Erkennung beitragen. Dies ist zugleich auch der wesentliche Unterschied zur Gesichtserkennung mittels Eigenfaces. Bei dieser Methode wird nicht versucht, der intuitiven menschlichen Vorstellung folgend, die markanten Merkmale eines Gesichtes, wie Augen, Nase, Mund, Gesichtsform, sowie deren relative Lage zueinander zu messen und zu vergleichen. Es hat sich gezeigt, dass hierbei wichtige Informationen über die Gesamtkonfiguration eines Gesichtes nicht beachtet werden.

Daher betrachtet man bei der Eigenfacemethode das Gesicht als Ganzes und versucht, die darin erhaltenen Informationen so effizient wie möglich zu kodieren, um sie dann mit in gleicher oder ähnlicher Weise kodierten Mustern vergleichen zu können. Die grundlegende Idee der Methode ist es, die Unterschiede in einer Menge von Bildern, völlig unabhängig von deren individuellen Merkmalen, zu erfassen und mit diesen Informationen einzelne Bilder zu kodieren und zu vergleichen.

Mathematisch ausgedrückt, werden die Hauptbestandteile einer Verteilung (Abbildung 15) von Gesichtern oder die Eigenvektoren der Kovarianzmatrix einer Menge von Gesichtsbildern berechnet, indem man ein Bild als einen Punkt (oder als einen Vektor) in einem sehr hochdimensionalen Raum behandelt. Diese Verteilung oder Menge von Gesichtern nennt man *Trainingsmenge* oder *Lernmenge*. Die erhaltenen Eigenvektoren sind geordnet. Man kann sich die Eigenvektoren als markante Merkmale vorstellen, die die Variationen zwischen den Gesichtern charakterisieren. Somit ist es das Ziel der Eigenfacemethode, jedes Gesicht als eine Linearkombination dieser Eigenvektoren darzustellen.

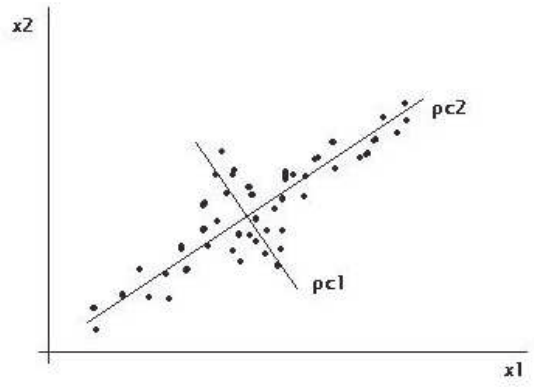

Abbildung 15: **Hauptkomponenten pc1 und pc2 einer Menge**

Jede Bildposition steuert mehr oder weniger zu jedem Eigenvektor bei, wodurch man einen Eigenvektor als eine Art Geistergesicht beschreiben kann, den man *Eigenface* oder *Eigengesicht* nennt. Eine Lernmenge, mit der hier gearbeitet wurde, ist in Abbildung 18

dargestellt. Die dazugehörigen Eigenfaces sind in Abbildung 20 zu sehen.

Wie bereits erwähnt, kann jedes einzelne Gesicht genau durch eine Linearkombination der Eigenfaces dargestellt werden. Jedes Gesicht kann außerdem angenähert werden, indem man nur die "besten" Eigenfaces verwendet. Die "besten" Eigenfaces sind diejenigen mit den größten Eigenwerten, die für die größten Unterschiede zwischen den Bilder in der Trainingsmenge stehen. Die besten M Eigenfaces spannen den M-dimensionalen Unterraum aller Gesichter auf, der "face space" genannt wird. Diesen "face space" gilt es somit, anhand der Hauptkomponentenanalyse, zu ermitteln.

Die Idee Eigenfaces zu verwenden wurde durch Arbeiten von Sirovich und Kirby [22] und Kirby und Sirovich [14] angeregt, die sich die Technik der Hauptkomponentenanalyse oder auch *principal component analysis* (PCA, siehe Kapitel 3.2) zu Nutzen machten, um Bilder von Gesichtern äußerst effizient zu kodieren. Beginnend mit einer Zusammenstellung von Trainingsbildern berechneten sie das beste Koordinatensystem für die Bildkompression, wobei jede Koordinate eigentlich ein Bild repräsentiert, das sie *Eigenpicture* nannten. Sie behaupteten, dass man jede Sammlung von Gesichtern annähernd rekonstruieren kann, indem man eine kleine Anzahl von Gewichten für jedes Gesicht und eine kleine Menge von Standardbildern speichert. Die Gewichte die jedes Gesicht beschreiben bekommt man, indem man das Gesicht auf jedes Eigenpicture projiziert.

Somit ergeben sich für diesen Ansatz der Gesichtserkennung die folgenden initialisierenden Arbeitsschritte:

1. Ermittlung einer Anfangsmenge von Gesichtsbildern (der Trainingsmenge).
2. Berechnung der Eigenfaces aus der Trainingsmenge. Es werden nur die M besten Bilder, die den höchsten Eigenwerten entsprechen, behalten. Diese M Bilder definieren den "face space". Wenn neue Bilder hinzukommen, können die Eigenfaces aktualisiert oder neu berechnet werden.
3. Berechnung der entsprechenden Verteilung im M-dimensionalen Gewichtsraum für jedes bekannte Individuum, indem man das Gesicht in den "face space" projiziert.

Erhöht sich die Anzahl der Bilder der Trainingsmenge können die Arbeitsschritte bei freier Rechnerkapazität auch von Zeit zu Zeit wiederholt werden.

Wenn das System initialisiert ist, werden folgende Schritte durchgeführt, um neue Gesichter zu erkennen:

1. Berechne eine Menge von Gewichten, die auf dem Eingabebild und den M Eigenfaces beruhen, indem man das Eingabebild auf jedes Eigenface projiziert.
2. Bestimme, ob das Bild überhaupt ein Gesicht ist. Hierzu überprüft man, ob das Bild dem "face space" ausreichend nahe ist.
3. Handelt es sich bei dem Eingabebild um ein Gesicht, so folgt eine Klassifikation des Gewichtsmusters nach bekannt oder unbekannt.
4. (Optional) Die Eigenfaces und/oder die Gewichtsmuster können aktualisiert werden.
5. (Optional) Wenn dasselbe unbekannte Gesicht mehrere Male vorkommt, dann berechne sein charakteristisches Gewichtsmuster und nehme es in die bekannten Gesichter mit auf.

3.2 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse ist eine klassische statistische Methode. Diese lineare Transformation wird häufig in der Datenanalyse und der Datenkompression verwendet. Die Hauptkomponentenanalyse basiert auf der statistischen Darstellung einer Zufallsvariablen. Angenommen, man hat eine Menge von zufälligen Vektoren x, wobei

$$x = (x_1, ..., x_n)^T \tag{13}$$

und der Durchschnitt der Menge durch

$$\mu_x = E\{x\} \tag{14}$$

und die Kovarianzmatrix der Daten durch

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\} \tag{15}$$

gegeben ist.

Die Komponenten von C_x, die durch c_{ij} gegeben sind, stellen die Kovarianzen zwischen den zufälligen Bestandteilen der Variablen x_i und x_j dar. Der Bestandteil c_{ii} ist die Varianz von x_i. Die Varianz einer Komponenten ist ein Anzeichen für die Verteilung der Werte der Komponente um seinen Mittelwert. Wenn zwei Bestandteile x_i und x_j der Daten unkorreliert sind, dann ist ihre Kovarianz Null ($c_{ij} = c_{ji} = 0$). Die Kovarianzmatrix ist per Definition immer symmetrisch.

Von einer Menge von Vektoren $x_1, ..., x_M$ kann man den Durchschnitt der Menge und die Kovarianzmatrix berechnen. Für eine symmetrische Matrix, wie die Kovarianzmatrix eine ist, kann man eine orthogonale Basis berechnen, indem man ihre Eigenwerte und Eigenvektoren berechnen. Die Eigenvektoren e_i und die zugehörigen Eigenwerte λ_i sind die Lösung der Gleichung

$$C_x e_i = \lambda_i e_i \quad i = 1, ..., n \tag{16}$$

Der Einfachheit halber nimmt man an, dass die Werte λ_i verschieden sind. Diese Werte können zum Beispiel dadurch gefunden werden, indem man die Lösung der charakteristischen Gleichung

$$\mid C_x - \lambda I \mid = 0 \tag{17}$$

findet, wobei I die Einheitsmatrix ist, die dieselbe Ordnung hat wie C_x und $\mid \ ... \ \mid$ für die Determinante der Matrix steht. Wenn der

Vektor der Daten n Elemente hat, dann wird die charakteristische Gleichung auch der Ordnung n sein. Das ist nur einfach zu lösen, wenn n genügend klein bleibt. Ansonsten muss man zu mathematischen Tricks wie in den Gleichungen 25 und 26 greifen.

Angenommen, man hat den Durchschnitt und die Kovarianzmatrix einer Menge berechnet, und A sei eine Matrix, deren Zeilen die geordneten Eigenvektoren der Kovarianzmatrix darstellen, so kann man den Vektor x durch

$$y = A(x - \mu_x) \tag{18}$$

in den Vektor y transformieren. y ist ein Punkt im orthogonalen Koordinatensystem, der durch die Eigenvektoren bestimmt ist. Die Elemente von y können als die Koordinaten in der orthogonalen Basis verstanden werden. Man kann den originalen Datenvektor x aus y durch

$$x = A^T y + \mu_x \tag{19}$$

rekonstruieren. A^T ist die transponierte Matrix A.

Zur Rekonstruktion muss man aber nicht alle Eigenvektoren verwenden. Sei A_K eine Matrix, die aus den ersten K Eigenvektoren besteht. Somit kann man eine ähnliche Transformation wie oben mit

$$y' = A_K(x - \mu_x) \tag{20}$$

und

$$x' = A_K^T y + \mu_x \tag{21}$$

erzeugen, das heißt, x und y können mit nur wenigen Eigenfaces durch x' und y' approximiert werden.

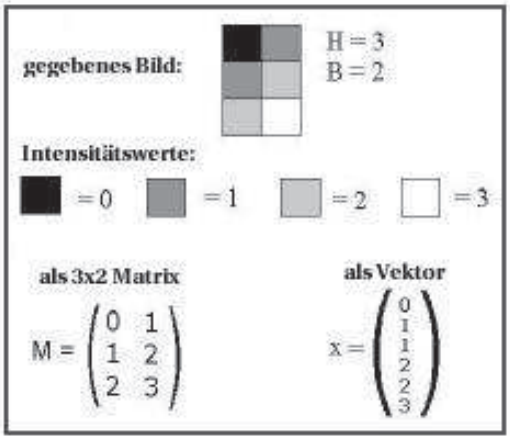

Abbildung 16: **Darstellung eines Bildes der Größe 3×2 als Vektor der Größe 3*2 (Quelle [24])**

3.3 Berechnung der Eigenfaces

Im Folgenden kann man sich ein Bild als einen Vektor vorstellen. Sei also ein Gesichtsbild I(x,y) ein zweidimensionales NxN Array mit 8-Bit Helligkeitswerten. Das Bild wird auch als Vektor der Dimension der Anzahl der Pixel, nämlich N^2, betrachtet (Abbildung 16). Jedes Pixel des Bildes repräsentiert somit eine Komponente des Vektors. Ein typisches Bild der Größe 256×256 stellt daher einen Vektor der Dimension 65.536 dar, oder anders ausgedrückt, ein Punkt im 65.536 - dimensionalen Raum.

Da sich die Bilder von Gesichtern im Großen und Ganzen ähneln, kommt man bei der Darstellung der Vektoren als Punkte in diesem hochdimensionalen Raum zu dem Ergebnis, dass die Punkte nicht beliebig im ganzen Raum verteilt sind. Sie konzentrieren sich an einer Stelle und bilden eine Wolke (Abbildung 17) und können somit anhand eines relativ kleinen Unterraumes, des Gesichtsraumes (*face space*), beschrieben werden.

Die eigentliche Idee der Hauptkomponentenanalyse (oder Karhunen-Loeve - Entwicklung) ist es, diejenigen Vektoren zu finden, die

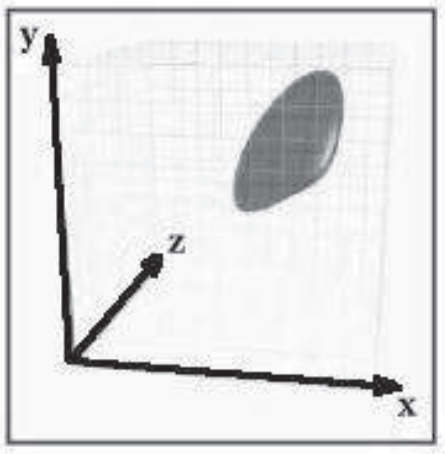

Abbildung 17: **Punktwolke im dreidimensionalen Raum (Quelle [24])**

die Verteilung der Gesichtsbilder im ganzen Bildraum am Besten beschreiben. Diese Vektoren definieren eben jenen Unterraum der Gesichtsbilder, den man *face space* nennt. Jeder Vektor der Länge N^2 beschreibt ein NxN Bild, und ist eine Linearkombination der originalen Gesichtsbilder. Da diese Vektoren die Eigenvektoren der Kovarianzmatrix sind, die zum originalen Bild gehört, und weil sie eine gesichtsähnliche Erscheinung haben, werden sie *Eigenfaces* oder *Eigengesichter* genannt. Sie sind die Basisvektoren des Gesichtsraumes.

Die Berechnung für Farbbilder würde sich wesentlich schwieriger gestalten, daher wird mit Graustufenbildern im ".pgm"-Format gearbeitet. Bevor die Berechnung des Durchschnittsbildes, der Differenzbilder und der Eigengesichter beginnen kann, ist es notwendig, dass alle Bilder eine einheitliche Größe besitzen. Daher werden die Bilder der Lernmenge, soweit nötig, vor der eigentlichen Berechnung in Graustufenbilder umgewandelt und auf eine einheitliche Größe skaliert.

Sei die Trainingsmenge der Gesichtsbilder $\Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_M$. Das Durchschnittsbild der Menge der Bilder ist definiert als:

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n$$

.

Abbildung 18: **Teil einer Lernmenge**

Aus dem Durchschnittsbild und den Bildern der Lernmenge lassen sich nun die Differenzbilder $\Phi_i = \Gamma_i - \Psi$ berechnen. Ein Beispiel für eine Trainingsmenge ist in Abbildung 18 dargestellt, ein Beispiel für ein Durchschnittsbild in Abbildung 19.

Diese Menge von sehr großen Vektoren unterliegt nun der Hauptkomponentenanalyse, die eine Menge von M orthogonalen Vektoren μ_n sucht, die die Verteilung der Daten am Besten beschreibt. Der k-te Vektor μ_k wird so gewählt, dass

$$\lambda_k = \frac{1}{M} \sum_{n=1}^{M} (\mu_k^T \Phi_n)^2 \tag{22}$$

ein Maximum gemäß

$$\mu_l^T \mu_k = \delta_l k = \left\{ \begin{array}{cc} 1, & wenn\ l = k0, \\ & sonst \end{array} \right\} \tag{23}$$

ist.

Die Vektoren μ_k und die Skalare λ_k sind die Eigenvektoren, beziehungsweise die Eigenwerte der Kovarianzmatrix

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \tag{24}$$

wobei die Matrix A durch $A = [\Phi_1 \Phi_2 ... \Phi_m]$ definiert ist. Die Kovarianzmatrix C ist eine (symmetrische) $N^2 \times N^2$ Matrix. Die Bestimmung der N^2 Eigenvektoren und Eigenwerte ist für normale Bildgrößen eine fast unlösbare Aufgabe. Bei einer Bildgröße von 256×256 Pixeln wären wohl selbst moderne Computer nicht sonderlich effizient bei dem Versuch eine 65.536×65.536 Matrix zu diagonalisieren und ihre 65.536 Eigenvektoren zu berechnen. Daher braucht man eine effizientere Methode, um die Eigenvektoren zu bestimmen.

Dies gelingt durch die Reduktion der $N^2 \times N^2$ Matrix auf eine $M \times M$ Matrix, wobei M die Anzahl der Bilder in der Lernmenge darstellt. Wenn nämlich die Anzahl der Datenpunkte im Bildraum kleiner als die Dimension des Raumes ist (M<N^2), dann gibt es anstatt N^2 nur M-1 aussagekräftige Eigenvektoren (die übrigen Eigenvektoren haben zugehörige Eigenwerte von Null). Um die N^2 Eigenvektoren zu bestimmen, kann man somit erst die Eigenvektoren der MxM Matrix berechnen, und danach die dazugehörigen Linearkombinationen der Gesichtsbilder Φ_i verwenden. Gegeben seien die Eigenvektoren ν_i von $A^T A$, so dass

$$A^T A \nu_i = \mu_i \nu_i \tag{25}$$

Multipliziert man beide Seiten mit A, so erhält man

$$AA^T A \nu_i = \mu_i A \nu_i \tag{26}$$

woraus sich ergibt, dass $A\nu_i$ die Eigenvektoren der Matrix $C = AA^T$ sind (man beachte die Reihenfolge der Faktoren). Wichtig ist hierbei, dass man die Eigenvektoren ν_i so normalisiert, dass $\|\nu_i\| = 1$ ist.

Folgt man dieser Analyse, so konstruiert man die MxM Matrix $L = A^T A$, wobei $L_{mn} = \Phi_m^T \Phi_n$ ist, und berechnet die Eigenvektoren

Abbildung 19: **Durchschnittsbild der Lernmenge in Abbildung 18**

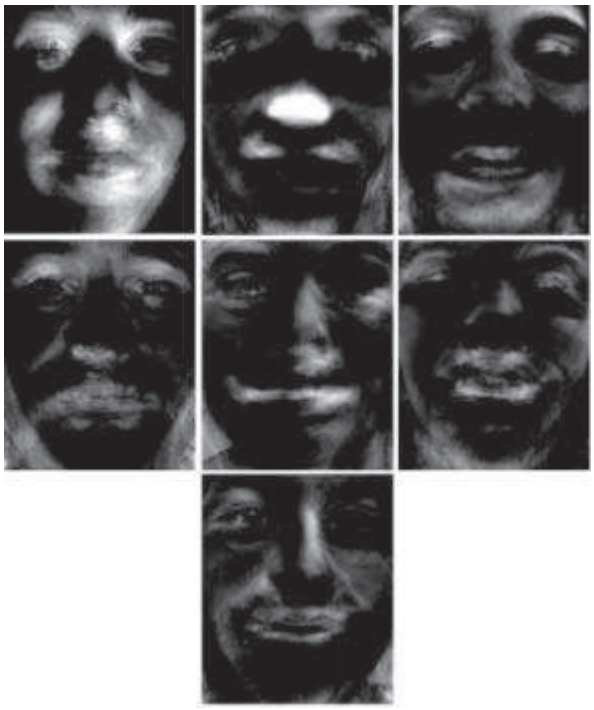

Abbildung 20: **Die sieben besten Eigenfaces der Lernmenge in Abbildung 18**

ν_n von L. Diese Vektoren bestimmen Linearkombinationen der M Lernmengenbilder, um die EigenFaces μ_i darzustellen.

$$\mu_l = \sum_{k=1}^{M} \nu_{lk} \Phi_k = A\nu_l \quad l = 1, ..., M \tag{27}$$

Mit diesem Verfahren werden die Berechnungen von der Anzahl der Pixel in einem Bild (N^2) auf die Anzahl der Bilder in der Trainingsmenge (M) sehr stark reduziert. In der Praxis wird die Menge der Trainingsbilder sehr viel kleiner sein als das Quadrat der Anzahl der Pixel im Bild ($M \ll N^2$) und die Berechnung wird durchführbar. Die assoziierten Eigenwerte erlauben es, die Eigenvektoren bezüglich ihres Nutzen für die Charakterisierung der Unterschiede der Bilder einzuordnen. Abbildung 20 zeigt die besten sieben Eigenfaces die von den Eingabebildern abgeleitet wurden.

3.4 Klassifikation der Eigenvektoren

Nach der Berechnung der Eigenvektoren stellt sich natürlich die Frage, welche der berechneten Eigenvektoren diejenigen sind, die man am besten verwendet. Turk und Pentland [26] haben behauptet, dass ganz einfach die Eigenvektoren mit den größten dazugehörigen Eigenwerten die besten seien, da sie am aussagekräftigsten bezüglich der Gesichterverteilung im Raum sind. Hierzu gibt es jedoch eine Untersuchung von O'Toole et al aus dem Jahre 1993 [9].

Der Nutzen der Eigenvektoren ist abhängig von ihrem Zweck. Will man ein Gesicht nur als bekannt oder unbekannt klassifizieren, so ist die von Pent und Turkland getroffene Wahl nicht unbedingt die effektivste. Im Folgenden werden nur kurz die Vorgehensweise und die Ergebnisse von O'Toole beschrieben. Ausführlich sind die Ergebnisse in [9] nachzulesen.

Aus 159 Gesichtsbildern mit 16 Graustufen wurden zufällig 100 Bilder für die Lernmenge herausgesucht. Alle 159 Bilder wurden dann einer Erkennungsprüfung unterzogen. Im Idealfall werden also 100 Personen korrekt erkannt, die restlichen 59 werden als "unbekannt" klassifiziert.

Ziel der Analyse war es, die Nützlichkeit der Eigenvektoren unter zwei verschiedenen Aspekten zu ermitteln. Einmal bezüglich der Qualität der Rekonstruktion eines Bildes anhand der Eigenvektoren ("physische" Ähnlichkeit mit dem Originalbild), andererseits bezüglich der reinen Erkennunsleistung (Einstufung als bekannt oder unbekannt).

Das Ergebnis lautet zusammenfassend, dass sich zur reinen Rekonstruktion eines Bildes die Eigenvektoren mit den größten zugehörigen Eigenwerten am besten eignen. Dies stimmt auch mit der Vorgehensweise der Hauptkomponentenanalyse überein, da genau diese Eigenvektoren für die größten Unterschiede sorgen und sich sehr gut eignen, die Gemeinsamkeiten der vielen Gesichter zu speichern.

Die Eigenvektoren mit kleineren zugehörigen Eigenwerten beinhalten dagegen mehr Informationen über individuelle Gesichter und tragen somit am meisten zur Erkennung eines Gesichtes bei. O'Toole haben die beste Erkennungsleistung bei Eigenvektoren mit kleineren Eigenwerten bekommen (im Beispiel bei den Eigenvektoren 45-80).

Abschließend läßt sich noch sagen, dass sich eine optimale Aus-

wahl an Eigenvektoren für die Gesichtsidentifikation nicht genau definieren läßt. Um zu bestimmen, ob es sich bei einem Eingabebild um ein Gesicht handelt, kann angeben werden, wieviele Eigenvektoren verwendet werden sollen. Für die Gesichtserkennung jedoch werden alle berechneten Eigenvektoren verwendet, um somit das bestmögliche Ergebnis zu erreichen.

3.5 Klassifikation von Gesichtsbildern

Die Eigenfaces, die aus den Eigenvektoren von L berechnet wurden, spannen eine Basismenge auf, mit der man Gesichtsbilder beschreiben kann. Sirovich und Kirby [22] haben eine eingeschränkte Version dieser Rahmenbedingungen mit M=115 Bildern ausgewertet, die unter kontrollierten Umständen digitalisiert wurden. Sie haben herausgefunden, dass ungefähr 40 Eigenfaces ausreichend waren, um eine sehr gute Beschreibung der Menge der Gesichtsbilder zu bekommen. Die Eigenfaces spannen einen M′-dimensionalen Unterraum des originalen N^2 Bildraumes auf. Als die M′ wichtigsten Eigenvektoren der Matrix L werden diejenigen gewählt, die die größten Eigenwerte haben.

Ein neues Gesichtsbild (Γ) wird durch eine einfache Berechnung in seine Eigenfacekomponenten überführt (in den "face space" projiziert).

$$\omega_k = \mu_k^T(\Gamma - \Psi) \quad k = 1, ..., M' \tag{28}$$

Abbildung 21 zeigt ein Originalbild und seine Projektionen mit jeweils 26, 52 und 78 (in diesem Falle allen) Eigenvektoren in den "face space".

Abbildung 21: **Links: Originalbild und Projektionen in den "face space" mit 26, 52 und 78 Eigenvektoren**

Natürlich kann man aber nicht nur komplette Gesichter mithilfe weniger Eigenvektoren in den "face space" projizieren. Es ist ebenfalls möglich, teilweise verdeckte Gesichter auf diese Weise vollständig zu rekonstruieren (Abbildung 22).

Die Gewichte ω_k bilden einen Vektor $\Omega^T = [\omega_1, \omega_2, ..., \omega_{M'}]$ dessen einzelne Komponenten ω_i den Koeffizienten der Darstellung des Differenzbildes ($\Gamma - \Psi$) als Linearkombination von Eigengesichtern entsprechen. Somit werden die Eigenfaces als eine Basismenge für Gesichtsbilder behandelt. Dieser Vektor kann nun in einem standardisierten Mustererkennungsalgorithmus dazu verwendet werden, um aus einer Menge von vordefinierten Gesichtsklassen, diejenige herauszufinden, die das Gesicht am Besten beschreibt. Die einfachste Methode um festzustellen, ist es, die Gesichtsklasse k zu finden, die den euklidischen Abstand

$$\epsilon_k^2 = \|\Omega - \Omega_k\|^2 \tag{29}$$

Abbildung 22: **Links: Verdecktes Gesicht, Mitte: Originalbild, Rechts: Projektionen in den "face space"**

minimiert, wobei Ω_k ein Vektor ist, der die k-te Gesichtsklasse beschreibt. Die Gesichtsklassen Ω_i werden berechnet, indem man die Ergebnisse der Eigenfacedarstellungen über eine kleine Anzahl von Bildern (möglich ist auch nur ein Bild) mittelt. Ein Gesicht wird zur Klasse k zugehörig gewertet, wenn das Minimum von ϵ_k unter einem gewählten Schwellenwert Θ_k liegt. Ansonsten wird das Gesicht als "unbekannt" klassifiziert und kann verwendet werden, um eine neue Gesichtsklasse anzulegen. Als Schwellwert ist zum Beispiel ein Wert von

$$\frac{1}{2} max \parallel \Omega_i - \Omega_j \parallel \quad i, j = 1, ..., M \tag{30}$$

möglich, wobei dies natürlich nur eine Verallgemeinerung ist. Es wäre besser, für jede einzelne Trainingsmenge einen eigenen Threshold zu finden, der die Gesichtserkennung optimiert.

Weil die Berechnung des Gewichtsvektors equivalent zur Projektion des Originalgesichtes in den niedrigdimensionalen "face space" ist, werden viele Bilder (die meisten von ihnen sehen nicht wie Gesichter aus) auf einen gegebenen Mustervektor projiziert. Dies ist jedoch kein Problem, da die Entfernung ϵ zwischen dem Bild und dem "face space" einfach die quadrierte Entfernung zwischen dem, um das Durchschnittsbild reduzierte, Eingabebild $\Phi = \Gamma - \Psi$ und seiner Projektion in den "face space", $\Phi_f = \sum_{i=1}^{M'} \omega_i \mu_i$ ist (Abbildung 23):

$$\epsilon^2 = \|\Phi - \Phi_f\|^2 \quad (31)$$

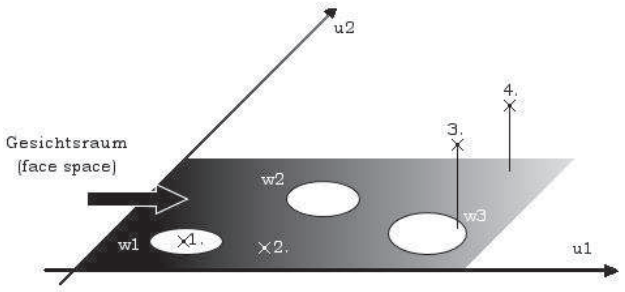

Abbildung 23: Der "face space" und die Gesichtsklassen w1-w3 (zweidimensionaler Fall)

Somit gibt es vier Möglichkeiten für das Eingabebild und seinen Mustervektor:

1. Unmittelbare Nähe zum Gesichtraum sowie zu einer Gesichtsklasse
   Konsequenz: Das Bild wird als bekanntes Gesicht identifiziert.

2. Unmittelbare Nähe zum Gesichtsraum, jedoch nicht zu einer Gesichtsklasse
   Konsequenz: Das Bild wird als unbekanntes Gesicht identifiziert.

3. Fern ab vom Gesichtsraum, jedoch nahe einer Gesichtsklasse.
   Konsequenz: Während die meisten Systeme hier ein "Falsepositive" - Ergebnis melden würden, wird das Eigengesicht - Verfahren aufgrund der deutlichen Entfernung zum Gesichtsraum zum (einzig richtigen) Ergebnis kommen, dass das Bild kein Gesicht darstellt.

4. Fern ab vom Gesichtsraum und von jeder Gesichtsklasse
   Konsequenz: Gleiches Ergebnis wie in 3, das heißt das Bild stellt kein Gesicht dar.

3.6 Zusammenfassung der Eigenfacemethode

Zusammenfassend beinhaltet der Eigenfaceansatz zur Gesichtserkennung die folgenden Schritte:

1. Erstelle eine Menge von charakteristischen Gesichtsbildern von bekannten Individuen. Diese Menge sollte für jede Person eine Anzahl von Variationen im Ausdruck und in der Lichtgebung und eventuell auch in der Neigung des Kopfes beinhalten (Variationsmöglichkeiten sind z.B. die Beleuchtung von vorne, links oder rechts, die Neigung des Kopfes (0° , 45°) oder der Gesichtsausdruck einer Person (lachen, weinen, normal schauen)). Je mehr Variationen in der Trainingsmenge vorhanden sind, desto besser funktioniert später die Erkennung unter variierenden Umständen. Die gesamte Anzahl der Bilder der Trainingsmene ist M.

2. Berechne die (MxM) Matrix L, finde ihre Eigenvektoren und Eigenwerte und wähle die M' Eigenvektoren mit den höchsten zugehörigen Eigenwerten aus.

3. Kombiniere die normalisierte Trainingsmenge der Bilder entsprechend Formel (27) und berechne die Eigenfaces μ_k.

4. Berechne für jede bekannte Person den Klassenvektor Ω_k, indem man die Eigenface-Mustervektoren Ω (aus der Gleichung 29), die aus den originalen Bildern der Person berechnet wurden, mittelt. Wähle einen Schwellenwert Θ_ϵ, der die maximale Entfernung vom face space (nach Gleichung 31) definiert.

5. Berechne für jedes neue Gesichtsbild, das identifiziert werden soll, den zugehörigen Mustervektor Ω, die Entfernung ϵ_k zu jeder bekannten Klasse und die Entfernung ϵ zum face space. Wenn die kleinste Entfernung $\epsilon_k < \Theta_\epsilon$ ist, und die Entfernung $\epsilon < \Theta_\epsilon$, dann klassifiziere das Eingabebild als eine Person, die mit der Vektorklasse Ω_k in Verbindung gebracht wird. Wenn die kleinste Entfernung $\epsilon_k > \Theta_\epsilon$ ist, aber die Entfernung $\epsilon < \Theta_\epsilon$ ist, dann kann man das Gesicht als "unbekannt" klassifizieren und optional, eine neue Gesichtsklasse anlegen.

6. Wenn das neue Bild als bekannte Person klassifiziert worden ist, dann kann dieses Bild zu der Originalmenge der bekannten Bilder hinzugefügt werden und die Eigenfaces können neu berechnet werden (Schritte 1-4).

4. GESICHTSDETEKTION

4.1 Detektion und Erkennung von Gesichtern

Für die Analysen im vorhergehenden Kapitel wurde angenommen, dass als Eingabebild ein zentriertes Gesicht vorliegt, das dieselbe Größe hat, wie die Bilder der Lernmenge und die Eigenfaces. Dafür ist es zunächst notwendig ein Gesicht in einer Szene zu lokalisieren, um anschließend die eigentliche Erkennung durchzuführen. Prinzipiell gibt es mehrere Möglichkeiten ein Gesicht in einem Bild zu finden. Drei verschiedene Verfahren werden im Folgenden kurz vorgestellt. Das erste ist sehr einfach und relativ effizient. Hierbei handelt es sich um einfaches *Motion-Tracking* (Kapitel 4.1.1). Beim zweiten Verfahren wird eine *neuronales Netz* zum Auffinden der Gesichter verwendet (Kapitel 4.1.2). Die dritte Möglichkeit, Gesichter in Bildern zu finden ist, obwohl sie nicht verwendet wurde, ist im Zusammenhang mit den Eigenfaces die Wichtigste. Es wird hierbei auf den schon bekannten *face space* zurückgegriffen (Kapitel 4.1.3).

4.1.1 Bewegungserkennung und Tracking des Kopfes

Personen bewegen sind die ganze Zeit. Sogar wenn man sitzt, zappelt man, ändert seine Körperstellung, nickt mit dem Kopf und dergleichen. Im Falle einer einzelnen Person, die sich in einer statischen Umgebung bewegt kann ein einfacher Algorithmus für Bewegungserkennung und Verfolgung, wie er in Abbildung 24 dargestellt ist, die Position des Kopfes finden und verfolgen.

Einfaches räumliches und zeitliches Filtern (zum Beispiel Frame Unterschiede) hebt Bildregionen heraus, die sich mit der Zeit verändern.

Die Pixel im gefilterten Bild werden mit einem Schwellwert verglichen. Um ein binäres Bewegungsbild zu erhalten, werden die Regionen mit hoher Bewegungsdichte über die Zeit untersucht, um zu entscheiden, ob die Bewegung durch eine Person verursacht wurde,

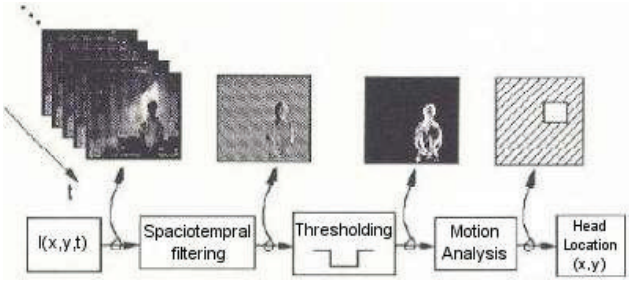

Abbildung 24: **Ein System zur Erkennung und Verfolgung des Kopfes. (Quelle [25])**

und um die Position des Kopfes zu bestimmen. Es werden einige einfache Regeln angewendet, zum Beispiel ist der Kopf der kleinere Bereich über dem größeren Bereich (dem Körper) und die Kopfbewegung muss einigermaßen langsam und zusammenhängend sein (von Köpfen erwarten man nicht, dass sie ziellos im Bild herumspringen). Abbildung 25 zeigt ein Bild, in dem der Kopf gefunden wurde. Zusätzlich ist der Bewegungspfad des Kopfes in der vorausgegangenen Abfolge von Frames eingetragen.

Abbildung 25: **Der Kopf wurde entdeckt - Das Bild im Rahmen wird zur Gesichtserkennung weiterverwendet. Die weiße Linie zeigt den Bewegungspfad des Kopfes über einige vorhergehende Frames an (Quelle [25]).**

Das Bewegungsbild lässt zusätzlich noch eine Schätzung der Skalierung zu. Die Größe der Region, von der angenommen wird, dass es sich um einen Kopf handelt, bestimmt die Größe des Unterbildes, mit dem bei der Erkennung weitergearbeitet wird. Dieses Unterbild wird skaliert, um zur Größe der Eigenfaces zu passen.

4.1.2 Finden eines Gesichtes mithilfe eines neuronalen Netzes

Diese Methode arbeitet im Wesentlichen so, wie schon in Kapitel 2.9 beschrieben. Der Algorithmus basiert auf der Arbeit von [19], die bereits im November 1995 ein System zur Gesichtsentdeckung vorstellten, das bei einer Entdeckungsrate von 90,5% nur wenige irrtümliche Gesichter erkannte. Da das Netzwerk mit Einzelbildern trainiert wird und auch nur Gesichter in Einzelbildern finden soll, muss für die Verarbeitung von Videos ein kleiner Anpassungsschritt unternommen werden. Es wird kein komplettes Video betrachtet, sondern die einzelnen Frames eines Videos werden separat der Analyse unterzogen.

Gibt es ein gewisses Vorwissen über eventuell im Bild enthaltene Gesichter, so kann der Algorithmus signifikant beschleunigt werden.Sollen im Cideo beispielsweise nur große oder nur kleine Gesichter gefunden werden, so kann der Algorithmus das Bild entsprechend vorskalieren. Ebenso ist es möglich, bei Farbbildern oder Filmen mögliche Gesichtsregionen zu lokalisieren und die Suche des neuronalen Netzes auf diese Regionen zu beschränken. Dadurch ergibt sich natürlich mit genügend Vorwissen auch einen großen Zeitvorteil.

Um mit dem neuronalen Netz zu arbeiten, muss es erst mit Gesichtsbildern trainiert werden. Hierfür sind für die Trainingsmenge noch ein paar Anpassungsschritte vorzunehmen. Die Bilder der Lernmenge werden, wenn nötig gedreht, so dass in der Trainingsmenge nur aufrechte Bilder vorhanden sind. Danach werden die Bilder einer, wie in Kapitel 4.3.3 beschriebenen, Belichtungskorrektur unterzogen. Sobald diese Schritte abgeschlossen sind, kann das Training beginnen und man kann das neuronale Netz auf seine Belange spezialisieren. Eine genauere Beschreibung der Algorithmen kann man in [16] und [19] nachlesen.

4.1.3 Finden eines Gesichtes mithilfe des ıface spacei

Man kann das Wissen über den face space dazu verwenden, Gesichter in einzelnen Bildern zu entdecken. Entweder als Alternative zur Lokalisierung von Gesichtern durch Bewegung (zum Beispiel, wenn es zu wenig Bewegung oder zu viele sich bewegende Objekte gibt) oder um eine höhere Präzision als durch den Gebrauch der Bewegungsanalyse zu erreichen. Diese Methode erlaubt es, die Anwesenheit von Gesichtern unabhängig von der Aufgabe der Identifizierung zu erkennen.

Wie man in Abbildung 26 sehen kann, verändern sich die Gesichter nicht allzu stark, wenn sie in den face space projiziert werden, während sich Bilder, die kein Gesicht enthalten, sehr stark unterscheiden. Diese grundlegende Idee wird verwendet, um das Vorhandensein eines Gesichtes in einer Szene zu bestätigen: Berechne für jede Stelle im Bild die Entfernung ϵ zwischen dem lokalen Unterbild und dem face space. Die Entfernung vom face space wird als Maß der “Gesichtsähnlichkeit“ genommen. Somit ist das Ergebnis der Berechnung der Entfernung vom face space für jeden Punkt im Bild eine “face map“ $\epsilon(x, y)$. Abbildung 27 zeigt ein Bild und seine Gesichtskarte - niedere Werte (die dunklen Gebiete) lassen das Vorhandensein eines Gesichtes vermuten. Natürlich muss das Gesicht auf dem zu untersuchenden Bild die gleiche Größe haben, wie die Gesichter der Lernmenge, oder es muss entsprechend skaliert werden.

Unglücklicherweise ist die direkte Anwendung von Formel 31 rechnerisch sehr aufwendig. Daher wurde eine einfachere, effizientere Methode entwickelt, um die Gesichtskarte $\epsilon(x, y)$ zu berechnen, die im Folgenden beschrieben wird.

Um die Gesichtskarte für jedes Pixel I(x,y) des Bildes zu berechnen, müssen wir das Unterbild, das auf diesem Pixel zentriert ist auf den face space projizieren. Danach muss man die Projektion vom Original abziehen. Um das Unterbild Γ auf den face space zu projizieren, muss man zuerst das Durchschnittsbild abziehen, woraus

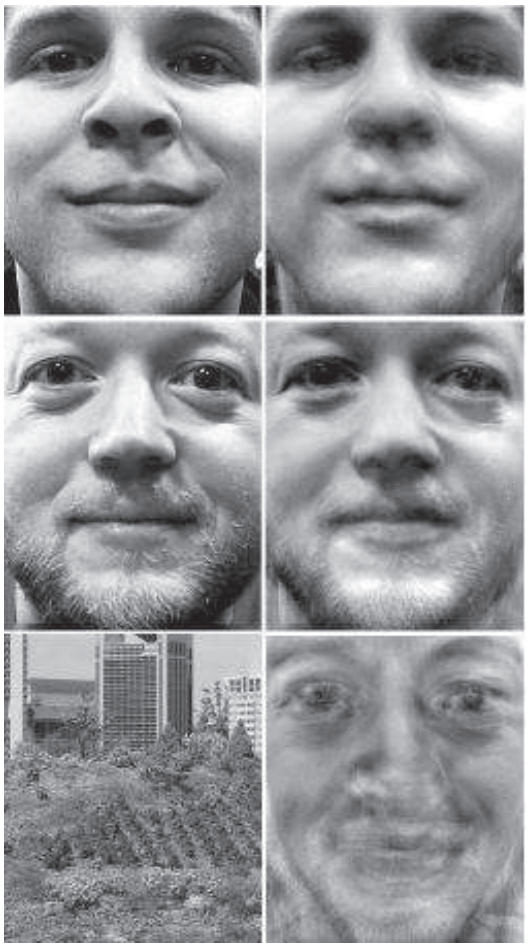

Abbildung 26: **Drei Bilder (links) und ihre Projektion mit 26 von 78 Eigenvektoren in den "face space" (rechts). Das relative Maß der Entfernung vom face space ist für a.) 5556 b.) 5701 und c.) 14271.20 Die Bilder a.) und b.) waren in der ursprünglichen Lernmenge enthalten**

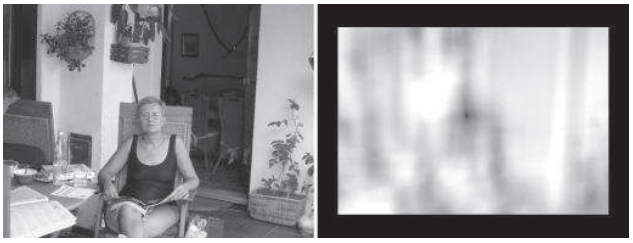

Abbildung 27: **Links: Originalbild rechts: Die entsprechende Gesichtskarte**

sich dann $\Phi = \Gamma - \Psi$ ergibt. Wenn Φ_f die Projektion von Φ auf den face space ist, dann ist das Entfernungsmaß an einer gegebenen Stelle durch

$$\begin{aligned}\epsilon^2 &= \|\Phi - \Phi_f\|^2 \\ &= (\Phi - \Phi_f)^T(\Phi - \Phi_f) \\ &= \Phi^T\Phi - \Phi^T\Phi_f - \Phi_f^T(\Phi - \Phi_f) \\ &= \Phi^T\Phi - \Phi_f^T\Phi_f\end{aligned} \tag{32}$$

definiert, wobei $\Phi_f \perp (\Phi - \Phi_f)$ ist. Hierbei ist Φ_f eine Linearkombination der Eigenfaces ($\Phi_f = \sum_{i=1}^{L} \omega_i \mu_i$), die Eigenfaces sind orthonormale Vektoren

$$\Phi_f^T\Phi_f = \sum_{i=1}^{L} \omega_i^2 \tag{33}$$

und

$$\epsilon^2(x,y) = \Phi^T(x,y)\Phi(x,y) - \sum_{i=1}^{L} \omega_i^2(x,y) \tag{34}$$

und $\epsilon(x,y)$ und $\omega(x,y)$ sind die Skalarfunktionen der Bildposition und $\Phi(x,y)$ ist eine Vektorfunktion der Bildposition.

Der zweite Teil der Gleichung 34 wird in der Praxis durch eine Korrelation mit den L Eigenfaces berechnet:

$$\begin{aligned}\sum_{l=1}^{L} \omega_l^2(x,y) &= \sum_{l=1}^{L} \Phi^T(x,y)\mu_l \\ &= \sum_{l=1}^{L} [\Gamma(x,y) - \Psi]^T \mu_l \\ &= \sum_{l=1}^{L} [\Gamma^T(x,y)\mu_l - \Psi^T\mu_l] \\ &= \sum_{l=1}^{L} [I(x,y) \otimes \mu_l - \Psi^T\mu_l]\end{aligned} \tag{35}$$

wobei $\otimes$ der Korrelationsoperator ist. Der erste Teil der Gleichung 34 wird zu

$$\begin{aligned}\Phi^T(x,y)\Phi(x,y) &= [\Gamma(x,y) - \Psi]^T[\Gamma(x,y) - \Psi] \\ &= \Gamma^T(x,y)\Gamma(x,y) - 2\Psi^T\Gamma(x,y) + \Psi^T\Psi \\ &= \Gamma^T(x,y)\Gamma(x,y) - 2\Gamma(x,y) \otimes \Psi + \Psi^T\Psi\end{aligned} \tag{36}$$

so dass

$$\epsilon^2(x,y) = \Gamma^T(x,y)\Gamma(x,y) - 2\Gamma(x,y) \otimes \Psi + \Psi^T\Psi + \sum_{l=1}^{L}[\Gamma(x,y) \otimes \mu_l - \Psi \otimes \mu_l] \quad (37)$$

ist. Weil das Durchschnittsbild Ψ und die Eigenfaces μ_l schon gegeben sind, können die Teile $\Psi^T\Psi$ und $\Psi \otimes \mu_l$ im Voraus berechnet werden.

Somit verwendet die Berechnung der Gesichtskarte nur $L+1$ Korrelationen über dem Eingabebild und die Berechnung des ersten Termes $\Gamma^T(x,y)\Gamma(x,y)$. Dieser wird durch die Quadrierung des Eingabebildes $\Gamma(x,y)$ und die Aufsummierung der quadrierten Werte des lokalen Unterbildes an jeder Position im Bild berechnet.

4.2 Neue Gesichter erlernen

Das Konzept des face space gibt dem System die Möglichkeit, zu lernen und anschließend neue Gesichter zu erkennen. Wenn ein Gesicht ausreichend nahe am face space liegt, aber keines der bekannten Gesichter ist, so wird es zu Beginn als *unbekannt* gekennzeichnet. Der Rechner speichert den Mustervektor und das entsprechende Gesicht. Wenn sich mehrere *unbekannte* Mustervektoren im Musterraum ballen, so wird angenommen, dass es sich um ein neues, aber noch nicht erkanntes Gesicht handelt.

Die Bilder die zu den Mustervektoren gehören und sich in diesem Ballungsgebiet anhäufen werden auf Ähnlichkeiten untersucht, indem gefordert wird, das die Entfernung jedes Bildes zum Durchschnitt der Bilder unterhalb eines vordefinierten Grenzwertes liegt. Wenn die Bilder den Ähnlichkeitstest erfolgreich durchlaufen, so wird das Mittel der Eigenschaftenvektoren als bekanntes Gesicht in die Datenbank eingefügt. Zusätzlich können die Eigenfaces neu berechnet werden, nachdem man die neuen Bilder der Trainingsmenge hinzufügt hat.

4.3 Probleme der Eigenfacemethode

4.3.1 Bildhintergrund entfernen

Bisher haben wir den Einfluss des Hintergrunds auf die Berechnung ignoriert. In der Praxis jedoch kann der Hintergrund einen starken Einfluss auf die Leistung der Erkennung haben, zumal die Eigenface Analyse das Gesicht nicht von Hintergrund unterscheidet.

Die in Kapitel 4.1.2 beschriebene Methode der Gesichtsfindung schneidet aber schon, soweit möglich, eben nur das Gesicht aus und verringert somit auch gleichzeitig den Anteil des Hintergrunds im vorhandenen Bild. Sollte der Hintergrund aber immer noch einen zu großen Anteil des Bildes einnehmen, so kann man Hintergrundbereiche im Eingabebild mit einem zweidimensionalen Gauss-Filter glätten. Hiermit wird der Einfluss des Hintergrunds noch weiter vermindert und die Mitte des Gesichtes wird hervorgehoben. Da die Position der Gesichter ausreichend genau gefunden werden konnte, wurde diese Methode aber im Rahmen der Arbeit nicht implementiert.

4.3.2 Größen- und Orientierungsunterschiede

Experimente haben gezeigt, dass die Erkennungsrate sehr schnell abnimmt, wenn die Größe des Kopfes, oder die Skalierung falsch beurteilt werden. Die Größe des Kopfes auf dem Eingabebild muss annähernd die Größe der Eigenfaces haben, damit das System gut arbeitet. Die Bewegungsanalyse liefert eine Schätzung der Kopfgröße, aus der das Bild des Gesichtes auf die Größe der Eigenfaces skaliert wird.

Eine andere Vorgehensweise zur Lösung des Skalierungsproblems ist die Verwendung der Eigenfaces in mehreren Größenskalen. Hierbei wird das Eingabebild mit Eigenfaces mehrerer Skalen verglichen. In diesem Fall wird das Gesicht dort am nächsten zum face space sein, wo es am besten zu der Skala eines Eigenfaces passt. Entsprechend könnte man das Eingabebild auch auf mehrere Größen skalieren und dann die Skala verwenden, die die kleinste Entfernung zum face space hat.

Obwohl der Eigenfaceansatz nicht sehr empfindlich gegenüber der Kopforientierung ist, wird eine Aufnahme, die nicht frontal ist, eine Verminderung der Leistung zur Folge haben. Eine genaue Schätzung der Kopfneigung wird die Erkennung mit Sicherheit verbessern. Auch hierfür gibt es zwei Möglichkeiten. Erstens kann man die Orientierung der Bewegungsregion des Kopfes berechnen. Dies ist aber nicht zuverlässig, da die Regionen immer eine Kreisform anstreben. Besser ist es, die Tatsache auszunutzen, dass Gesichter mehr oder weniger symmetrische Muster sind. Somit kann man die Orientierung anhand einfacher Symmetrieoperatoren schätzen. Ist die Orientierung abgeschätzt, so kann das Bild gedreht werden, um den Kopf mit den Eigenfaces abzustimmen.

Ausgehend von der in [16] beschriebenen und hier verwendeten Methode zur Gesichtserkennung werden diese Probleme jedoch mehr oder weniger umgangen. Da die gefundenen Gesichter mit dem in Kapitel 4.1.2 beschriebenen Verfahren sehr präzise aus dem gesamten Bild ausgeschnitten werden können, enthalten alle Bilder die Gesichter der Personen. Somit ist das Skalierungsproblem recht einfach zu lösen. Es werden einfach alle Bilder auf eine über alle Bilder der Lernmenge gemittelte Durchschnittsgröße skaliert. Dadurch haben auch die Gesichter eine fast identische Größe.

Ebenso löst die Arbeit in [16] das Problem mit seitlich geneigten Gesichtern. Neben dem Originalbild werden nach links und rechts gedrehte Bilder nach Gesichtern untersucht. Somit ist es auch möglich, geneigte Gesichter zu erkennen.

4.3.3 Belichtungsunterschiede

Ein weiteres Problem für die Eigenface-Methode sind unterschiedliche Belichtungsverhältnisse, sowohl in der Trainingsmenge, als auch in den Eingabebildern. In der Realität sind selten optimale Bedingungen für die Aufnahme der Gesichter gegeben, das heißt, das Gesicht wird selten gleichmäßig ausgeleuchtet sein. Da die Eigenface-Methode ein Bild aber als Ganzes betrachtet, werden ein und dasselbe Gesicht, nur einmal von links und einmal von rechts beleuchtet, unter Umständen als zwei unterschiedliche Gesichter erkannt (Abbildung 28).

Daher wird versucht, die Belichtungsverhältnisse durch eine Belichtungskorrektur und einen anschließenden Histogrammabgleich zu normalisieren. Die Belichtungskorrektur versucht ungleichmäßige Belichtungsverhältnisse anzugleichen, während der Histogrammabgleich anschließend den Kontrast maximiert. Für die Belichtungskorrektur wird eine lineare Mehrfachregression verwendet. Die Regression beschreibt die Abhängigkeit wie folgt,

$$y_i^{'} = b_0 + b_1 x + b_2 y \quad (38)$$

Abbildung 28: **Dasselbe Gesicht einmal von rechts und einmal von links beleuchtet.**

wobei $y_i^{'}$ der neue Schätzwert für den Pixelwert y_i ist. Die Faktoren b_0, b_1 und b_2 werden wie folgt berechnet,

$$b_0 = \frac{\sum_{j=1}^{x}\sum_{i=1}^{y} y_{ji} - b_1 \sum_{j=1}^{x}\sum_{i=1}^{y} j - b_2 \sum_{j=1}^{x}\sum_{i=1}^{y} i}{xy} \tag{39}$$

$$b_1 = \frac{xy\sum_{j=1}^{x}\sum_{i=1}^{y} iy_{ji} - \sum_{j=1}^{x}\sum_{i=1}^{y} j \sum_{j=1}^{x}\sum_{i=1}^{y} y_{ji}}{xy\sum_{j=1}^{x}\sum_{i=1}^{y} j^2 - (\sum_{j=1}^{x}\sum_{i=1}^{y} j)^2} \tag{40}$$

$$b_2 = \frac{xy\sum_{j=1}^{x}\sum_{i=1}^{y} iy_{ji} - \sum_{j=1}^{x}\sum_{i=1}^{y} i \sum_{j=1}^{x}\sum_{i=i}^{y} y_{ji}}{xy\sum_{j=1}^{x}\sum_{i=1}^{y} i^2 - (\sum_{j=1}^{x}\sum_{i=1}^{y} i)^2} \tag{41}$$

wobei x und y die Dimensionen des Bildes sind, und y_{ji} das Pixel des Bildes an der Position (j, i) ist.

Durch diese Berechnung erleidet das Eingabebild jedoch einen Kontrastverlust. Dieser Verlust wird durch den Histogrammabgleich kompensiert. Beim Histogrammabgleich wird eine lineare Skalierung der Grauwerte vorgenommen [11]. Dafür werden der maximale (*max*) und der minimale (*min*) Grauwert des Bildes ermittelt. Nach der Ermittlung der Extrema wird jeder Pixelwert y_i durch den Schätzwert $y_i^{'}$ ersetzt

$$y_i^{'} = \frac{(y_i - min)255}{max - min} \quad (i = 1, ..., xy) \tag{42}$$

Dadurch erhält man ein Bild mit stark erhöhtem Kontrast (Abbildung 29).

Die Helligkeitskorrektur und der Histogrammabgleich wurden im Rahmen von [16] schon innerhalb der *MoCA*-Bibliothek implementiert. Da sie aber für [16] individuell angepasst waren, wurden die Algorithmen teilweise in abgeänderter Form übernommen.

Abbildung 29: **Links: Original, mitte: nach Helligkeitskorrektur, rechts: nach Histogrammabgleich**

Lernmenge	Anzahl Bilder	Beschreibung
A	30	20 frontal aufgenommene Gesichter, 10 Bilder, die keine Gesichter darstellen
B	78	26 Personen; Gesichter frontal ohne Hintergrund; unterschiedliche Gesichtsausdrücke; unterschiedliche, nicht einheitliche Ausleuchtung, drei Bilder pro Person
C	144	16 Personen, neun Bilder pro Person, Gesichter von vorne aufgenommen. Verschiedene Ausleuchtung und Kopfhaltung (aufrecht, 45° nach links und rechts geneigt), mit gleichbleibendem Hintergrund
D	Video	ein fünfminütiges Video einer Nachrichtensendung, drei Personen, ca. 3500 Frames mit Gesichtern von 7500 Frames
E	Video	ein fünfzehnminütiges Video einer Nachrichtensendung, 15 Personen, ca. 8700 Frames mit Gesichtern von 17800 Frames

Tabelle 1: Die verwendeten Lernmengen

Weitere Details und eine genauere Beschreibung der Regressionsanalyse können in [16] nachgelesen werden.

5. EXPERIMENTE

5.1 Training

Die Experimente wurden in zwei Hauptkategorien eingeteilt. Es wurden Versuche mit Einzelbildern und Videos durchgeführt. Die Lernmengen basieren ebenfalls auf den Einzelbildern beziehngsweise auf den Gesichtsbildern, die in den Videos gefunden wurden. Aus den Informationen über die Gesichter lassen sich natürlich, über die Zusammenstellung der Lernmenge hinaus, weitere semantische Informationen ableiten.

Im Laufe der Testphase wurden mehrere Lernmengen erstellt und darauf basierend das Durchschnittsbild, die Eigenvektoren, der Grenzwert, die Gewichte und die Gesichtsklassen ermittelt. Voraussetzung für die Experimente war, dass alle Bilder in einem von der *MoCA*-Bibliothek unterstützen Format vorliegen. Momentan begrenzt sich dies auf Dateien in den Formaten ".jpg", ".png", ".pnm" und ".pgm". Alle Bilder in den genannten Formaten wurden mithilfe der *MoCA*-Bibliothek in Graustufenbilder derselben Größe umgewandelt und im ".pgm"-Format gespeichert. Einen Überblick über die vier Lernmengen und eine Testmenge A gibt Tabelle 1.

5.2 Durchführung

5.2.1 Referenzmenge A

Die Referenzmenge A (Abbildung 30) wird nicht als eigene Lernmenge verwendet, sondern zum Test der Zuverlässigkeit der Gesichtserkennung verwendet. Sie besteht aus insgesamt 30 Bildern, von denen 20 ein Gesicht darstellen und 10 kein Gesicht. Nachdem das System mit Lernmengen trainiert wurde, werden diese 30 Bilder ebenfalls der Gesichtserkennung zugeführt. Hierbei wird überprüft, inwieweit diese Bilder erkannt werden können, beziehungsweise, wie gut eine Identifikation von Nichtgesichtsbildern erfolgt.

Abbildung 30: **Die Referenzmenge A**

5.2.2 Testmenge B

Aus der Testmenge B werden mehrere Lernmengen für das System erstellt und es wird mit Bildern aus der Testmenge B und testmengenfremden Bildern experimentiert. Für einen ersten Durchgang wird aus der Menge B von jeder Person nur ein Bild von insgesamt drei Bildern pro Person verwendet, was eine Anzahl von 26 Bildern für die Lernmenge ergibt. Für den zweiten Durchgang werden zwei Bilder von jeder Person verwendet (insgesamt 52 Bilder) und beim dritten Durchgang alle drei (78 Bilder).

Einen Auszug der jeweiligen Lernmengen ist in Abbildung 31 zu sehen. Getestet wird die Gesichtserkennung dann jeweils mit allen Bildern der Testmenge B und mit den 30 Bildern der Menge A, die nicht der Testmenge B entstammen. Die 20 Gesichter der Referenzmenge werden in einer zweiten Testreihe zusätzlich bei jedem der drei Durchgänge in die Lernmenge mit aufgenommen. In einer dritten Testreihe wird in jedem Durchgang der Grenzwert für die Erkennung von Gesichtern gesenkt, nachdem die Gesichtsbilder der Menge A in die Lernmenge mit aufgenommen wurden.

Im Szenario S1, bei dem jeweils nur ein Bild jeder Person der Testmenge B in der Lernmenge vorhanden war, werden alle Bilder der Mengen A und B, die Gesichter enthalten, auch als Gesichtsbilder erkannt. Drei Bilder, die keine Gesichter darstellen, werden fälschlicherweise als Gesichter ingterpretiert und die Gesichter der Menge A werden fälschlicherweise Personen der Menge B zugewiesen, obwohl sie als *unbekannt* eingestuft werden sollten. Dies liegt einfach an einem zu hoch gewählten Schwellwert, der eine zu hohe Toleranz für den Abstand zu den Gesichtsklassen bietet. Die Bilder der Testmenge B können zum Großteil bei der Gesichtserkennung auch den richtigen Personen zugewiesen werden. Neun von 78 Gesichtern, das entspricht in etwa 12 Prozent, werden falsch zugewiesen.

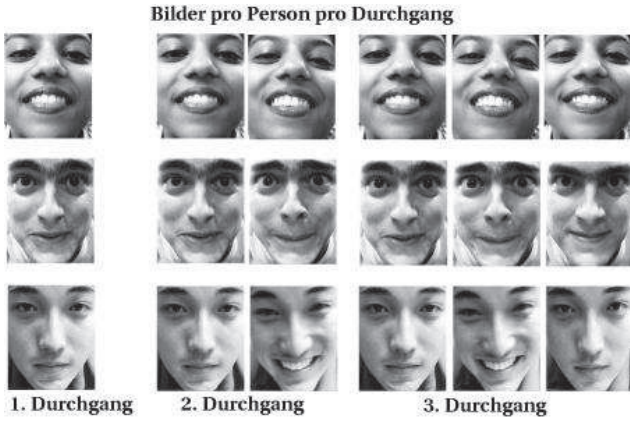

Abbildung 31: **Links: ein Bild pro Person in der Lernmenge im ersten Durchgang, mitte: zwei Bilder pro Person in der Lernmenge im zweiten Durchgang, rechts: drei Bilder pro Person in der Lernmenge im dritten Durchgang**

Ein wesentliches Problem bei der Erkennung liegt hierbei in den Bildern der Lernmenge. Ist das Bild der Lernmenge einer Person stark unterschiedlich im Vergleich zu einem anderen Bild derselben Person belichtet, so unterscheidet sich die Verteilung der Grauwerte ebenfalls stark. Somit ist es immer möglich, dass das Bild einer anderen Person eine ähnlichere Grauwertverteilung hat, wie das zu untersuchende Bild.

Deutlich macht das Abbildung 32. Das linke Gesicht gehört zur Lernmenge, das mittlere Gesicht ist das zu untersuchende und das rechte Gesicht ist das dem mittleren Bild zugeordnete Bild aus der Lernmenge. Deutlich zu erkennen, ist hier die ähnliche Helligkeitsverteilung der beiden rechten Bilder (die Lichtquelle ist vom Betrachter aus links) im Vergleich zum linken Bild (die Lichtquelle ist vom Betrachter aus rechts).

Da die Eigenfacemethode eben nach Ähnlichkeiten der Hauptkomponenten der Gesichter sucht, und keine geometrischen Merkmale oder andere Möglichkeiten der Gesichtserkennung hinzuzieht, werden diese Fehler nicht erkannt. Abgemildert wird dieser Effekt durch die in Kapitel 4.3.3 beschriebene Belichtungskorrektur. Am Besten jedoch kann man diesem Fehler vorbeugen, indem man entweder nur Gesichter mit der gleichen oder sehr ähnlicher Ausleuchtung verwendet, oder die Lernmenge um Bilder mit der entsprechenden Ausleuchtung erweitert.

Im zweiten Szenario, S2, werden von jeder Person in der Testmenge B zwei Bilder in die Lernmenge gegeben. Hier wird nun schon deutlich, dass die Fehler der falschen Zuordnung reduziert werden. Dies ist auch zu erwarten, da nun für die meisten Personen zwei Bilder mit unterschiedlicher Beleuchtung in der Lernmenge vor-

Abbildung 32: **Die Bilder links und rechts sind in der Lernmenge enthalten, das Gesicht in der Mitte wurde untersucht.**

handen sind, und somit auch mehrere Beleuchtungsvarianten erkannt werden können. Auch im Ergebnis spiegelt sich das wieder. Gab es im ersten Durchgang noch neun falsche Zuordnungen der Menge B bei der Gesichtserkennung, so werden nun alle 78 Gesichter der Menge B richtig zugeordnet. Bei den Bildern der Referenzmenge A ergibt sich kaum ein Unterschied. Alle Bilder, die Gesichter enthalten, werden wiederum als Gesichter erkannt, sie werden wegen des zu hohen Grenzwertes aber immer noch den falschen Personen zugewiesen. Die Anzahl der Fehlinterpretationen von Bildern ohne Gesicht als Gesichtsbild bleibt bei drei.

Im dritten Szenario, S3, in dem von jeder Person in der Testmenge B drei Bilder in die Lernmenge gegeben werden, ändert sich für die Bilder der Menge B nichts. Alle Gesichter werden erkannt und korrekt zugeordnet. Auch alle Bilder der Menge A, die ein Gesicht enthalten, werden erkannt, aber immernoch falsch zugewiesen. Die Zahl der Bilder, die kein Gesicht enthalten, aber als Gesichtsbild erkannt werden steigt aber auf fünf Bilder an.

Dies läßt sich durch die steigende Anzahl von Bildern in der Lernmenge erklären. Mit steigender Bilderanzahl erhöht sich auch die Wahrscheinlichkeit, dass die Differenz des Gewichtsvektors des Nichtgesichtsbildes und irgendeines Gewichtsvektors der Lernmenge zufällig doch unter den Schwellenwert fällt.

Diese drei Szenarien werden nun wiederholt, nur dass die Bilder der Menge A, die Gesichter enthalten jeweils zusätzlich in die Lernmenge aufgenommen werden. Für die Bilder der Testmenge B ändert sich hierbei am Ergebnis in den drei Durchgängen nichts. Die Gesichter der Menge A werden nun natürlich auch den richtigen Personen der Lernmenge zugewiesen. Einziges Problem bei dieser Durchführung ist der immer noch zu hoch gewählte Schwellwert, bei dem Bilder, die kein Gesicht enthalten, immernoch als Gesichtsbilder erkannt werden. Für die drei Szenarien in denen auch die Bilder der Referenzmenge A in der Lernmenge enthalten sind, ergeben sich für die Menge A Fehlerquoten von 20% für S1, und jeweils 26,66% für S2 und S3.

Dieses Problem lässt sich relativ einfach beheben. Da sich mit Zunahme der Bilder in der Lernmenge die Möglichkeiten zur Darstellung eines Gesichtes verfeinern, kann man den Schwellwert für die Zuordnung zu den Gesichtsbildern senken. Ist dies getan, so ergibt sich eine konstante Fehlerquote von genau einem Bild, was einer Fehlerquote von drei Prozent für die Menge A entspricht. Das weiterhin fehlerhaft klassifizierte Bild ist in Abbildung 33 zu sehen.

Da die Entfernung vom “face space“ als die Differenz von Originalbild minus Durchschnittsbild und der Projektion des Originalbildes

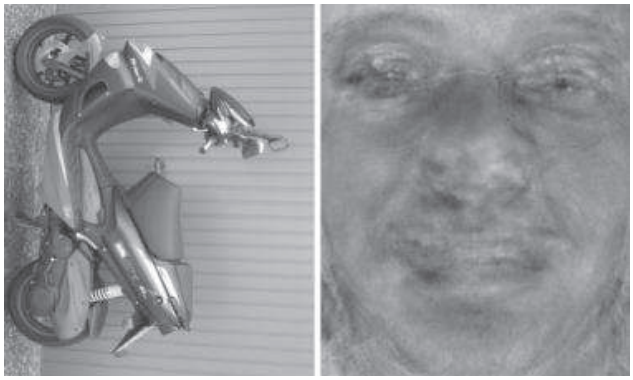

Abbildung 33: **Ein Bild und seine Projektion in den “face space “. Das Bild wird durch Zufall immer als Gesichtsbild eingestuft.**

Abbildung 34: **Ein Gesichtsbild und seine Projektion in den “face space “ mit den gleichen Eigenvektoren wie in Abbildung 33.**

errechnet wird, ergibt sich hier durch Zufall immer eine Entfernung die unterhalb des Grenzwertes liegt.

5.2.3 Testmenge C

Auch aus der Testmenge C werden mehrere Lernmengen erstellt, wobei der Test im Vergleich zu den Tests mit der Testmenge B etwas anders durchgeführt wird. Die Testmenge besteht aus Fotos von 16 Personen. Pro Person gibt es neun Aufnahmen, die jeweils immer in Dreiergruppen unterteilt sind. Jede Gruppe enthält ein Bild mit aufrechten und einem um 45° nach links und rechts geneigten Kopf. Die drei Dreiergruppen unterscheiden sich in der Belichtung der Bilder. Die erste Gruppe ist gleichmäßig von vorne ausgeleuchtet, die zweite Gruppe 45° von vorne rechts und die dritte Gruppe ist direkt von rechts beleuchtet (Abbildung 35).

Abbildung 35: Alle neun Bilder einer Person in der Testmenge C

Hauptaugenmerk dieser Testreihe ist es, teilweise verdeckte Gesichter und Gesichter unter verschiedenen Belichtungszuständen zu erkennen. Für die Verdeckungen werden einfach schwarze Balken unterschiedlicher Größe über die Gesichter gelegt.

Für das Testszenario S4 wird von jeder Person nur ein Bild in die Lernmenge aufgenommen. Hierbei handelt es sich immer um die frontale Aufnahme mit der direkten Ausleuchtung von vorne. Genau dieselben Bilder werden nun mit Balken in vier unterschiedlichen Größen verdeckt (Abbildung 36).

Abbildung 36: Gesichter mit vier unterschiedlichen Verdeckungen

Das Szenario S4 wird nun noch dreimal durchgespielt. Beim ersten Mal werden nun zwei Bilder pro Person (frontal und 45° nach rechts geneigt, Ausleuchtung direkt von vorne) in die Lernmenge gegeben, beim zweiten mal alle drei der ersten Gruppe (frontal und 45° nach rechts und links geneigt, Ausleuchtung direkt von vorne). Anschließend wird wieder versucht, die verdeckten Gesichter zu erkennen.

Im letzten Durchgang für S4 werden zum Abschluss alle Bilder aller Personen in die Lernmenge gegeben und es wird wieder versucht, die verdeckten Gesichter zu erkennen. Diesesmal hat das System, trotz Verdeckungen, keine Probleme alle Bilder den richtigen Personen zuzuweisen.

Zur Durchführung dieses Szenarios lässt sich sagen, dass nicht die Gesichtserkennung das vermutete Problem ist, sondern überhaupt die Feststellung, ob es sich um ein Gesicht handelt oder nicht. Wie die Tests zeigen, werden die verdeckten Gesichter nicht als unbekannte Gesichter eingestuft, sondern als Bilder, auf denen kein Gesicht abgebildet ist. Für diese Bilder ist der Grenzwert zu hoch. Alle Bilder werden als Gesicht erkannt und dann auch richtig zugewiesen. Da hier Bilder in der Lernmenge sind, die stark unterschiedlich ausgeleuchtet sind, steigt auch der Grenzwert, so dass es einfacher für das System wird, ein Bild als ein Gesichtsbild zu markieren. Damit ist das Hauptproblem auch gelöst. Die Gesichtserkennung danach läuft dann, auch bei altem Grenzwert, sehr gut ab, das heißt, es gibt keine falschen Zuordnungen bei der Erkennung.

Die Erkennungsleistung nimmt nur bei den Balken der Größe 128x20 Pixel ab. Dies ist eigentlich verwunderlich, da ja im Vergleich zu der Testreihe mit den Balken der Größe 64x20 zusätzlich nur noch Hintergrund verdeckt wird. Daraus läßt sich schließen, dass die Bedeutung des Hintergrundes nicht ganz irrelevant ist. Nimmt er einen großen Teil des Bildes ein, so erhält er auch eine größere Bedeutung bei der Hauptkomponentenanalyse. Es wird nicht nur das Gesicht auf dem Bild zur Berechnung der Eigenvektoren verwendet, sondern das komplette Bild. Daher lassen sich in dieser Testreihe auch Gesichter noch richtig zuordnen, obwohl ein Großteil des Gesichtes verdeckt ist.

Das ist natürlich ein großes Problem, da es die Ergebnisse verfälscht. Somit sollte für die Gesichtserkennung gewährleistet sein, dass so wenig wie möglich Hintergrund auf den Bildern vorhanden ist, und nur die Gesichter Bestandteil der Hauptkomponetenanalyse sind.

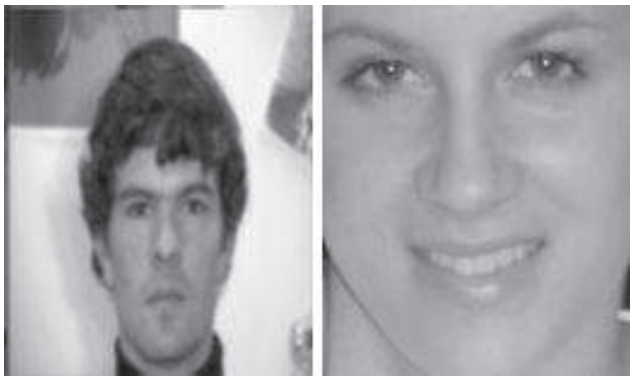

Abbildung 37: Gesichtsbilder mit und ohne Hintergrund

Dieses Problem wird wiederum durch die vorgeschaltete Gesichts-

erkennung der *MoCA*-Bibliothek gelöst. Hier werden die Gesichter immer knapp ausgeschnitten, so dass es kaum Hintergrund gibt. Dies führt dann auch zu den gewünschten unverfälschten Ergebnissen.

Nun wird ein weiteres Szenario, S5, durchgeführt, in dem von neun Personen alle Bilder in die Lernmenge aufgenommen werden und von den übrigen sieben Personen kein einziges. Nachfolgend wird die Entfernung vom "face space" berechnet und untersucht, wie gut unbekannte Gesichter mit gleicher Ausrichtung des Kopfes und gleicher Beleuchtung klassifiziert werden. Im ersten Durchgang werden alle Gesichter auch als Gesichtsbilder erkannt, aber die unbekannten Gesichter werden nicht als solche identifiziert. Erst nachdem man den Schwellwert für die Entfernung zu den Gesichtsklassen reduziert hat, in diesem Fall halbiert hat, tritt das gewünschte Ergebnis ein. Alle Personen, von denen Bilder in der Lernmenge enthalten sind, werden erkannt und richtig zugeordnet, alle anderen Personen werden als unbekannt eingestuft.

Hier liegt das Problem also nicht bei der Erkennung als Gesichtsbild, sondern bei der Entscheidung, ob das Eingabebild eine schon bekannte Person enthält, oder nicht. Da hier keine Verdeckungen vorhanden sind, und die Gesichtsbilder alle ähnlich sind, werden die Bilder der unbekannten Personen auch zuverlässig als Gesichtsbilder erkannt. Es soll aber nach bekannt oder unbekannt klassifiziert werden. Daher sollte man den Schwellwert für die Entfernung zu den Gesichtsklassen anpassen, um die Zahl der falsch zugeordneten Gesichtsbilder zu reduzieren.

Im letzten Szenario, S6, werden alle Bilder der ersten Gruppe in die Lernmenge genommen und es wird untersucht, wie gut die Erkennung abläuft, wenn es größere Belichtungsunterschiede auf einzelnen Bildern der Personen gibt. Auch hier tritt zuerst das Problem des zu hohen Schwellwertes für die Entfernung zu den Gesichtsklassen auf. Solange der Schwellwert nicht halbiert wird, werden alle Gesichtsbilder zugewiesen, auch wenn es dabei zu falschen Zuweisungen kommt. Erst nachdem der Schwellwert reduziert wird, werden Bilder, die zu weit von den Gesichtsklassen entfernt sind als unbekannt klassifiziert. Es stellt sich heraus, dass alle Bilder als Gesichtsbilder klassifiziert werden. Die Erkennung der Person beziehungsweise die Nichterkennung folgt keiner Regelmäßigkeit. Das einzige was sich sagen läßt, ist, dass wenn eine Erkennung stattfindet, das Gesichtsbild auch der richtigen Person zugewiesen wird. Insgesamt werden 45 von den 144 Bildern als unbekannt eingestuft. In einem zweiten Durchgang werden alle Bilder der ersten und zweiten Gruppe in die Lernmenge gegeben. Die Ergebnisse ähneln den Ergebnissen des ersten Durchgangs von Szenario S6, nur dass sich die Erkennungsrate verbessert. Es werden nur noch 21 von 144 Bildern als unbekannt eingestuft. Um das Szenarion noch abzuschließen werden in einem obligatorischen dritten Durchgang alle Bilder aller Personen in die Lernmenge gegeben. Wie zu erwarten ist, werden nun alle Personen korrekt zugewiesen.

Für die Erkennung und richtige Zuordnung der Bilder spielt die Wahl des Grenzwertes auch hier eine entscheidende Rolle. Zum einen hat man die Möglichkeit, eine möglichst große Anzahl von korrekten Zuweisungen zu erhalten, und damit aber die Gefahr, manche Gesichter falsch zuzuweisen, zum anderen besteht die Möglichkeit, eine kleinere Anzahl von korrekt zugewiesenen Bildern, eine größere Anzahl als unbekannt eingestufter Bilder und dadurch eine stark reduzierte Anzahl von falsch zugewiesenen Bildern zu erhalten.

Aus den Ergebnissen kann man schließen, dass es für eine gute Erkennungsleistung notwendig ist, Gesichter so genau wie möglich auszuschneiden, um Hintergrund zu vermeiden, da dieser auch zur Berechnung der Eigenvektoren beiträgt. Die Eigenvektoren sollen aber die Hauptbestandteile der Gesichter darstellen. Weiterhin lassen sich die Ergebnisse durch eine geeignete Wahl des Schwellwertes verbessern. Bei teilweise verdeckten Gesichtern besteht bei zu geringem Schwellwert die Gefahr, dass die Gesichtsbilder nicht mehr als solche erkannt werden und somit auch keiner weiteren Untersuchung unterzogen werden. Andererseits birgt ein zu hoher Schwellwert die Gefahr, dass unbekannte Personen als bekannt eingestuft werden. Somit sollte man die Schwellwerte für die Entfernung zum "face space" und zu den Gesichtsklassen individuell einstellen.

5.2.4 Testvideo D

Für die vorletzte Testreihe wurde keine Bildermenge verwendet, sondern ein digitalisiertes, fünf minütiges Video einer Nachrichtensendung. Es wurde versucht, den Sprecher und Personen zu erkennen, die in den Berichten der Nachrichten auftreten. Dazu musste das Video erst einmal eine leicht veränderte Variante der Gesichtsdetektion der *MoCA*-Bibliothek durchlaufen, bei der alle gefundenen Gesichter in einem Verzeichnis abgespeichert werden. Dies ist notwendig, um Gesichtsbilder für die Trainingsmenge zu erhalten. Hierbei treten die ersten Problemen auf. Es ist schwierig, eine gute Auswahl an Bildern für die Trainingsmenge zu finden, da sich Personen in Videos zumeist doch leicht bewegen, den Kopf neigen, die Augen kurz schließen, den Mund öffnen und schließen (Abbildung 38). Daher ist es notwendig, eine breite Variation von Bildern der Personen, die in einem Video vorkommen können zu erhalten und diese der Lernmenge zur Verfügung zu stellen.

Abbildung 38: Dieselbe Person mit unterschiedlichen Gesichtsausdrücken (Quelle: Testvideo D)

Für die Analyse einer Nachrichtensendung wurden einige Voraussetzungen angenommen:

- Verschiebt sich ein Gesicht innerhalb zweier Frames um weniger als 20 Pixel in eine Richtung, so wird es als zur selben Person gehörend betrachtet. Dies ist notwendig, um zusammenhängende Sequenzen derselben Person erkennen zu können. Dies reduziert später den Rechenaufwand.
- Wird für eine bestimmte Anzahl von Frames (<10) kein Gesicht gefunden, und danach wird wieder ein Gesicht an einer Stelle gefunden, an der vor 10 oder weniger Frames ein Gesicht war, so wird es als zur selben Person gehörig betrachtet.

Nach beendeter Gesichtsfindung werden Vektoren errechnet, die die Position eine Gesichtes enthalten und zählen, in wie vielen Frames hintereinander dieses Gesicht vorkommt. Nur wenn ein Gesicht in mehr als 10 Frames nacheinander gefunden wird, wird sein Vektor weiterverwendet. Bei allen anderen Vektoren wird angenommen, dass es sich um falsche Entdeckungen von Gesichtern handelt. Aus den Vektoren die übrig bleiben, werden jeweils 15 zufällige Gesichter entnommen und der Gesichtserkennung zugeführt. Werden von diesen 15 Gesichtern mehr als acht derselben Person zugeordnet, so wird angenommen, dass in dem zugehörigen Vektor eine Sequenz von Gesichtern der erkannten Person gespeichert ist.

Die Gesichtsfindung lieferte zur zwei Personen. Zum einen die Sprecherin, zum anderen einen Bericht.

Abbildung 39: **Gesichter, die gefunden und korrekt zugeordnet wurden.**

Nach der automatischer Auswertung gab es 10 Sequenzen, in denen die Sprecherin auftauchte und fünf Sequenzen die den Berichten zugeordnet wurden. Von den 10, der Sprecherin zugeordneten, Sequenzen wurden alle korrekt erkannt. Von den fünf Berichtssequenzen wurde eine ausgeschlossen, weil in dieser niemand erkannt wurde. Die anderen vier wurden korrekt als Bericht angezeigt. Somit wurden alle Sequenzen korrekt behandelt und erkannt beziehungsweise ausgeschlossen. In der Sequenz die ausgeschlossen wurde, hat die Gesichtsdetektion fälschlicherweise ein Gesicht entdeckt, obwohl keines vorhanden war. Die Lernmenge wurde mit 22 zufällig ausgewählten Bildern der Sprecherin und 16 zufällig ausgewählten Bildern des Mannes erstellt. Eine Sequenz eines anderen Mannes wurde leider von der Gesichtsfindung nicht erkannt. Dies lag jedoch daran, dass das Gesicht zu stark um die eigene Achse gedreht war (Abbildung 40).

5.2.5 Testvideo E

In der letzten Testreihe wird wiederum ein Video analysiert. Diesmal wird eine 15-minütige Nachrichtensendung ausgewertet. Erneut muss das Video erst eine Gesichtsdetektion durchlaufen und die gefundenen Gesichter werden wieder in einem Verzeichnis gespeichert. Nach der automatischen Auswertung werden insgesamt

Abbildung 40: **Das Gesicht der Person wurde nicht gefunden, da es zu stark gedreht war.**

63 Sequenzen erkannt, die ein Gesicht enthalten. Dies ist jedoch mit Vorsicht zu betrachten, da ja, wie beim vorhergehenden Nachrichtenvideo, eine Sequenz berechnet werden kann, bei der fälschlicherweise Gesichter entdeckt werden, obwohl keine vorhanden sind. Nach zufälliger Auswahl der Bilder für die Trainingsmenge werden im Test 13 der 15 in der Lernmenge enthaltenen Personen gefunden und erkannt. In acht der 63 Sequenzen wird keine Person erkannt, in den restlichen 55 Sequenzen wird jeweils eine Person erkannt. Insgesamt werden also zwei Personen der Lernmenge nicht im Video gefunden.

Bei der ersten Person, die nicht gefunden wird, unterscheidet sich die Positionen des Gesichtes im Video am Anfang der Sequenz und am Ende der Sequenz so stark voneinander, dass die Gesichter doch nicht als zueinander gehörend eingestuft werden und die Sequenz somit als fehlerhaft eingestuft wird.

Bei der zweiten Person, die in der Lernmenge vorhanden ist, aber im Video nicht erkannt wird, liegt der Fehler wiederum bei der Gesichtsfindung. Die Sequenzen, in denen das Gesicht gefunden wird, sind zu kurz und nicht zusammenhängend genug, als dass die Sequenzen für eine Weiterverarbeitung beachtet würden.

Daher läßt sich sagen, dass für eine gute Erkennungsleistung in Videos zwei Kriterien ausreichend erfüllt sein müssen.

Erstens muss die Gesichtsdetektion die Gesichter in einer genügend langen Sequenz lückenlos finden und zweitens muss die Lernmenge ausreichend groß gewählt werden, damit die Gesichter einer Person trotz Mimikunterschieden zur selben Person zugewiesen werden können. Sind diese Voraussetzungen erfüllt, so können auch für die Gesichtserkennung in Videos gute Erkennungsraten erreicht werden.

5.2.6 Weitere Experimente

Selbstverständlich sind mit den entwickelten Programmen und durchgeführten Experimenten die vorhandenen Möglichkeiten der Gesichtserkennung noch lange nicht ausgeschöpft. Es lassen sich mit

geeigneten Lernmengen eine Unmenge von weiteren semantischen Fragen lösen. Man kann sich Filme aus einer Datenbank anzeigen lassen, in denen bestimmte Schauspieler Dialoge miteinander führen. Dazu muss man das aufgezeichnete Video einfach nur von einem Programm untersuchen lassen, das Dialoge erkennen kann, für diese Filmabschnitte dann die hier angewendete Gesichtsfindung und Gesichtserkennung verwenden und man erhält das gewünschte Ergebnis.

Natürlich braucht man sich aber nicht nur auf Dialoge beschränken. Man kann anhand der Gesichtsfindung und Gesichtserkennung viele Informationen über ein Video einholen. Zum Beispiel welche beziehungsweise wie viele unterschiedliche Personen in dem aufgezeichneten Video vorkommen. Man kann die Relevanz einer Person anhand der Länge und Anzahl ihrer Auftritte im Video untersuchen. Ein erweitertes System kann natürlich auch bei der Auswahl von Videos helfen. Man kann sich alle archivierten Videos, in denen eine bestimmte Person vorkommt oder eben nicht vorkommt anzeigen lassen.

Wie im Experiment mit der Nachrichtensendung kann man ermitteln, über welche Themen der Sprecher redet. Dies kann man sogar mit anderen Fachgebieten der Erkennung kombinieren. Wird die Nachrichtensendung von Gesichtserkennung und zum Beispiel noch von Texterkennung oder einer Spracherkennung untersucht, so kann man sich eine gute Zusammenfassung der Nachrichtenthemen erstellen lassen und sich nur die für sich selbst relevanten Nachrichten anzeigen lassen.

Selbstverständlich kann man auch Statisiken für Filme errechnen lassen. Welche Person kommt im Video vor? Wie lange am Stück ist die Person in den Szenen zu sehen? Mit welchen Leuten verkehrt die Person? Wie also schon oben erwähnt, sind die Möglichkeiten sehr groß.

6. ZUSAMMENFASSUNG

In der hier vorliegenden Arbeit ist ein System entwickelt worden, das menschliche Gesichter in Bildern oder Videosequenzen findet und diese mit bereits abgespeicherten Mustern menschlicher Gesichter vergleichen kann, um Personen zu erkennen. Das System unterstützt die Inhaltsanalyse des *MoCA*-Projektes (Movie Content Analysis). Durch die simple Grundidee und die Reduzierung eines Gesichtsbildes auf einen Gewichtsvektor ist das System recht schnell und kann Gesichter gut vergleichen. Besonders zuverlässig ist das System bei Einzelbildern von Personen, die frontal aufgenommen wurden und kaum Hintergrund enthalten.

Das System basiert auf einer einfachen Idee aus [25]. Die Menge aller Bilder fällt in einen einzigen riesigen Raum, Gesichtsbilder jedoch nehmen nur einen sehr kleinen Teil dieses Raumes ein, den so genannten "face space". Für diesen Raum werden die Hauptbestandteile (Eigenvektoren) mittels Hauptkomponentenanalyse ermittelt. Anschließend kann man bestimmen, welcher Eigenvektor wie viel zu einem bestimmten Gesichtsbild beiträgt. Die Beiträge jedes Eigenvektors werden dann in einem Gewichtsvektor gespeichert, der somit ein Gesicht repräsentiert. Da man zur Erkennung eines Gesichtes nun nur noch die Gewichtsvektoren vergleichen muss, geht diese Berechnung sehr schnell vonstatten.

Ausgehend von den Gewichtsvektoren lässt sich ein Gesicht bis zu einem gewissen Grad rekonstruieren. Ist ein Teil eines Gesichtes verdeckt, so wird für dieses Bild ein Gewichtsvektor ermittelt. Mit diesem errechneten Vektor und den Eigenvektoren wird nun eine Projektion des Bildes auf den "face space" berechnet, der den verdeckten Teil des Gesichtes rekonstruiert. Anschließend muss noch überprüft werden, ob diese Projektion ein vernünftiges Ergebnis liefert. Ist dies nicht der Fall, war die Verdeckung des Gesichtes zu groß und es konnte nicht rekonstruiert werden.

Selbstverständlich ist es auch möglich Gesichter einer Person mit unterschiedlicher Mimik, oder bei unterschiedlicher Beleuchtung zu vergleichen und zu erkennen. Auch hier wird ein Gewichtsvektor ermittelt und mit schon gespeicherten Vektoren verglichen. Wiederum wird die euklidische Entfernung zwischen zwei Gewichtsvektoren dermittelt. Ist der Abstand genügend klein, so kann davon ausgegangen werden, dass es sich um ein und dieselbe Person handelt. Hier tritt aber auch eines der größten Probleme der Gesichtserkennung mit Eigenfaces auf. Zwei Bilder einer Person, die sehr unterschiedlich beleuchtet sind, können häufig einander nicht zugeordnet werden, da das hellere Bild andere Hauptbestandteile enhält, als das dunklere. Daher wurde vor jeder Erkennung noch eine Belichtungskorrektur vorgeschaltet, die die Helligkeitsverteilung innerhalb eines Bildes bis zu einem gewissen Grad angleichen kann.

Das Verfahren kann auch dazu verwendet werden, Gesichter zu lokalisieren. Jedes Pixel des Hauptbildes wird als Mittelpunkt eines kleineren Bildes verstanden, für das die Entfernung zum "face space" berechnet wird. Je näher das Bild um das Pixel dem "face space" ist, desto dunkler wird es später in der Gesichtskarte, der "face map", eingefärbt. So entsteht ein Muster mit hellen und dunklen Stellen, wobei sehr dunkle Regionen auf das Vorhandensein eines Gesichtes schließen lassen.

Ein weiteres Problem tritt bei der Analyse von Videosequenzen auf. Beispielhaft wurde in dieser Arbeit eine Nachrichtensendung analysiert. Da sich die Personen in den Videoströmen leider nicht "optimal" verhalten, das heißt, sie sitzen nicht still, sie blinzeln, sie reden, sie drehen den Kopf, ..., ist es schwierig, eine geeignete Lernmenge für die Erkennung von Personen zu definieren. Unterscheiden sich die Bilder einer Person nur geringfügig (Augen auf, Augen zu), so wird der berechnete Grenzwert zur Erkennung als Gesicht auch sehr klein ausfallen. Nun wird ein Bild derselben Person, die auch noch lacht unter Umständen nicht mehr erkannt, weil die Entfernung zum "face space" über dem errechneten Grenzwert liegt. In solchen Fällen ist es sinnvoll, einen individuellen Grenzwert zu verwenden, um bessere Ergebnisse zu erzielen.

Abschließend lässt sich sagen, dass das System der Eigenfaces unter kontrollierbaren Umständen und mit einer geeigneten Lernmenge ein sehr robustes Verfahren ist, das sich auch durch einfach Maßnahmen, zum Beispiel das Tragen einer Sonnenbrille, nicht so leicht täuschen lässt. Somit könnte man dieses System zum Beispiel gut an Stellen in Flughäfen integrieren, die Menschen sowieso passieren müssen, wie zum Beispiel dem Metalldetektor. Dort lässt sich dann auch ein qualitativ hochwertiges Bild machen, das sich sehr gut mit Bildern in einer Datenbank vergleichen ließe.

Für die Videoanalyse ist das System gut geeignet, wenn man in die Vorbereitung zur Analyse wesentlich mehr Arbeit einfließen lässt. Damit ist hauptsächlich eine ausreichend große Menge von Bildern für die Lernmenge gemeint. Ist dieser Arbeitsschritt getan, so liefert die Methode der Eigenfaces auch bei der Gesichtserkennung in Filmsequenzen gute Ergebnisse.

7. LITERATURVERZEICHNIS

[1] A.J.Colmenarez and T. Huang. Face detection and recognitio. In *Proceedings of the NATO Advanced Study Institute to Applications*, 1997.

[2] T. Becher, W. Eppler, T. Fischer, H. Gemmeke, and G. Kock. *Neurolution: Integrierte Hard- und Software für die Entwicklung neuronaler Systeme*. 1999.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *IEEE Transactions on Pattern Analysis and Maschine Intelligence*, volume 19(7), July 1997.

[4] R. Brunelli and T. Poggio. Face recognition through geometrical features. Technical report, Instituto per la Ricerca Scientifica e Tecnologica, I38050 Povo, Trento, Italy, 1992.

[5] R. Brunelli and T. Poggio. Face recognition: Features vs. templates. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15 (10), pages 1.042–1.053, Okt. 1993.

[6] S. Carey and R. Diamond. From piecemeal to configurational representation of faces. In *Science*, volume 195, pages 312–313, 1977.

[7] R. Chellappa, C. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. In *Proceeding of the IEEE*, volume 83(5), pages 705–740, 1995.

[8] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[9] O. et al. Low-dimensional representation of faces in higher dimensions of the face space. In *Journal of American Optical Society*, volume 10, pages 405–411, 1993.

[10] R. Fisher. The use of multiple measures in taxonomic problems. In *Annals of Eugenics*, volume 7, pages 179–188, 1936.

[11] P. Haberäcker. *Praxis der Digitalen Bildverarbeitung und Mustererkennung*. Carl Hanser, München, Wien, 1995.

[12] B. Horn. *Computer Vision*. MIT Press, Cambridge, Mass., 1986.

[13] A. Howell. Introduction to face recognition. In L. Jain, U. Halici, I. Hayashi, S. B. Lee, and S. Tsutsui, editors, *Intelligent Biometrix Techniques in Fingerprint and Face Recognition*, pages 217–284. CRC Press, 1999.

[14] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 12(1), 1990.

[15] L.Wiskott. Labeled graphs and dynamic link matching for face recognition and scene analysis. In *Reihe Physik*, volume 53. Verlag Harri, 1995.

[16] D. Müller. Automatische detektion von gesichtern in bewegtbildern. Master's thesis, Universität Mannheim, 1997.

[17] A. Pentland and T. Choudhury. Face recognition for smart environments. In *IEEE Computer*, volume 33(2), pages 50–55, 2000.

[18] M. Pötzsch. Gabor filter. Technical report, Ruhr Universität, Bochum, Germany, 1996.

[19] H. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. Technical Report CMU-CS-95-158R, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1995.

[20] A. Shashua. *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, Massachusetts Institute of Technology, 1992.

[21] W. Silver. *Determining Shape and Reflectance Using Multiple Images*. PhD thesis, Massachusetts Institute of Technology, 1980.

[22] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. In *Journal of the Optical Society of America A*, volume 4(3), pages 519–524, 1987.

[23] H. Spies. Face recognition – a novel technique. Master's thesis, University of Dundee, 1995.

[24] J. Tautges. Gesichtserkennung mithilfe von eigengesichtern. Technical Report Proseminar Robuste Signalidentifikation, Universität Bonn, 2004.

[25] M. Turk and A. Pentland. Eigenfaces for recognition. In *Journal of Cognitive Neuroscience*, volume 3(1), 1991.

[26] M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computing Vision and Pattern Recognition*, Juni 1991.

[27] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. In *Proceedings of International Conference on Computer of Images and Patterns (CAIP)*, volume 1296, pages 456–463, 1997.

[28] R. Woodham. Analysing images of curved surfaces. In *Artificial Intelligence*, volume 17, pages 117–140, 1981.