DIPLOMARBEIT

Gesichtserkennungs-System zur Raumüberwachung

ausgeführt am Institut für

Mustererkennung und Bildverarbeitung

unter der Anleitung von
o.Univ.Prof. Dr. Walter Kropatsch und Univ.Ass.Dr. Horst Bischof

durch

Thomas Nemec

Alexanderstr.6, 3032 Eichgraben

25.September 1995 .....

Abstract

The goal of this diploma thesis is to develop a face recognition system that uses a CCD camera fixed on a robot arm. In order to extend the “visual field“ of the system the robot moves with the camera. The task of the system is to find people and to identify them. After a generally introduction some methods of face recognition and detection are explained. The structure of the system and its tasks are presented. To evaluate the whole system a database consisting of people from the staff of the department is created. The goal is then to check if a person entering the room belongs to the staff or not. In the case the person belongs to the staff its name should be given.

The system consists of following major modules: Motion detection, Face detection, and Face recognition. In particular the following methods are used within the modules: Motion Energy detection for Motion detection, Multilayer Perceptrons for Face detection, and the Eigenface approach of Turk & Pentland for Face recognition.

After evaluating the individual modules we tested the face recognition system on different people entering the observation room. Provided one person stays app. 10 seconds in the observation room the overall recognition rate is 82.4 % which is a good result considering the speed of one frame per second on fully automatic face recognition.

Übersicht

In dieser Diplomarbeit soll ein Gesichtserkennungs-System entwickelt werden. Eine CCD-Kamera wird dabei fix auf einem Roboterarm montiert. Der Roboter soll mit der Kamera in verschiedene Richtungen „schauen“ können, damit das „visuelle Feld“ größer wird. Das System soll den Aufenthaltsort einer Person im überwachten Raum bestimmen und sie zu identifizieren versuchen. Nach einer kurzen Einführung in die allgemeine Forschung der Gesichtserkennung werden verschiedene Methoden der Gesichtserkennung und -detektion erläutert. Im Anschluß daran wird der Systemaufbau vorgestellt und auf die Verteilung der Aufgaben näher eingegangen. Zur Evaluierung des Gesamt-Systems wird eine Datenbasis, die aus Angestellten des Institutes besteht, aufgenommen. Das entworfene System soll selbständig erkennen, ob Personen, die den Überwachungsraum betreten, zum Institut gehören oder nicht. Im ersten Fall soll der Name des Angestellten ausgegeben werden.

Das Gesichtserkennungs-System besteht aus drei Hauptkomponenten: der Motion Detection, der Face Detection und der Identification. Folgende Methoden werden in den Modulen verwendet: Motion Energy Detection in der Motion Detection, Multilayer Perceptrons in der Face Detection und der Eigenface-Ansatz von Turk & Pentland im Identification-Modul.

Nach der Evaluierung der einzelnen Module wurde die Performance des Systems bei der Erkennung von verschiedenen Leuten des Institutes, die den Überwachungsraum betreten, getestet. Unter der Voraussetzung, daß sich eine Person ca. 10 Sekunden im Überwachungsbereich aufhält, wird eine Erkennungsrate von 82.4 % erzielt, was in Anbetracht einer Geschwindigkeit von einem Frame pro Sekunde bei voll automatischer Gesichtserkennung ein guter Wert ist.

Inhaltsverzeichnis

1. EINLEITUNG 1

2. PSYCHO-BIOLOGISCHE GRUNDLAGENFORSCHUNG 2

Gesichter erkennen 2

Spezielle Zellen zur Gesichtserkennung 2

Einteilung der Gesichtserkennung 3

3. METHODEN DER GESICHTSERKENNUNG 6

Arten der Repräsentation 6

Vorverarbeitung 7

Detektion 7
Bewegungserkennung 7
Schablonenvergleich (Template Matching) 8
Feature-Detektoren (z.B. Symmetrieoperatoren) 8
„Face Space"-Projektion 9
Neurale Netze 11

Identifikation 13

4. AUFBAU DES FACE-RECOGNITION-SYSTEMS 17

Aufgabenverteilung 17

Funktionsbeschreibung der einzelnen Module 17
Gesichts-Detektion am PC 18
Gesichts-Identifikation auf der WS 18

Bestimmung der Auswahlkriterien 19

5. AUSWAHL UND IMPLEMENTIERUNG DER VERFAHREN FÜR DIE EINZELNEN MODULE 21

Motion Detection 21

„Faceness" 23
Vergleich zwischen Eigenface-Ansatz und MLP 24
Beschreibung des „Faceness"-Moduls 33

Face Location **33**

Face Identification **33**

Robot Control **35**

Schnittstellen zwischen den Modulen **35**
Robot Control ↔ Motion Detection 36
Motion Detection ↔ Face Location 36
Face Location ↔ „Faceness“ 36
„Faceness“ (PC) → (WS) Face Identification 36

6. EVALUIERUNG DES FACE-RECOGNITION-SYSTEMS **37**

Aufnahme der neuen Datenbasis **37**

Training der Module „Faceness“ & Identification **38**

Performance der einzelnen Module und des Gesamtsystems **40**
Erkennungsraten der einzelnen Module 40

Erkennungsrate und Geschwindigkeit des Gesamtsystems **45**

Überlegungen zur Steigerung der Performance **46**
Motion Detection 46
„Faceness“ 47
Identification 48

7. ZUSAMMENFASSUNG **50**
LITERATURVERZEICHNIS 51

1. EINLEITUNG

In den letzten Jahrzehnten wurde bei den Forschungen am menschlichen Organismus zum Teil großes Augenmerk auf das „Sehen“ gelegt. Dieser menschliche Sinn ist für uns vom informationstheoretischen Standpunkt aus gesehen von großer Bedeutung, weil durch ihn große Mengen an Information aufgenommen werden können, die anschließend im Gehirn weiterverarbeitet wird.
Besonders interessant erscheint unter anderem der Bereich der **Gesichtserkennung**, der gerade in den letzten Jahren erhöhte Aufmerksamkeit von den Forschern aus unterschiedlichen Disziplinen hat. Das Ziel solcher Forschungen soll und kann mit der heutigen Technik nicht die perfekte Nachbildung aller menschlichen Gesichtserkennungsmechanismen sein. Vielmehr soll der Computer dazu verwendet werden große Mengen von Gesichtern, die die Kapazitäten des menschlichen Gehirnes überschreiten, zu erkennen.

Im Zuge dieser Diplomarbeit soll ein Gesichtserkennungssystem entstehen, das für Überwachungsaufgaben eingesetzt werden kann. Das Hauptaugenmerk wird dabei auf die Lokalisierung von Gesichtern in einer bewegten Szene gelegt. Das System soll modular aufgebaut werden, sodaß die in den einzelnen Modulen verwendeten Methoden jederzeit durch andere ersetzt werden können und dadurch laufende Verbesserungen des Systems möglich sind.
Meine Aufgabe besteht darin, die gängigen Methoden der Gesichtserkennung den verschiedenen Modulen zuzuordnen, wobei eine Auswahl zwischen den sich anbietenden Algorithmen getroffen werden muß. Bei der Entwicklung des Systems sind das richtige Zusammenspiel zwischen den Modulen und die Funktionalität des Gesamtaufbaus weitaus wichtiger als eine perfekte Optimierung der einzelnen Systemkomponenten, die jederzeit nach Belieben verändert werden können.

In den folgenden zwei einleitenden Kapiteln möchte ich die Probleme, die bei der Erkennung von Gesichtern entstehen, etwas näher beleuchten.
Kapitel 2 behandelt einige psychologische und biologische Aspekte der Gesichtserkennung und führt uns somit Zusammenhänge vor Augen, die wir bei technischen Betrachtungen dieser Thematik immer im Hinterkopf behalten sollten.
In Kapitel 3 stelle ich einige Methoden der Gesichtserkennung vor. Das Kapitel umfaßt bisher verwendete Methoden, die zur Verarbeitung von Gesichtern entwickelt wurden. Der grundsätzliche Aufbau des Face-Recognition-Systems wird in Kapitel 4 beschrieben. In Bezugnahme auf das Kapitel 3 werde ich in Kapitel 5 eine Auswahl geeigneter Techniken für die einzelnen Module des Systems treffen. Wenn mehrere Methoden für ein Modul zur Verfügung stehen, wird durch Versuche die jeweils günstigste bestimmt.
Schließlich wird in Kapitel 6 eine Evaluierung des Gesamt-Systems durchgeführt und es werden für zukünftige Verbesserungen Vorschläge erarbeitet. Den Abschluß der Diplomarbeit bildet eine kurze Zusammenfassung.

2. PSYCHO-BIOLOGISCHE GRUNDLAGENFORSCHUNG

Einen Einblick in die Materie der visuellen Wahrnehmung des Menschen liefern eine Vielzahl von Forschungsergebnissen und Testreihen im Bereich der Biologie und auch der Psychologie. Ich möchte im folgenden die wichtigsten Aspekte der Gesichtserkennung herausgreifen.

Gesichter erkennen

Speziell im Bereich der Gesichtserkennung stellen sich zwei grundlegende Fragen. Erstens: Wird ein Gesicht als Ganzes oder als Kombination einzelner Features erkannt ? Und zweitens: Wie wird uns ein Gesicht vertraut ? Die Antworten auf beide Fragen scheinen eng miteinander verbunden zu sein.
Die Antwort auf die erste Frage gibt uns Sergent [Ser86]. Sowohl die Information über die einzelnen Komponenten (Features) als auch über deren Konfiguration ist notwendig, um Gesichter zu erkennen. Als Beispiel ist in Bild 1 ein den meisten sicher unbekanntes Gesicht dargestellt, daß mit einem Tiefpaßfilter (filtert hohe Frequenzanteile heraus) behandelt wurde. Während Gesicht (c) schon ausreicht, um festzustellen, daß es sich um ein Gesicht handelt, können wir aus Gesicht (e) auch das Geschlecht bestimmen und Aussagen über das Alter und das generelle Aussehen der Person treffen. Zur Identifikation werden auch die höheren Ortsfrequenzanteile (wie in Gesicht (i)), die die einzelnen Features genauer definieren, herangezogen.
Trotzdem könnte jemand, der die Person kennt, das Gesicht (e) bereits identifizieren, obwohl noch keine einzelnen Features sichtbar sind, sondern nur ihre relativen Positionen im Gesicht (Konfiguration). Anscheinend ist in den tieferen Ortsfrequenz-anteilen die meiste relevante Information enthalten. Falls wir uns jedoch nicht sicher sind, verwenden wir noch weitere Informationen aus den höheren Ortsfrequenzen.
Sergent stellt fest, daß je geläufiger uns ein Gesicht ist, desto komplexer ist es im Gehirn repräsentiert. Es scheint als würden unbekannte Gesichter nur durch tiefe Ortsfrequenzanteile und Gesichter, die wir gut kennen, zusätzlich durch hohe Ortsfrequenzanteile in unserem Gehirn verankert sein.

Spezielle Zellen zur Gesichtserkennung

Wichtige Hinweise über die Funktion komplexer Zellen in der Schläfenhirnrinde (cortex temporalis) lieferten die Versuche von Perrett et al. [PMP86] an Makaken (Affenart). Sie fanden fünf verschiedene Zellentypen, von denen jeder auf eine andere Ansicht des Kopfes reagierte. Diese fünf Ansichten nannten sie: full face, profile, back of head, head up und head down. Zellen des gleichen Typs sind auch im Gehirn anatomisch nebeneinander angeordnet. Dabei wird jede dieser (prototypischen oder orthogonalen) Ansichten separat und parallel verarbeitet. Zum Beispiel wird ein Kopf, der um 45 Grad zur Seite gedreht ist, teilweise von den „full face“-Zellen und teilweise von den „profile“-Zellen erkannt, denn sie fanden keine Zellen, die maximal auf diese Ansicht reagierten.

Weiters stellten sie fest, daß ungefähr 10% aller Zellen identitätssensitiv sind, d.h., sie reagieren nur auf Kopfansichten einer bestimmten Person.

Bild 1: ***Ein Gesicht mit steigenden Ortsfrequenz-Anteilen von (a) bis (i), wobei (i) das Originalgesicht ist (pro Gesicht um Faktor √2 erhöht, zum Vergleich:(a) enthält 1.55 Perioden/Gesichtsbreite, ..., (c) 3.1, ..., (e) 6.2, ..., (g) 12.4, usw...). (aus [Ser86].)***

Einteilung der Gesichtserkennung

Die Erkennung von Gesichtern stellt das visuelle System des Menschen vor mehrere Probleme, von denen ich drei herausgreifen möchte (siehe auch [SI92]) : Repräsentation, Detektion (Finden) und Identifikation.

Über die Art der **Repräsentation** von Gesichtern im menschlichen Gehirn gibt es noch zu wenig Hinweise. Zumindest kann man annehmen, daß fremde Gesichter intern anders repräsentiert sind als jene, die uns geläufig sind (siehe [Ser86]).
Detektion ist notwendig, weil wir zuerst die Position eines Gesichtes in der Szene bestimmen müssen, um es analysieren zu können. Alles deutet darauf hin, daß für diesen Zweck das Gesicht als Ganzes (Gestalt) verarbeitet wird. Wir können daher auch teilweise verdeckte Gesichter lokalisieren, indem die fehlende Information durch unser Wahrnehmungssystem einfach ersetzt wird. Dieser Detektions-Mechanismus ist auch extrem robust in Bezug auf wechselnde Lichteinflüsse und Distanzveränderungen. Zwei wichtige Aspekte bei der Detektion sind die minimale Auflösung und die minimale Anzahl der Graustufen, die notwendig ist, um ein Gesicht noch detektieren zu können. In Bild 2 sehen wir ein menschliches Gesicht (entnommen aus der Datenbasis in [TP91]) bei verschiedenen Bildauflösungen (je niedriger die Auflösung, desto geringer der Anteil hoher Ortsfrequenzen), welches bis zu einer Auflösung von 32×32 noch gut erkennbar ist. Und wenn wir Bild 3 betrachten, kann auch im Bild mit nur 1 bpp (Bit pro Pixel) das Vorhandensein eines Gesichtes registriert werden, weil die Bildauflösung (mit 128×128) entsprechend hoch ist. Es ist experimentell von Samal [Sam91] nachgewiesen worden, daß für die Detektion von Gesichtern eine Auflösung von $32 \times 32 \times 4$ bpp ausreichend ist. Andere setzen diese untere Grenze sogar noch niedriger an (bis zu 100-200 bits pro Gesicht).
Unter **Identifikation** versteht man die Zuordnung eines Namens (oder eines anderen Bezeichners) zu einem Gesicht, d.h. verschiedene Bilder des gleichen Gesichtes werden immer als dieselbe Person erkannt. Im Durchschnitt kennt jeder Mensch über 700 Leute persönlich und tausende von namentlich unbekannten Gesichtern. Allgemein spielen Features wie Haare und Augen eine größere Rolle bei der Identifikation als Features wie Kinn und Lippen, was Goldstein et al. [GHL71] schon 1971 herausgefunden haben. Laut ihren Schätzungen wächst die Anzahl der notwendigen Features mit der Anzahl der Gesichter, die unterschieden werden sollen. Danach könnte eine Person unter Verwendung von nur zehn verschiedenen Features etwa 1000 Gesichter identifizieren. Auch für die Identifikation von Gesichtern reicht nach Samal [Sam91] eine Auflösung von $32 \times 32 \times 4$ bpp aus.

Weiters möchte ich noch den Begriff der **Wiedererkennung** von Gesichtern einführen. Bei der Identifikation ist die Zuordnung eines Namens (oder Labels) zu einem Gesicht unbedingt notwendig. Für das Wiedererkennen eines Gesichtes hingegen ist es nur erforderlich, daß man das Gesicht schon einmal gesehen hat und sich daran erinnert, d.h. es muß ein vertrautes Gesicht sein.

Bild 2: ***Ein menschliches Gesicht bei verschieden Bildauflösungen: (a) 128×128, (b) 64×64, (c) 32×32, (d) 16×16. (Alle Bilder haben 8 bpp und sind auf dieselbe Größe gebracht worden.)***

Bild 3: ***Ein menschliches Gesicht bei verschiedener Anzahl von Grauschattierungen: (a) 8 bpp, (b) 4 bpp, (c) 2 bpp, (d) 1 bpp. (Alle Bilder haben eine Auflösung von 128×128.)***

3. METHODEN DER GESICHTSERKENNUNG

In diesem Kapitel möchte ich auf die verschiedenen Verfahren, die bei der Erkennung von Gesichtern verwendet werden, näher eingehen. Der Einfachheit und Übersichtlichkeit halber behalte ich dabei die Einteilung bei, die auch Samal und Iyengar [SI92] in ihrem zusammenfassenden Report gewählt haben, und die schon im vorigen Kapitel von mir verwendet wurde, nämlich: Repräsentation, Detektion und Identifikation. Eine ausführliche Beschreibung aller Gesichtserkennungsmethoden würde den Rahmen dieser Diplomarbeit bei weitem sprengen. Daher werde ich versuchen hier nur die wichtigsten Methoden zu beleuchten und jeweils ihre Funktionsweise kurz zu erläutern.

Arten der Repräsentation

In der Gesichtserkennung werden zwei Formen der Repräsentation unterschieden: 2D-Grauwertbilder und Feature-Vektoren.
Am einfachsten kann man ein Gesicht mit einem **2D-Array aus Grauwerten (Intensitätsbilder)** darstellen. Der Nachteil von 2D-Arrays ist, daß sie auf größere Veränderungen der Lichtverhältnisse und der Orientierung und Größe des Kopfes empfindlich reagieren. Zur Vermeidung solcher Probleme werden die Gesichter vorverarbeitet (preprocessing, z.B. Normalisierung von Größe und Position). Trotzdem sind die meisten Systeme, die 2D-Arrays verwenden, robuster als jene auf Basis von Feature-Vektoren , wie in [SI92] berichtet wird. Diese Repräsentation hat weiters den Vorteil, daß sie sowohl Information über einzelne Features (Details sind wichtig bei der Identifikation) als auch deren Konfiguration (Anordung im Gesicht) enthält.
Durch eine Verringerung der Auflösung auf $32 \times 32 \times 4$ bpp ([Sam91]) kann ein Gesicht mit 512 Bytes repräsentiert werden, d.h. tausend Gesichter benötigen nur einen halben Megabyte Speicher.

Eine in vielen Arbeiten verwendete Repräsentation für Gesichter ist der **Feature-Vektor** (siehe [SI92]). Dabei können die Features entweder aus Grauwertbildern oder aus Gesichtsprofilen abgeleitet werden. Feature-Beispiele der ersten Kategorie wären Helligkeit des Haares, Größe und Abstand der Augen, Abstand zwischen Augen und Lippen, usw. In Profilbildern werden die Features mit Hilfe einer Menge von charakteristischen Punkten des Profils erzeugt, z.B. die Kerbe zwischen Stirn und Nase, die Nasenspitze, die Kerbe zwischen Nase und Oberlippe und die Kinnspitze. Normalerweise erzeugt man aus den Abständen und Winkeln zwischen diesen charakteristischen Punkten die benötigten Features.

Aufgrund der oben genannten Vorteile von 2D-Arrays werde ich in meiner Arbeit nur auf Techniken eingehen, die auf der direkten Verarbeitung von Intensitätsbildern beruhen.

Vorverarbeitung

Oft werden die Bilder, in denen Gesichter gefunden werden sollen, mit Methoden der Bildverarbeitung vorverarbeitet. Zwei gute Beispiele dafür sind die Kantendetektion und die Bildpyramiden.
Durch die **Kantendetektion** in einem Bild werden Konturen hervorgehoben. Das hat den Vorteil, daß nur die für das Lokalisieren eines Gesichtes relevante Information weiterverarbeitet werden muß. Außerdem werden dadurch Helligkeitsschwankungen, die durch unterschiedliche Lichtverhältnisse bei der Aufnahme der Bilder entstehen, weitgehend ausgeschaltet.
Um Gesichter verschiedener Größenordnungen (bzw. Entfernungen) in einem Bild erkennen zu können, werden **Bildpyramiden** verwendet. Die bekannteste Form ist die Gauß-Pyramide (oder Tiefpaß-Pyramide), bei der von Ebene zu Ebene jeweils der Mittelwert von mehreren benachbarten Bildpunkten in einem gaußgewichteten Fenster gebildet und zu einem Bildpunkt reduziert wird (siehe auch [Kro91]), d.h. man wendet ein Tiefpaßfilter auf das Ausgangsbild an und reduziert die Auflösung des Bildes um den Reduktionsfaktor. Dadurch verschwinden die Details mehr und mehr. Der Nachteil der Gaußpyramide ist die Mehrfachdarstellung von großen Bildmerkmalen in allen Ebenen. Daher entwickelte Burt die Laplace-Pyramide (oder Bandpaß-Pyramide). Ihre Ebenen entstehen jeweils aus der Differenz von zwei aufeinanderfolgenden Ebenen der Gauß-Pyramide. Burt [Bur88] verwendet die Laplace-Pyramide zur Nachbildung einer elektronischen Fovea (Sehgrube des Menschen).

Detektion

Die Detektion von Gesichtern in einem Bild ist ein Segmentationsproblem. Jene Bereiche im untersuchten Bild, in denen sich Gesichter befinden, müssen vom restlichen Bild getrennt werden.
In vielen Arbeiten im Bereich der Gesichtserkennung wurde der Vorgang der Identifikation isoliert betrachtet, mit der Annahme, daß die Position des jeweiligen Objektes bereits bekannt ist. Deshalb gab es bisher nur wenige Vorschläge das Problem der automatischen Detektion von Gesichtern zu lösen.
Man kann die wichtigsten der heute verwendeten Techniken grundsätzlich in fünf Hauptgruppen einteilen: Bewegungserkennung, Schablonenvergleich (Template-Matching), Feature-Detektoren (z.B. Symmetrieoperatoren), die „Face-Space“-Projektion und Neurale Netze. Sie können bei Bedarf auch untereinander kombiniert werden.

Bewegungserkennung

Wenn sich Objekte in der von uns betrachteten Szene bewegen, erregen sie unsere Aufmerksamkeit. Nach demselben Prinzip funktioniert auch die **Motion Detection** (oder Motion-Energy Detection). Unter der Annahme, daß der Kopf von Personen (und damit ihr Gesicht) ständig in Bewegung ist, wird eine Kamerasequenz von Bildern einer Raum-Zeit-Filterung unterworfen. Die einfachste Methode ist die Subtraktion zeitlich aufeinanderfolgender Bilder, wodurch die Szene in „aktive“ und „inaktive“ Bereiche zerlegt wird. Diese Technik erreicht jedoch nur bei langsam bewegten Objekten (wie es

z.B. Menschen sind) eine ausreichende Positionsgenauigkeit. Eine verbesserte Version dieser Methode stellen Murray und Basu [MB94] in ihrer Arbeit vor.
Eine andere Möglichkeit zur Erkennung von Bewegung wären die **Optical Flow-Methoden** ([DC88]). Dabei werden die Bewegungsvektoren für Felder bestimmter Größe berechnet, d.h. es wird festgestellt, in welche Richtung und mit welcher Geschwindigkeit sich der Bildinhalt des betrachteten Feldes von einem Bild zum nächsten bewegt hat.

Um speziell die Gesichter bewegter Personen zu lokalisieren, werden zusätzlich zur Bewegungserkennung manchmal **Regelbasierte Methoden** verwendet. Darunter versteht man die Anwendung bestimmter Regeln, wie z.B.: suche nach einer kleinen Fläche (Kopf) über einer größeren (Körper), zur Ermittlung von vermutlichen Gesichtspositionen (siehe auch [TP91]).

Schablonenvergleich (Template Matching)

Eine Schablone dient zum Auffinden von übereinstimmenden Gesichtern in einem Bild und repräsentiert ein oder mehrere typische Merkmale des Gesichtes (z.B. Augen). Oft ist sie nur aus groben Elementen, wie Anordnungen von Punkten und Linien, aufgebaut. Govindaraju et al. [GSS90] verwendeten nur zwei gerade Linien und zwei Bögen zum Lokalisieren von Gesichtern in Zeitungsbildern. In einer Arbeit von Chetverikov und Lerch [CL93] repräsentieren fünf Punkte (blobs) und vier Streifen (streaks) die Form eines Gesichtes. Dabei werden zuerst bei grober Auflösung (Laplace-Pyramide) „Nasen-Kandidaten“ bestimmt, die dann im Originalbild genaueren Vergleichen unterzogen werden.
Die Schablone kann aber auch ein Teil eines ausgewählten Gesichtes sein. In [BP93] wird zur Detektion eine Augenschablone verwendet, die vom Gesicht des Autors stammt. Als Vergleichskriterium benutzen sie einen Korrelationskoeffizienten. An den Stellen im Bild, wo die größten Korrelationskoeffizienten auftreten, befindet sich mit hoher Wahrscheinlichkeit auch ein Gesicht. Das Problem der Skalierung lösen sie durch Anfertigung von fünf verschieden skalierten Schablonen.

Feature-Detektoren (z.B. Symmetrieoperatoren)

Die „Gestalt-Psychologie“ ([Köh29]) betrachtet die Symmetrie als ein fundamentales Prinzip der menschlichen Wahrnehmung. Daher erfolgt die Anwendung der hier besprochenen Symmetrieoperatoren unter der Voraussetzung, daß Gesichter symmetrisch sind. Diese Technik zur Bestimmung natürlich vorkommender Objekte ist relativ neu auf dem Gebiet der Gesichtserkennung. Yeshurun et al. [YRW92] haben einen Symmetrieoperator entwickelt, der Symmetrien bezüglich mehrerer vordefinierter Achsen innerhalb eines bestimmten kreisförmigen Bereiches erkennt. Mit diesem Operator können sie sowohl vollständige Köpfe bei grober Auflösung als auch einzelne Features, wie Augen, Nase und Mund, bei normaler Auflösung detektieren. Die Autoren demonstrieren auch eine Möglichkeit, wie dieser Operator mit der Hilfe von **Neuralen Netzen** leicht implementiert werden kann.

Die übrigen beiden Methoden möchte ich etwas ausführlicher erklären. Die „Face Space“-Projektion, weil sie auch bei der Identifikation sehr gute Ergebnisse liefert und

Neurale Netze, da sie meiner Meinung nach zukunftsweisend im Bereich der Gesichtsdetektion, aber auch bei der Identifikation, sind.

„Face Space"-Projektion

Aus einer Menge von N Gesichtern Γ_n können mit Hilfe der **Hauptkomponentenanalyse** („Principal Component Analysis" oder „Karhunen-Loeve-Expansion") jene orthonormalen Vektoren gefunden werden, die die Verteilung der Gesichtsbilder im gesamten Bildraum repräsentieren. Die M besten dieser Eigenvektoren (das sind jene mit den größten Eigenwerten) können als eine Menge von Features beschrieben werden, die zusammen die günstigste Charakterisierung der Variation zwischen allen Gesichtern ergeben. Die M **Eigenfaces** u_m spannen einen Teilraum (auch „Face-Space" genannt) auf. Die mathematische Ableitung zur Berechnung der Eigenvektoren ist in [TP91] zu finden.
Ein neues Gesichtsbild Γ wird durch Gleichungen 1 auf den Face Space projeziert. Das Mittelwert-Gesicht Ψ ist das Durchschnittsbild aller Trainingsbilder Γ_n. Die Gewichte w_k formen einen Vektor Ω (Punkt im Face Space), der den Anteil von jedem Eigenface an der Repräsentation des Input-Bildes Γ angibt.

Gleichungen 1: ***Projektion des Input-Bildes Γ auf den Face Space mit Hilfe von M Eigenfaces u_k. Ψ ist das Mittelwert-Bild des Trainingssets Γ_n.***

$$w_k = u_k{}^T \cdot (\Gamma - \Psi) \qquad \text{mit } k = 1,\ldots,M$$

$$\Omega^T = [w_1, w_2, \ldots, w_M] \qquad \text{Vektor der Projektion auf den Face Space}$$

Der Vektor Ω kann anschließend mit den aus den Trainings-Gesichtern Γ_n berechneten Vektoren Ω_n (oder mit K<M Klassen davon) verglichen werden, um jenes Gesicht aus dem Trainingsset zu finden, das dem Input-Bild am besten entspricht. Die einfachste Methode dafür ist die Bestimmung der minimalen Euklidischen Distanz ε_{min}, die unter einem bestimmten Threshold Θ_{dist} sein muß (Gleichungen 2).

Da es durch den niedrig-dimensionalen Raum auch viele Bilder gibt, die nichts mit einem Gesicht zu tun haben, aber trotzdem nahe einer Gesichtsklasse sind, wird zusätzlich der Abstand zum Face Space ε berechnet, der die „Gesichtsähnlichkeit" des Bildes beschreibt. Die Distanz zum Face Space (distance from face space) wird nach Gleichungen 3 bestimmt und muß auch unter einem festgelegten Schwellwert Θ_{dffs} sein (siehe auch Bild 4).

Gleichungen 2: ***Berechnung der Euklidischen Distanzen ε_n zwischen dem projizierten Vektor Ω des Input-Bildes und den Vektoren Ω_n aus dem Trainingsset. Ist die minimale Distanz ε_{min} kleiner als der Treshold Θ_{dist} , dann kann das Input-Bild einem Trainings gesicht zugeordnet werden.***

$$\varepsilon_n = \left\| (\Omega - \Omega_n) \right\| = \sqrt{(\Omega - \Omega_n)^T \cdot (\Omega - \Omega_n)}$$ Euklidische Distanz zu einer Gesichtsklasse

$$\text{mit } \varepsilon_{min} = \min_n (\varepsilon_n)$$ minimale Distanz

$$\text{und } \varepsilon_{min} < \Theta_{dist}$$

Gleichungen 3: ***Bestimmung des Abstandes zum Face Space ε aus den Differenzen des Original bildes Φ und der rücktransformierten Face Space-Projektion Φ_f .Ist das Bild nahe am Face Space ($\varepsilon < \Theta_{dffs}$), dann wird das Input-Bild als Gesicht klassifiziert.***

$$\varepsilon = \left\| (\Phi - \Phi_f) \right\|$$ Abstand zum Face Space

$$\text{mit } \Phi = \Gamma - \Psi$$ Um Mittelwert bereinigtes Input - Bild

$$\text{und } \Phi_f = \sum_{i=1}^{M} w_i \cdot u_i$$ Ruecktransformation der Face Space - Projektion

$$\text{außerdem } \varepsilon < \Theta_{dffs}$$

Turk & Pentland [TP91] berichten über die Möglichkeit der Lokalisierung von Gesichtern, indem sie das gesamte zu untersuchende Bild auf den „Face-Space“ projizieren. Da diese Projektion an jeder Pixelposition im Bild durchgeführt werden müßte, was sehr aufwendig wäre, haben die Autoren eine vereinfachte Berechnungsmethode (Berechnung der „Face-Map“) entwickelt. Die Bestimmung der Eigenvektoren kann ebenso mit einem linearen **Neuralen Netz** implementiert werden ([TP91]).

Vor kurzem veröffentlichten Pentland et al. [PMS94] eine verbesserte Version ihres Systems, das mit sogenannten „View-based Eigenspaces“ arbeitet. Bei dieser Erweiterung des herkömmlichen Ansatzes wird für jede Kombination von verschiedenen Seitenansichten und Größen der Gesichter ein unabhängiger Eigenraum (von den Eigenvektoren aufgespannter Teilraum) erzeugt. Die neu entwickelte Methode wird mit einer noch größeren Datenbasis (als in [TP91]), die aus 7562 Gesichtern von etwa 3000 Personen besteht, getestet. Der „View-based Eigenspace“-Ansatz ist erwartungsgemäß effektiver als die Repräsentation aller Ansichten durch einen einzigen Eigenraum.

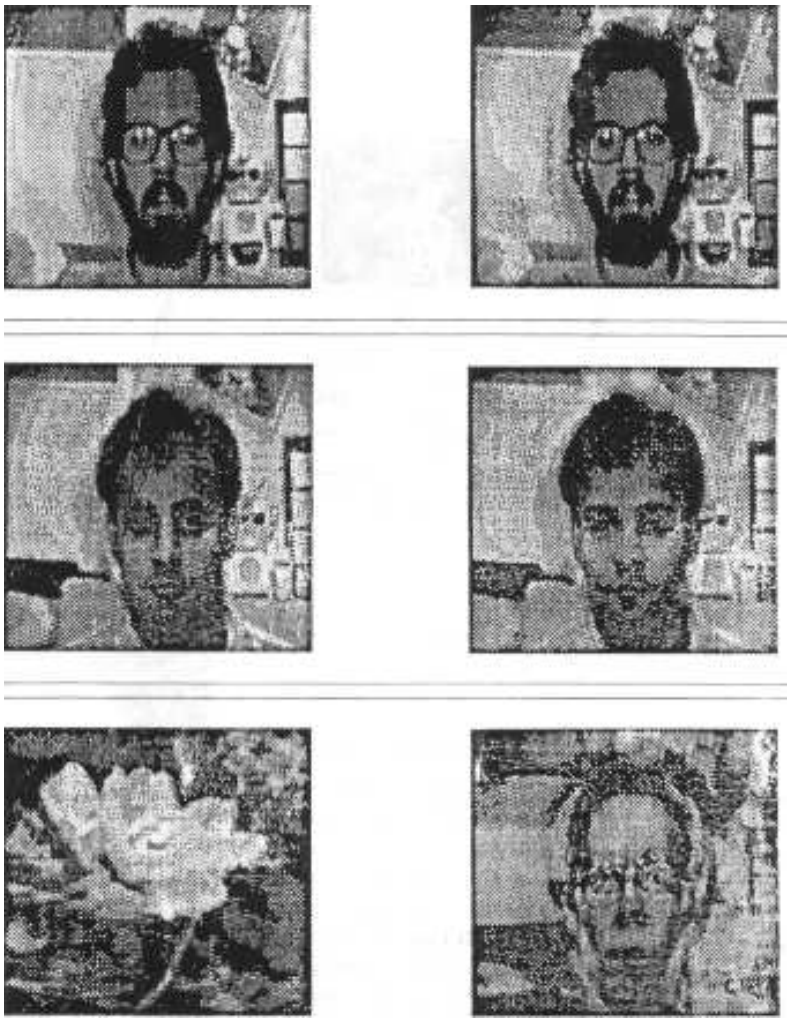

Bild 4: ***Links sind jeweils die Originalbilder und rechts die auf den „Face-Space“ projizierten und wieder rekonstruierten Bilder zu sehen. Die beiden Gesichter, die nahe am Face-Space sind ($\varepsilon < \Theta_{dffs}$), bleiben bei der Rücktransformation fast unverändert, während die Blume im rechten Bild nicht mehr erkennbar ist. (aus [TP91].)***

Neurale Netze

Neurale Netzwerke (NN) eignen sich auch zur Implementierung von einigen Ansätzen der Gesichtserkennung (wie z.B: Face Space-Projektion und Symmetrieoperatoren), indem die Gewichte durch den jeweiligen Ansatz bestimmt sind. Hier sollen jedoch nur solche NN besprochen werden, deren Gewichte sich durch eine Lernstrategie (meist Backpropagation unter Verwendung von Trainingsbeispielen) selbständig einstellen.
Bild 5 zeigt ein Beispiel für ein dreischichtiges NN (oder **Multi Layer Perceptron - MLP**) in feed-forward-Architektur mit I Input Units, J Hidden Units und K Output Units. Dieses wird mit einer Backpropagation-Lernstrategie trainiert, die sich in zwei Phasen gliedert. Zuerst werden die Aktivierungen der Inputs u_i über das Netzwerk bis zu den Outputs u_k gemäß den Gleichungen 4 vorwärts ausgebreitet (forward propagation).

Gleichungen 4: ***Die Input-Werte u_i werden über die Gewichte w_{ji} zu den Hidden-Units u_j vorwärts-propagiert und über die Gewichte w_{kj} weiter zu den Output-Units u_k . net_i , net_j sind die Netto-Inputs der Hidden- bzw. Output-Units. Aus dem jeweiligen Netto-Input wird über die sigmoide Aktivierungsfunktion f(x) der Output-Wert berechnet.***

$$net_j = \sum_i u_i \cdot w_{ji}, \quad u_j = f(net_j)$$ Netto - Input und Output der j - ten Hidden - Unit

$$net_k = \sum_j u_j \cdot w_{kj}, \quad u_k = f(net_k)$$ Netto - Input und Output der k - ten Output - Unit

$$\text{mit} \quad f(x) = \frac{1}{1+e^{-x}}$$ sigmoide Aktivierungsfunktion

In der zweiten Phase werden zuerst die Fehlersignale δ_k der Output-Units bestimmt und daraus über die Gewichte w_{kj} (backpropagation) die Fehlersignale δ_j der Hidden-Units erechnet. Danach werden aus den Fehlersignalen die Gewichtsänderungen Δw_{ji} und Δw_{kj} ermittelt (siehe Gleichungen 5).
Für die genaue mathematische Ableitung der Backpropagation sei auf Rumelhart et al. [RHW86] verwiesen.

Gleichungen 5: ***Rückpropagieren der Fehlersignale δ_k und δ_j durch Änderung der Gewichte w_{ji} und w_{kj} mit der Lernrate k. t_k ist die k-te Komponenete des Sollmustervektors und f'(x) die erste Ableitung der Aktivierungsfunktion.***

$$\delta_k = f'(net_k) \cdot (t_k - u_k)$$ Fehlersignal der k - ten Output - Unit

$$\delta_j = f'(net_j) \cdot \sum_k \delta_k \cdot w_{kj}$$ Fehlersignal der j - ten Hidden - Unit

$$\text{mit} \quad f'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = f(x) \cdot [1 - f(x)]$$ Ableitung der Aktivierungsfunktion

$$\Delta w_{ji} = k \cdot \delta_j \cdot u_i$$ Gewichtsaenderung zw. Input - und Hidden - Layer

$$\Delta w_{kj} = k \cdot \delta_k \cdot u_j$$ Gewichtsaenderung zw. Hidden - u. Output - Layer

Ein gutes Beispiel ist die Arbeit von Rowley, Baluja und Kanade [RBK95]. Sie haben ein NN-basierendes System zur Detektion von Gesichtern entwickelt, das eine Erkennungsrate von 92.9 % (bei 0.001% False Detection Rate) erreicht. Die Autoren verwenden mehrere Bildvorverarbeitungs-Mechanismen, Mehrfach-Netzwerke zur Klassifikation und einen **Bootstrapping-Algorithmus**, der False Detections aus Szenen ohne Gesichter während des Lernvorganges dem Trainingsset hinzufügt. Die ungewohnt gute Performance wird aber mit einer gegenwärtigen Detektionszeit von 120 s pro Bild (160x120 Pixel) erkauft.

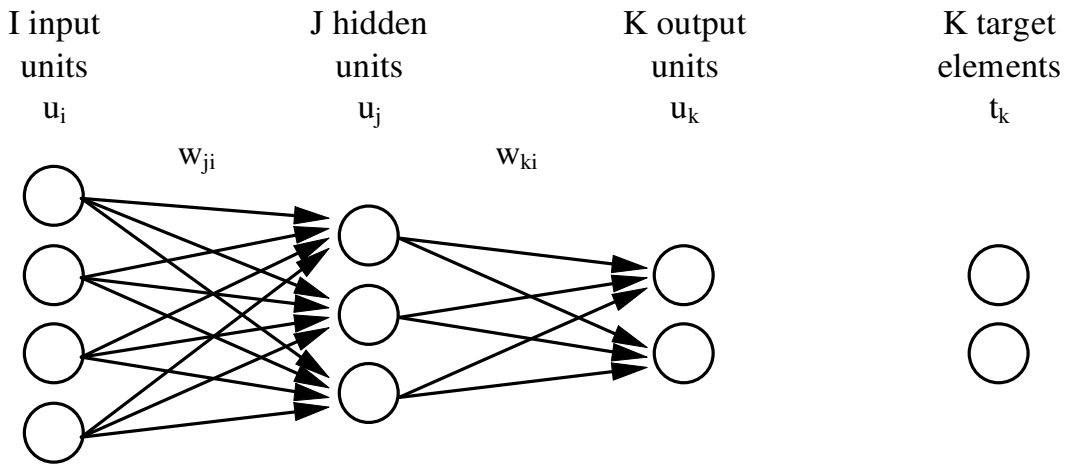

Bild 5: ***Dreischichtiges MLP (feed-forward-Topologie) mit I Input Units u_i . J HiddenUnits u_j und K Output Units u_k . Die K Sollmuster-Komponenten t_k dienen beim Trainingsvorgang dem Vergleich mit dem Output.***

Das von Sung und Poggio [SP94] entwickelte System ist eine Mischung aus einem abgewandelten Eigenface-Ansatz und NN. Aus Beipielen von Gesichtern und Nicht-Gesichtern (auch mit Bootstrapping) wird eine Verteilung mit sechs Gesichts- und sechs Nicht-Gesichts-Gruppen errechnet. Die Distanzen eines Testbildes zu diesen Gruppen werden mit einem mehrschichtigen NN (MLP) klassifiziert. Die Erkennungsraten sind annähernd so gut wie beim vorher erwähnten Ansatz von Rowley et al. [RBK95]. Der Rechenaufwand, der sich hauptsächlich aus der Bildvorverarbeitung und der Berechnung der Distanzen zusammensetzt, ist auch hier relativ hoch.

Identifikation

Im folgenden behandle ich sowohl Methoden ,die nur eine Wiedererkennung (siehe Seite 4) von Gesichtern ermöglichen, als auch solche, die darüber hinaus auch für die Identifikation geeignet sind.
Neben der Extraktion von Features aus Gesichtsprofilen, die ich hier nicht näher erläutern möchte, gab es zahlreiche Versuche **aus Intensitätsbildern (2D-Arrays) abgeleitete Features** für die Identifikation von Gesichtern zu verwenden.
Ein Beispiel dafür sind Goldstein et al. [GHL71], die bei ihren Experimenten 22 Features verwendeten, die von Menschen ausgewählt wurden. Jedes Feature wurde in 5 verschiedenen Auflösungen abgespeichert und als Vergleichskriterium diente die kleinste Euklidische Distanz zwischen dem unbekannten Gesicht und einem Gesicht aus der Datenbank. Auf ähnliche Weise versuchte Buhr [Buh86] mit 33 primären und 12 sekundären Features unter Verwendung eines linearen Entscheidungsbaumes Gesichter zu identifizieren. Auch Brunelli und Poggio [BP93] stellen eine Gesichtserkennungsmethode dieser Art vor, die sie „Feature-basierender Vergleich“ nennen.

Aufbauend auf die Ergebnisse von Baron [Bar81] entwickelten Brunelli und Poggio [BP93] eine **Schablonenvergleichs-Strategie**. Sie benutzen dabei 4 Schablonen von Augen, Nase, Mund und ganzem Gesicht, die durch die Berechnung von sogenannten normalisierten Kreuz-Korellationskoeffizienten mit dem unbekannten Gesicht verglichen werden (Bild 6). Die Autoren erwähnen, daß ihre Schablonenvergleichs-Strategie bessere Ergebnisse erzielte als der vorher erwähnte, auf Features basierende Vergleich. Jedoch ist für den Schablonenvergleich der Berechnungs- und der Speicheraufwand etwas höher.

Bild 6: ***Die aufgehellten Flächen zeigen die Form der vier von Brunelli & Poggio verwendeten Templates - Augen, Nase, Mund und ganzes Gesicht. (aus [BP93].)***

Nakamura et al. [NMM91] verwenden **Helligkeits-Isolinien (isodensity lines)**, das sind Kurven gleicher Helligkeit, zur Gesichtserkennung. Diese Isolinien dienen zur Extraktion von Konturen, die anschließend ebenfalls mit Schablonen verglichen werden.

Ein sehr guter Lösungsvorschlag kam von Turk und Pentland [TP91], die in ihrer Arbeit die schon lange bekannte Methode der **Hauptkomponentenanalyse** (PCA oder Karhunen-Loeve-Expansion, Beschreibung auf Seite 9) für die Gesichtserkennung unter Verwendung einer großen Datenbasis neu aufbereiten. Inspiriert von Kirby und Sirovich [KS90] entwickeln Turk & Pentland unter Verwendung von „Eigenfaces“ ein Gesichtserkennungs-System, das mit 2500 Gesichtern von 16 Personen getestet wird.

Jedes Gesicht wird bei allen Kombinationen von drei Kopforientierungen (aufrecht und seitliches Kippen ±45°), drei Kopfgrößen und drei verschiedenen Beleuchtungen aufgenommen. Aufgrund der damit durchgeführten Experimente treffen die Autoren Aussagen über die Robustheit ihres Ansatzes gegenüber Veränderungen des Lichtes sowie der Größe und Orientierung des Kopfes. Die Erkennungsraten werden von den Lichtverhältnissen nur wenig beeinflußt, während eine Größenänderung zu einer starken Verschlechterung der Erkennung führt. Wie schon erwähnt, kann die Berechnung der Eigenvektoren genauso mit einem linearen Neuralen Netz implementiert werden. Bild 7 zeigt die ersten 16 „Eigenfaces“, die mit Hilfe der Datenbasis von Turk & Pentland [TP91] erzeugt wurden.

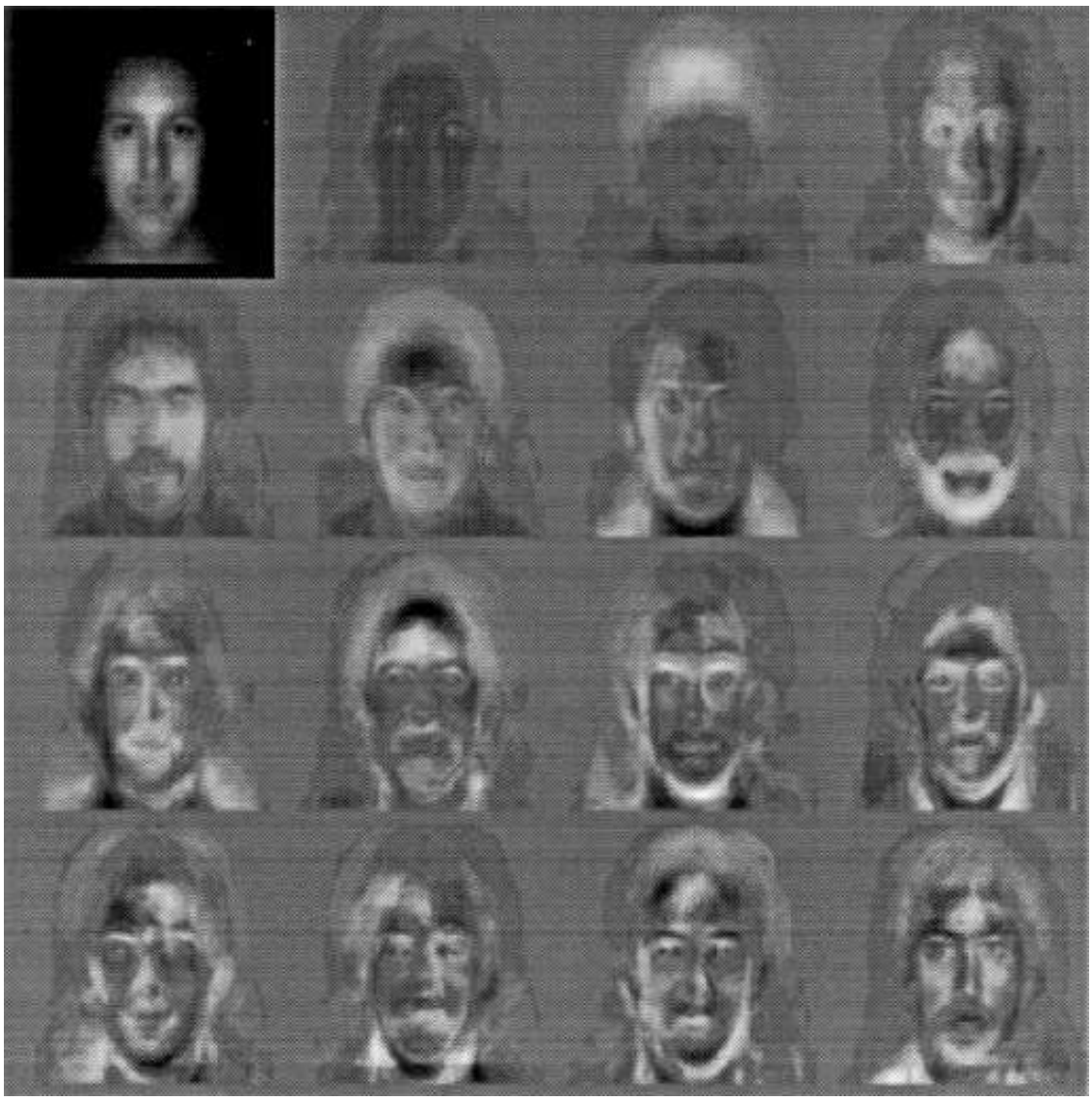

Bild 7: ***Die ersten 16 "Eigenfaces", die mit einer Datenbasis von 2500 Gesichtern mit der Software von Turk & Pentland errechnet wurden. (aus [Hra94].)***

In einer neueren Arbeit zeigen Moghaddam und Pentland [MP94], daß das Konzept der „Eigenfaces“ auf „Eigenfeatures“, wie Eigen-Augen, Eigen-Mund, usw. erweitert werden kann. Sie dokumentieren Detektionsraten von 94% für Augen, 80% für Nase und 56% für den Mund mit einer Datenbasis aus 7562 Gesichtern.
Weiters stellen die Autoren die Methode der „Modular Eigenspaces“ vor, bei der durch Kombination von „Eigenfaces“ und „Eigenfeatures“ eine modulare Representation eines Gesichtes erzeugt werden kann. Ihr System enthält zusätzlich Informationen über Geschlecht, Rasse, ungefähres Alter und Gesichtsausdruck jeder Aufnahme. Bei einer Datenmenge von 7562 Gesichtern erreichen sie eine Erkennungsgenauigkeit von über 95%. Die Autoren illustrieren damit die Einsatzmöglichkeiten ihres Systems in der Personenkontrolle.

Eine der frühesten Anwendungen von **Neuralen Netzen (NN)** in der Gesichtserkennung ist in [Koh88] zu finden. Dabei wird ein zweischichtiges auto-assoziatives Netzwerk darauf trainiert, eine kleine Anzahl von Gesichtern, selbst wenn diese verrauscht oder teilweise verdeckt sind, zu reproduzieren (Bild 8).

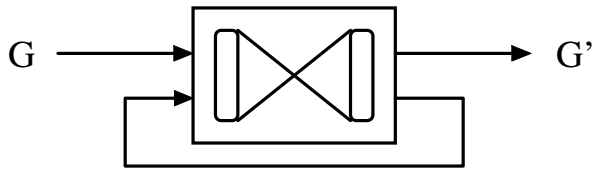

Bild 8: ***Zweischichtiges Netz zur Reproduktion von Gesichtern***

Eine weitere Einsatzmöglichkeit von NN zeigen Cottrell und Fleming [CF90]. Sie benutzen ein dreischichtiges BP-NN, das aus 4096 Input-, 80 sigmoiden (nichtlinearen) Hidden- und 4096 Output-Units besteht, zur Kompression von 64 Gesichtern und 13 andersartigen Bildern, die einer Datenbasis von 231 Bildern entnommen sind. Die 80 Hidden-Units bilden den Input für ein zweites Netzwerk, das nach dem Training mit Back-Propagation die Gesichter nach „Gesichtsähnlichkeit“, Geschlecht und Identität klassifiziert.
Ein anderes Beispiel kommt von Huang et al. [HWA93], die eine NN-ähnliche Struktur, „Cresceptron“ genannt, entwickelten. Es hat den Aufbau einer mehrfach auflösenden Bild-Pyramide und ist nach Außen hin mit Fukushima’s Neocognitron vergleichbar. Das „Cresceptron“ lernt vollautomatisch und hat bei einem kleinangelegten Versuch mit 50 Personen gute Ergebnisse erzielt.
Zu erwähnen wäre auch noch die „Dynamic Link Architecture“ (DLA), die in [LVB93] verwendet wird. Dabei wird über das Gesicht ein Gitter aus Neuronen gelegt, das sich Veränderungen von z.B. Größe, Neigung oder Gesichtsausdruck anpassen kann.

4. AUFBAU DES FACE-RECOGNITION-SYSTEMS

Aufgabenverteilung

Im Bild 9 ist der Grobaufbau des Systems dargestellt. Eine CCD-Kamera wird auf einen beweglichen Roboterarm fix montiert. Durch diese Konstruktion soll der Überwachungsraum der Kamera vergrößert werden.

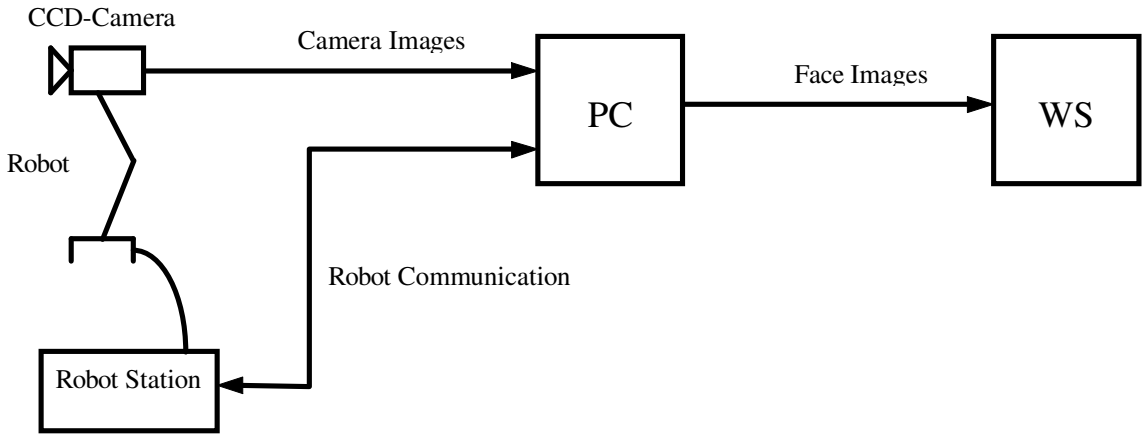

Bild 9: Grobe Darstellung der Aufgabenteilung

Während der Roboter in Bewegung ist, sollen keine Bilder von der Kamera aufgenommen werden („unechte“ Bewegungen). Erst wenn der Roboter vollständig zum Stillstand gekommen ist, wird die Kameraaufnahme fortgesetzt. Mit Hilfe der aufgenommenen Bilder sollen vom PC zuerst die „aktiven“ Bereiche (jene Teile des Bildes, in denen Bewegung detektiert wurde) bestimmt und diese anschließend auf das Vorkommen von Gesichtern durchsucht werden. Die gefundenen Gesichtsbilder werden an die Workstation (WS) zur Identifikation weitergeleitet. Der Name des identifizierten Gesichtes wird auf der WS ausgegeben.
Daher gliedert sich meine Diplomarbeit in drei Aufgabenbereiche: Roboter & Kamera, Detektion am PC und Identifikation auf der Workstation.

Funktionsbeschreibung der einzelnen Module

Bild 10 zeigt ein Prozeßablauf-Schema, in dem das Gesichtserkennungs-System in einzelne Module zerlegt worden ist. Die von der Kamera aufgenommenen Bilder können so Schritt für Schritt analysiert und letztendlich auf der WS identifiziert werden.

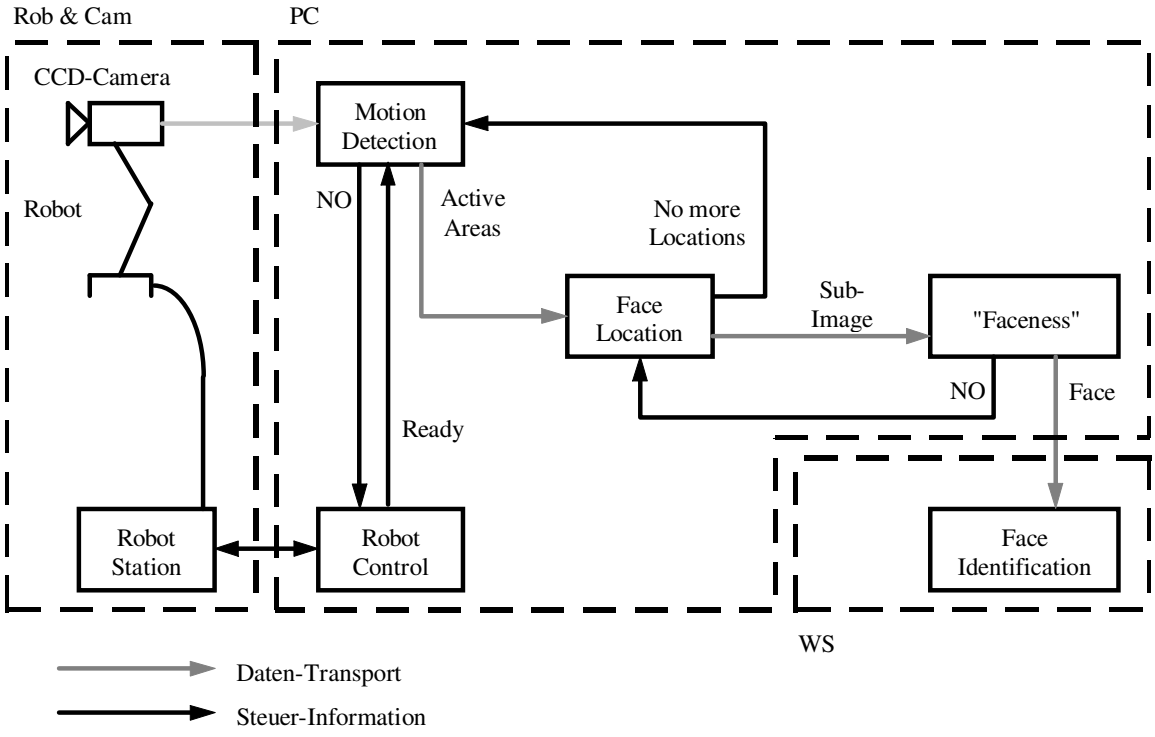

Bild 10: Prozeßablauf-Diagramm

Gesichts-Detektion am PC

In der **Motion Detection** werden mittels der von der Kamera gelieferten Bildinformation aus dem gerade betrachteten Teilbereich des Überwachungsraumes die „aktiven" Regionen herausgefiltert. Wird keine Bewegung festgestellt, so wird die **Robot Control** angewiesen, den Roboter und damit auch die Kamera in eine andere Richtung zu bewegen. Nachdem der Roboter zum Stillstand gekommen ist, wird abermals die Motion Detection ausgeführt. Das geschieht solange, bis in einem Teilraum bewegte Objekte festgestellt werden. Danach sollen aus den „aktiven" Regionen alle Teilbilder, die Gesichter sein könnten (Grobselektion), herausgeschnitten und nacheinander an das **„Faceness"-Modul** weitergeleitet werden. Dieses Modul klassifiziert die Teilbilder nach ihrer „Gesichtsähnlichkeit" („faceness"), d.h. es stellt fest, ob es sich um ein Gesicht oder um ein anderes Objekt handelt. Die erfolgreich klassifizierten Gesichtsbilder werden an die WS gesendet.

Gesichts-Identifikation auf der WS

Die vom PC empfangenen Gesichtsbilder sollen in der **Face Identification** durch eine vorher erlernte Datenbasis wiedererkannt und durch Ausgabe des jeweils dazugehörigen Namens identifiziert werden.

Bestimmung der Auswahlkriterien

In diesem Abschnitt möchte ich jene Merkmale festlegen, mit denen die Methoden im Lokalisierungs- und Klassifikationsmodul verglichen werden sollen. Wie im Bild 11 dargestellt, nimmt von der Bewegungserkennung bis zur Identifikation die Komplexität der Algorithmen ständig zu, während die jeweils betrachteten Bereiche des von der Kamera gelieferten Bildes immer kleiner werden (bzw. deren Anzahl abnimmt). Einerseits ergibt sich also ein größerer Zeitaufwand durch komplexere Verfahren, andererseits jedoch müssen damit weniger Informationen verarbeitet werden.

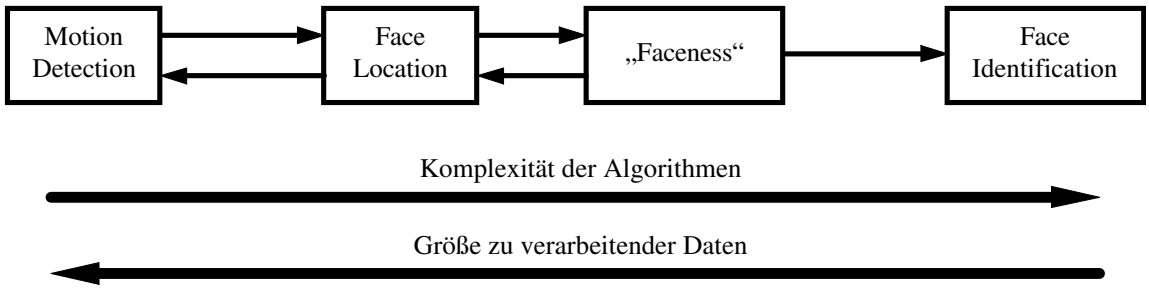

Bild 11: Betrachtungen der einzelnen Module in Bezug auf Komplexität und Informationsmenge

Aus diesen Überlegungen ergeben sich die folgenden drei Merkmale, die für jedes Modul betrachtet werden müssen :

1. **Zeit** - mit der Komplexität des Algorithmus' steigt auch der Zeitaufwand
2. **Erkennungsrate (Gesichter)** - komplexere Algorithmen erreichen meist höhere Erkennungsraten
3. **Datenreduktion** - je mehr Nicht-Gesichter ausgesondert werden können, desto weniger Daten muß das nächste Modul bearbeiten

Natürlich stehen diese Kriterien in enger Beziehung zueinander. So kann man z.B. bessere Erkennungsraten erzielen, indem man einen höheren Zeitaufwand oder eine geringere Datenreduktion in Kauf nimmt. Ersteres hängt direkt mit der Komplexität der Algorithmen zusammen. Im zweiten Fall steigt die sogenannte „False Positive Rate“ (= Anteil an falsch erkannten Nicht-Gesichtern), während die „False Negative Rate“ (= Anteil an nicht erkannten Gesichtern = 1 - Erkennungsrate für Gesichter) sinkt, d.h. also, daß bei geringerer Datenreduktion weniger Gesichter verloren gehen. Tabelle 1 zeigt eine Übersicht über die verschiedenen Erkennungsraten, die sich durch die Kombination von tatsächlichen und klassifizierten Werten ergeben.

Durch Abwägung dieser drei Faktoren für jedes Modul soll schließlich die Erkennungsrate des Systems optimiert werden. Zum besseren Verständnis möchte ich hier auch ein praktisches Beispiel anführen :

		classified as	
		F	**¬ F**
real class	**F**	recognition rate of faces	„False Negative Rate“
	¬ F	„False Positive Rate“	recognition rate of non-faces

Tabelle 1: Einteilung der Erkennungsraten in Bezug auf die Klassifikation von Gesichtern (F) und Nicht-Gesichtern (¬ F)

In Tabelle 2 sind die Werte zweier (fiktiver) Methoden gegenübergestellt. Die zweite Methode hat zwar eine schlechtere Erkennungsrate für präsentierte Gesichter (ist dafür auch schneller) als die erste Methode, dafür können im gleichen Zeitraum (Zeit zur Bearbeitung von einem Frame mit Methode 1) fünf Frames hintereinander betrachtet werden und dadurch wird eine bessere „Overall-False Positive Rate“ von $0.5^5 \cong 0.031$ erreicht.

	recognition rate of faces	**time per frame**	**FalsePositiveRate**
method 1	0.9	1.0	0.1
method 2	0.5	0.2	0.5

Tabelle 2: Charakteristika zweier fiktiver Methoden

Aus diesem einfachen Beispiel sieht man, daß zur Optimierung der Gesamt-Performance des Systems sowohl die Zeit als auch die Erkennungsraten der Algorithmen berücksichtigt werden müssen, und daß es durchaus von Vorteil sein kann, einen schnellen Algorithmus mit hoher Fehlerrate einem langsamen mit hoher Erkennungsrate vorzuziehen.

5. AUSWAHL UND IMPLEMENTIERUNG DER VERFAHREN FÜR DIE EINZELNEN MODULE

In diesem Teil der Diplomarbeit werde ich die in Kapitel 3 besprochenen Gesichtserkennungs-Methoden in Bezug auf die Verwendbarkeit in den verschiedenen Modulen untersuchen. Jene Methoden, die für das System als besonders geeignet erscheinen, werden einer genaueren Prüfung unterzogen. Schließlich soll die endgültige Ausführung jedes Moduls gezeigt werden.
In der Tabelle 3 sehen wir eine übersichtlich dargestellte Einteilung der vorgestellten Methoden nach ihrer Verwendbarkeit in den einzelnen Modulen.

PC Motion Detection	**PC Face Location**	**PC „Faceness“**	**WS Face Identification**
• *Optical Flow*-Methoden	• *Regelbasierte Methoden*		• *Feature-Based Matching*
• *Motion Energy Detection*	• *Template Matching* : universelle Schablonen		• *Template Matching* : spezielle Schablonen
	• *Symmetrieoperat.* : niedrige Auflösung	• *Symmetrieoperat.* : hohe Auflösung	
	• *Eigenfaces* : „Face Space“		• *Eigenfaces* : Nähe zu Gesichtsklasse
		• *Nichtlineares BP-Netz*	• *Neurale Netze*

Tabelle 3: Aufstellung über die im System verwendbaren Methoden

Motion Detection

In die Kategorie der Bewegungserkennung fallen zwei Arten von Methoden. Erstens gibt es die **Optical Flow-Methoden** (siehe [MB94]). Sie sind extrem rechenaufwendig und liefern mehr Information als tatsächlich benötigt wird. Aufgrund des Rechenaufwandes können diese Methoden in einem System, das annähernd in Echtzeit funktionieren soll, natürlich nicht verwendet werden.
Die zweite Art ist die sogenannte **Motion Energy Detection** ([MB94]). Es handelt sich dabei um eine sehr schnelle Methode (besonders in Verbindung mit Bildverarbeitungs-Hardware) und ist daher hervorragend für mein System geeignet. Die Funktionsweise dieser Methode wurde schon einführend in Kapitel 3 beschrieben.

Im Zuge dieses Moduls kommt eine einfache Subtraktion aufeinanderfolgender Bilder (Frames) zur Anwendung (siehe Gleichung 6). Wie eigene Versuche mit 100 Subtraktionsbildern, die mit dem Gesichtserkennungs-System aufgenommen wurden, gezeigt haben, ist die Positioniergenauigkeit dieser Methode mit ± 2 Pixel (in Bezug auf die höchste Auflösung, die verarbeitet werden muß) bei der Detektion von gehenden Menschen völlig ausreichend.

Gleichung 6: ***Motion Energy Detection - $d_t(x,y)$ ist die Differenz von zwei Intensitätsbildern $I_{t-1}(x,y)$ und $I_t(x,y)$, die zu den Zeitpunkten t-1 und t aufgenommen wurden.***

$$d_t(x,y) = I_{t-1}(x,y) - I_t(x,y)$$

Um festzustellen, ob überhaupt eine signifikante Bewegung in der Szene stattgefunden hat, wird die Summe B der Differenzpixel des gesamten Bildes berechnet. Eine Bewegung wird erst detektiert, wenn dieser Wert B größer als der Schwellwert Θ_{mot} wird (Gleichungen 7). Dieser Schwellwert Θ_{mot} wird so gewählt, daß sich noch eine hohe Empfindlichkeit ergibt.

Gleichungen 7: ***Wenn die Summe B aller Differenzen $d_t(x,y)$ größer als der Schwellwert Θ_{mot} ist, dann wird eine Bewegung detektiert.***

$$B = \sum_{x,y} \left| d_t(x,y) \right| \qquad \text{und} \qquad B > \Theta_{mot}$$

Anschließend wird auf das Differenzbild eine Schwellwertoperation unter Verwendung von Θ_{bin} angewendet (Bild 12 links). Θ_{bin} wird so hoch eingestellt, daß durch Rauschen entstandene Differenzen vollständig verschwinden. Dieser Threshold kann konstant gehalten werden, weil die Kamera mit einer automatischen Verstärkungskontrolle arbeitet und somit Lichtveränderungen, die das Rauschen beeinflussen, in einem großen Bereich ausgleichen kann. Der Schwellwert kann aber auch automatisch eingestellt werden, indem er von der Anzahl der detektierten Punkte abhängig gemacht wird. D.h., daß Θ_{bin} kleiner gemacht wird, wenn zu viele Punkte detektiert werden und wieder vergrößert wird, wenn die Anzahl der Punkte abnimmt.
Dieses erhaltene Motion Energy Bild wird einer 2x2/4-Maximum-Pyramide (Auswahl des höchsten Grauwertes, der durch das Thresholding nur schwarz oder weiß sein kann) unterworfen, was zu einer Aufteilung der Szene in aktive und inaktive Bereiche in verschiedenen Auflösungen führt (Bild 12 rechts).
Alle hier besprochenen Bildoperationen werden auf spezieller Bildverarbeitungs-Hardware (Matrox-Board) realisiert.

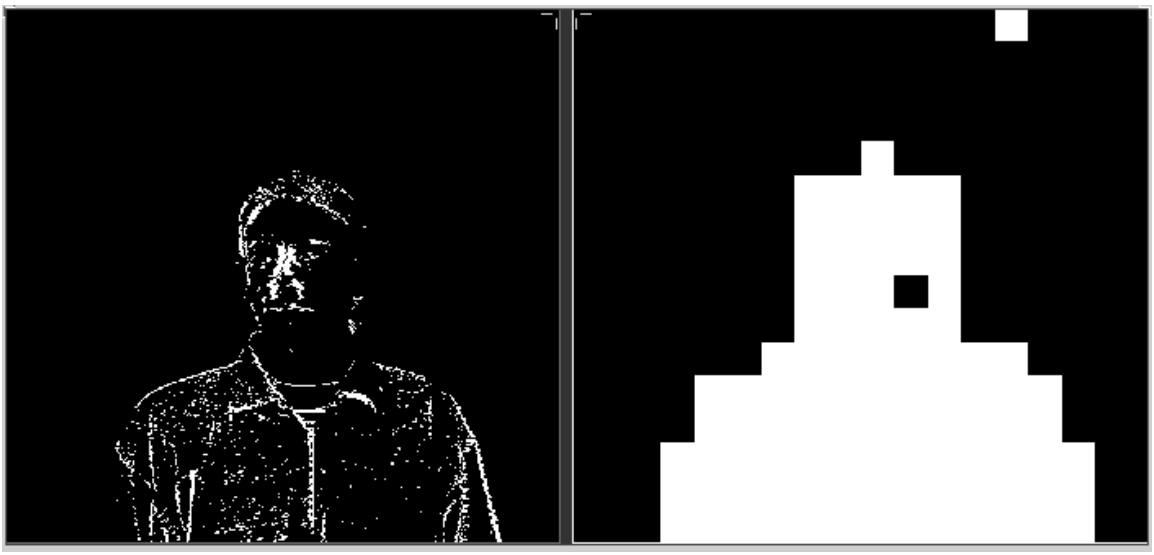

Bild 12: ***Links - Das Subtraktionsbild zweier aufeinanderfolgender Frames nach dem Thresholding. Rechts - Dasselbe Bild nach Anwendung einer 2x2/4-Maximum-Pyramide bis zu einer Auflösung von 16x16. Aktive Bereiche sind weiß und inaktive schwarz dargestellt.***

„Faceness“

Aufgrund der geforderten hohen Verarbeitungsgeschwindigkeit in diesem Modul ist die Verwendung von den relativ rechenaufwendigen **Symmetrieoperatoren** nicht zielführend.
Die nächsten beiden Möglichkeiten bilden das Template Matching und die Eigenfaces. Wie schon erwähnt, muß beim **Template Matching** eine geeignete Schablone, die auf möglichst viele Gesichter paßt, gefunden werden. Diese meist händisch bestimmte Schablone kann dann mit präsentierten Bildteilen verglichen werden.
Ähnlichkeiten zu dieser Technik kann man auch beim **Eigenface-Ansatz** feststellen. Der eigentliche Unterschied liegt darin, daß anstatt einer Schablone M Eigenfaces verwendet werden, die automatisch unter Verwendung eines Trainingsets berechnet werden können. Der Vergleich sieht hier so aus, daß mit Hilfe der errechneten M Eigenfaces u_m das präsentierte Bild auf den Face Space projiziert und anschließend die Differenz des rückprojizierten Bildes zum Originalbild berechnet wird. Je kleiner also der Abstand ε zum Face Space ist, desto „gesichtsähnlicher“ ist das Bild (siehe Gleichungen 3).
Die Entscheidung zwischen diesen beiden Ansätzen fiel zugunsten der Eigenface-Methode aus, weil erstens nach der Berechnung der Eigenfaces jede Projektion mit vergleichsweise geringem Rechenaufwand durchgeführt werden kann und zweitens die umständliche (händische) Bestimmung einer allgemeinen Schablone wegfällt (siehe auch [SP94]).
Eine weitere Möglichkeit zur Bestimmung der „Faceness“ ist ein (mind. dreischichtiges) Neurales Klassifikations-Netzwerk oder auch **MLP** (Multi Layer Perceptron, [LMN92]).

Auch diese Methode ist aufgrund ihrer leichten Handhabbarkeit (automatisches Lernen) und dem verbleibenden geringen Rechenaufwand nach der Trainingsphase von mir als Kandidat für das „Faceness“-Modul ausgewählt worden. Für den Vergleich verwende ich ein dreischichtiges Netz. Die Units des Hidden-Layers stellen eine komprimierte Darstellung des Trainingssets dar, mit denen dann eine Klassifikation nach „Gesicht“ oder „Kein Gesicht“ erfolgen kann.

Vergleich zwischen Eigenface-Ansatz und MLP

Die beiden verbleibenden Methoden werden in Bezug auf ihren Rechenaufwand (Auswahlkriterium Zeit) und ihre Erkennungsraten bei der Bestimmung der „Faceness“ von präsentierten Bildern verglichen.
Da dieses Modul in einem annähernden Echtzeit-System arbeiten soll, ist für mich nur der Rechenaufwand der Testseterkennung (unabhängig von der Trainingszeit) beider Ansätze ausschlaggebend.
Beim Pentland-Ansatz kann die Berechnung der Eigenfaces und die Projektion eines Bildes auf den Face-Space ebenso mit einem neuralen Netzwerk erfolgen, das im Gegensatz zum MLP jedoch nur lineare Units enthält. Die Gewichte zwischen dem Input- und dem Hidden-Layer entsprechen dabei den Eigenfaces und die Hidden-Units dem Mustervektor, der den Abstand des Eingabe-Bildes zum Face-Space beschreibt.
Abgesehen vom etwas höheren Aufwand der Rückprojektion beim Eigenface-Ansatz im Vergleich zur Klassifikation beim MLP (weniger Output-Units) kann der Berechnungsaufwand für beide Methoden als proportional zu der Anzahl der verwendeten Eigenfaces bzw. Hidden Units angenommen werden (Bild 13). Deshalb werden die Vergleiche zwischen dem Eigenface-Ansatz und MLP jeweils bei der gleichen Anzahl von Eigenfaces bzw. Hidden-Units durchgeführt.
Um die anderen beiden direkt zusammenhängenden Auswahlkriterien Erkennungsrate (Gesichter) und Datenreduktion (Erkennungsrate für Nicht-Gesichter) zu berücksichtigen wird die gemessene Erkennungsrate für das Testset nach Gleichung 8 berechnet.

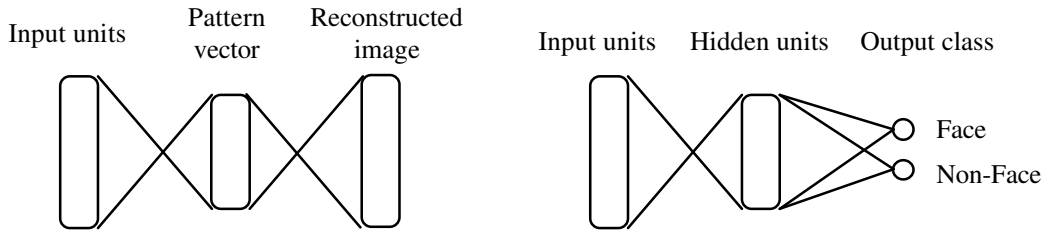

Bild 13: ***Vergleich der Berechnungsschritte zwischen Eigenface-Ansatz und MLP. Links - Beim Eigenface-Ansatz entsprechen die Gewichte von den Input-Units zu den linearen Hidden-Units den Eigenfaces. Aus dem berechneten Mustervektor (in den Hidden-Units) kann durch Rückprojektion leicht der Abstand ε zum Face Space bestimmt werden. Rechts - Beim MLP wird mit Hilfe der aus dem Input errechneten Werte der nichtlinearen Hidden-Units eine Klassifikation nach Gesicht und Nicht-Gesicht durchgeführt.***

Gleichung 8: Berechnung der Gesamt-Erkennungsrate aus den gemessenen Werten

$$\text{recognition rate test set} = \frac{\#\text{recognized faces} + \#\text{recognized non-faces}}{\#\text{faces} + \#\text{non-faces}}$$

Diese Berechnungsmethode der Erkennungsrate beurteilt die Anzahl der richtig erkannten Testbeispiele „als Ganzes“ und stellt somit ein Maß für die Klassifikationsgenauigkeit des jeweiligen Ansatzes bezüglich der „Faceness“ dar.

Datenbasis für den Vergleich

Für diese Vergleichsmessungen wurden die aus 16 verschiedenen Gesichtern bestehende Datenbasis von Turk & Pentland [TP91] verwendet. Die Gesamtanzahl der Bilder entspricht 432, die durch die Kombination von drei Neigungen, drei Skalierungen und drei Belichtungen für jedes Gesicht erreicht wird (Bild 14). Die Lage der Gesichter im Bildrahmen ist dabei so normiert, daß der Mittelpunkt der Augenachse genau in der Mitte des Bildes zu liegen kommt.
Aus dieser Datenbasis verwendete ich 50% von zufällig ausgewählten Bildern als Trainingsset und den Rest als Testset. Das heißt, daß eine bestimmte Ansicht eines Gesichtes nur entweder im Trainings- oder im Testset vorkommen kann.
Zusätzlich mußte ich eine Datenbasis aus Nicht-Gesichtern aufbauen. Diese setzt sich aus 28 Originalen zusammen, die durch Spiegelungen (Bilder sind unsymmetrisch zu den Spiegelachsen) und Generierung von jeweils vier Bildausschnitten auf eine Gesamtanzahl von 384 Nicht-Gesichtern gebracht wurden (Bild 15).
Auch hier wurden jeweils 50% der Bilder der Datenbasis durch zufällige Auswahl auf das Trainings- bzw. Testset aufgeteilt.
Um die Messungen bei verschiedenen Auflösungen durchführen zu können, wurde eine 3x3/4-Gauß-Bildpyramide verwendet, die Bildgrößen von 128x120 bis hinunter zu 16x15 abdeckt.

Verwendete Sofware

Zur Bestimmung der „Faceness“ nach dem *Eigenface-Ansatz* wurde die auf dem Institut vorhandene Software von Turk & Pentland [TP91] benutzt. Alle Rechendurchgänge der Pentland-Software wurden auf einer SUN-SPARC-Server-Station durchgeführt. Die leichten Differenzen bei den Trainingszeiten trotz gleicher Bedingungen rühren daher, daß von diesem Rechner auch andere Workstations des Instituts versorgt werden müssen.
Der *MLP-Ansatz* wurde auf dem Xerion-Netzwerksimulator implementiert, der am Institut auf einer DEC-Server-Station installiert ist. Die Einsatz eines anderen Rechners beim Vergleich ist in diesem Fall unproblematisch, weil die Trainingszeiten keinen Einfluß auf die Auswertung der Ergebnisse haben.

Bild 14: ***Alle Ansichten einer Person aus der Pentland-Datenbasis ([TP91]), die für Vergleich herangezogen wurde.***

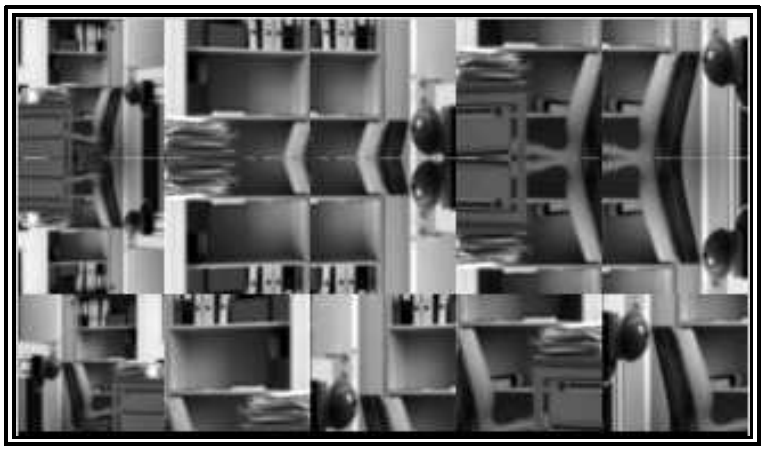

Bild 15: ***Erzeugung mehrerer „Nicht-Gesichter“ für den Vergleich aus einem aufgenommenen Original (links oben).***

Testmodalitäten

Für den *Eigenface-Ansatz* wird zuerst das Training mit dem Gesichter-Trainingsset durchgeführt. Die Anzahl der erkannten Gesichter bzw. Nicht-Gesichter kann dann durch Evaluierung der Testsets für Gesichter bzw. Nicht-Gesichter bestimmt werden. Dabei wird im ersten Fall jedes Bild, dessen Abstand ε zum Face Space (distance from face space = dffs) kleiner als ein bestimmter Schwellwert Θ_{dffs} ist, als Gesicht klassifiziert. Im zweiten Fall hingegen werden jene Bilder als Nicht-Gesichter erkannt, für die gilt, daß $\varepsilon > \Theta_{dffs}$. Da bei Erhöhung des Schwellwertes Θ_{dffs} die Anzahl der erkannten Gesichter steigt, aber zugleich die Anzahl der erkannten Nicht-Gesichter sinkt, wurde Θ_{dffs} jeweils so gewählt, daß die Gesamt-Erkennungsrate (siehe Gleichung 8) am größten ist.
Das Trainingsset und das Testset für das *MLP* setzt sich sowohl aus den Gesichtern als auch den Nicht-Gesichtern zusammen. Hier wird der Wert der Gesamt-Erkennungsrate, die ebenfalls nach Gleichung 8 bestimmt wird, nach dem Training direkt vom Xerion-Netzwerksimulator ausgegeben. Die Abbruchbedingung der Trainingsphase war jeweils das Maximum der Gesamt-Erkennungsrate des Testsets (diese sinkt wieder, wenn man zu lange trainiert).

Messung 1:

Die Pentland-Software wird mit dem Trainingsset dreier Bildgrößen (64x60, 32x30, 16x15) bei unterschiedlicher Anzahl von Eigenfaces trainiert und anschließend die Erkennungsrate des dazugehörigen Testsets ermittelt (Tabelle 4).
Die besten Ergebnisse wurden bei einer Auflösung von 16x15 erzielt. Dadurch wird die Theorie bestärkt, daß zur Gesichtserkennung nicht die Details, sondern vielmehr die Lage der einzelnen Features zueinander maßgebend sind. Durch eine weitere Erhöhung der Anzahl der Eigenfaces könnte die Erkennungsrate noch verbessert werden, jedoch wäre der daraus folgende hohe Rechenaufwand für dieses Modul nicht mehr vertretbar.
In Tabelle 5 sind die einzelnen Erkennungsraten der besten Pentland-Konfiguration nocheinmal aufgeschlüsselt.

Messung 2:

Das hier verwendete MLP besteht aus drei Schichten, wobei der Output-Layer zwei Units (Gesicht, Nicht-Gesicht) beeinhaltet (Bild 13). Die Menge der Input-Units ergibt sich aus der Bildgröße (jedes Pixel entspricht einer Unit). Die Anzahl der Hidden-Units wird ebenfalls variiert (Tabelle 6). Auch hier ist ersichtlich, daß bei einer Größe von 16x15 außer bei fünf Hidden-Units (zuwenig Freiheiten zum Generalisieren) die besten Erkennungsraten erzielt werden. Da die Trainingszeit bei einer Bildgröße von 64x60 mit über zwei Stunden sehr hoch war und trotzdem keine bessere Performance als mit den niedrigeren Auflösungen erzielt werden konnte, wurde die Erkennungsrate nur bei zehn Hidden-Units gemessen.
In Tabelle 7 sind die einzelnen Erkennungsraten des besten Netzwerkes nocheinmal aufgeschlüsselt.

PENTLAND

# eigenfaces	picture size (pixels)	training time (min:sec)	threshold Θ_{dffs}	recognition rate test set (%)
5	64x60	4:40	6200	82.7
	32x30	4:20	5500	87.0
	16x15	3:30	4500	92.2
10	64x60	9:40	5200	87.2
	32x30	4:50	4400	91.0
	16x15	3:50	3400	94.5
16	64x60	8:20	4700	86.5
	32x30	5:00	3800	89.5
	16x15	3:40	2700	94.5
20	64x60	7:20	4300	88.7
	32x30	4:50	3600	91.5
	16x15	4:00	2400	96.2

Tabelle 4: Testset-Erkennungsraten der Pentland-Software bei unterschiedlicher Anzahl Eigenfaces und Bildgrößen. Der Threshold Θ_{dffs} wurde jeweils so gewählt, daß die Erkennungsrate am größten ist.

Recognition rates of best Pentland (%, (#))		**classified as** G	**classified as** ¬ G
real class	**G (206)**	**98.1 (202)** (erkannte Gesichter)	**1.9 (4)** („False Negative Rate“)
	¬ G (193)	**5.7 (11)** („False Positive Rate“)	**94.3 (182)** (erkannte Nicht-Gesichter)

Tabelle 5: Darstellung aller Erkennungsraten der besten Pentland-Konfiguration (20 Eigenfaces, 16x15). In Klammern steht die jeweilige Anzahl der klassifizierten Bilder.

Vergleich von Messung 1 mit Messung 2:

Die Testset-Erkennungsraten von Pentland-Ansatz und MLP bei gleicher Anzahl von Eigenfaces bzw. Hidden-Units sind in Tabelle 8 sowie in Bild 16 gegenübergestellt. Es zeigt sich, daß das MLP bei der Bestimmung der „Faceness“ unter Verwendung der gemischten Datenbasis dem Eigenface-Ansatz durchgehend überlegen ist. Weiters ist zu erkennen, daß eine Auflösung von 16x15 zur Klassifikation nach der „Faceness“ völlig ausreicht.

MLP				
# hidden units	**picture size (pixels)**	**training time (min)**	**iterations**	**recognition rate test set (%)**
5	64x60	-	-	-
	32x30	20	56	99.3
	16x15	4	58	99.0
10	64x60	130	46	96.5
	32x30	22	31	97.7
	16x15	7	70	98.8
16	64x60	-	-	-
	32x30	28	30	98.3
	16x15	6	43	98.5

Tabelle 6: ***Testset-Erkennungsraten des MLP bei unterschiedlicher Anzahl von Hidden-Units und untersch. Bildgrößen.***

Recognition rates of best MLP (%, (#))		**classified as**	
		F	**¬ F**
real class	**F (206)**	99.5 (205)	0.5 (1)
	¬ F (193)	1.0 (2)	99.0 (191)

Tabelle 7: ***Darstellung aller Erkennungsraten des besten MLP (5 hidden, 32x30). In Klammern steht die jeweilige Anzahl der klassifizierten Bilder. F bedeutet Gesicht.***

PENTLAND versus MLP							
recognition rates test set (%)		**# eigenfaces resp. hidden units**					
		5		**10**		**16**	
		PENT	**MLP**	**PENT**	**MLP**	**PENT**	**MLP**
picture size (pixels)	**64x60**	-	-	87.2	96.5	-	-
	32x30	87.0	99.3	91.0	97.7	89.5	98.3
	16x15	92.2	98.8	94.5	98.8	94.5	98.5

Tabelle 8: ***Vergleich der Testset-Erkennungsraten von Pentland-Software und MLP bei verschiedenen Bildgrößen und versch. Anzahl von Eigenfaces bzw. Hidden Units. Trainingsset = 50% aus gemischter Datenbasis, Testset = restl.50%.***

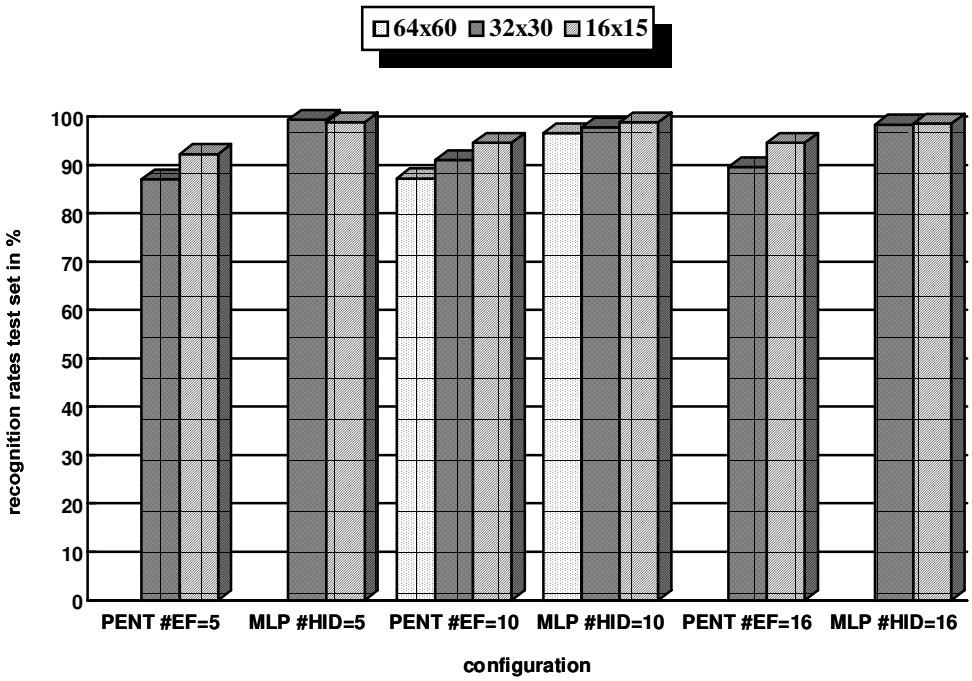

Bild 16: ***Grafischer Vergleich der Testset-Erkennungsraten (in %) von Pentland-Software und MLP bei verschiedenen Bildgrößen und versch. Anzahl von Eigenfaces bzw. Hidden Units.***

Messung 3:

Mit dieser Messung soll die Sensibilität der Pentland-Software auf nicht genau zentrierte Eingabebilder in Bezug auf die „Faceness"-Bestimmung untersucht werden. Die neuen Trainingssets entstehen durch Verschiebung des Ausgangssets bei einer Auflösung von 128x120 um jeweils +1, +2 u. +4 in X- u. Y-Richtung. Die zusätzlichen Randpunkte werden durch Nachbarpixel ersetzt. Das eigentliche Training findet dann bei einer Auflösung von 32x30 statt, sodaß sich in dieser Pyramidenebene Verschiebungen von +0.25, +0.5 u. +1 ergeben (Tabelle 9).
Auffallend ist hier, daß die Performance umso mehr von der Verschiebung beeinträchtigt wird, je größer die Anzahl der verwendeten Eigenfaces ist. Der Grund dafür liegt wahrscheinlich darin, daß durch eine höhere Anzahl von Eigenfaces auch die (zentrierte) Position des Trainingssets exakter repräsentiert wird.

Messung 4:

Zur Bestimmung der Sensibilität auf Verschiebungen wird dasselbe MLP wie in Messung 2 verwendet. Die Tabelle 10 zeigt deutlich, daß sich beim neuralen Netz eine Verschiebung der Trainingsbilder bis zu einem Pixel (bei einer Auflösung von 32x30) in X- u. Y-Richtung nur unerheblich auf die Testset-Erkennungsraten auswirkt, da das MLP viel flexibler als die Pentland-Software auf Korrelationsunterschiede benachbarter Punkte reagiert.
In Bild 17 sind die Messungen 3 und 4 nocheinmal grafisch gegenübergestellt.

PENTLAND				
# eigenfaces	**translation on 32x30 (pixels)**	**training time (min:sec)**	**threshold Θ_{dffs}**	**recognition rate test set (%)**
5	0.00	4:20	5500	87.0
	0.25	4:20	5500	86.2
	0.50	5:50	5500	85.5
	1.00	5:10	5700	82.7
10	0.00	4:50	4400	91.0
	0.25	4:30	4500	90.5
	0.50	4:40	4600	90.0
	1.00	4:10	4800	84.5
16	0.00	5:00	3800	89.5
	0.25	5:10	3800	89.2
	0.50	5:20	3800	87.7
	1.00	5:30	4200	81.7

***Tabelle 9:** Testset-Erkennungsraten der Pentland-Software bei einer Bildgröße von 32x30 und unterschiedlichen Translationsweiten. Zum Vergleich ist auch die Performance ohne Translation dargestellt. Der Threshold Θ_{dffs} wurde jeweils so gewählt, daß die Erkennungsrate am größten ist.*

MLP				
# hidden units	**training time (min)**	**iterations**	**translation on 32x30 (pixels)**	**recognition rate test set (%)**
5	20	56	0.00	99.3
			0.25	98.8
			0.50	98.8
			1.00	98.0
10	22	31	0.00	97.7
			0.25	97.7
			0.50	98.0
			1.00	97.5
16	28	30	0.00	98.3
			0.25	98.3
			0.50	97.7
			1.00	97.5

***Tabelle 10:** Testset-Erkennungsraten des MLP bei einer Bildgröße von 32x30 und unterschiedlichen Translationsweiten. Zum Vergleich ist auch die Performance ohne Translation dargestellt.*

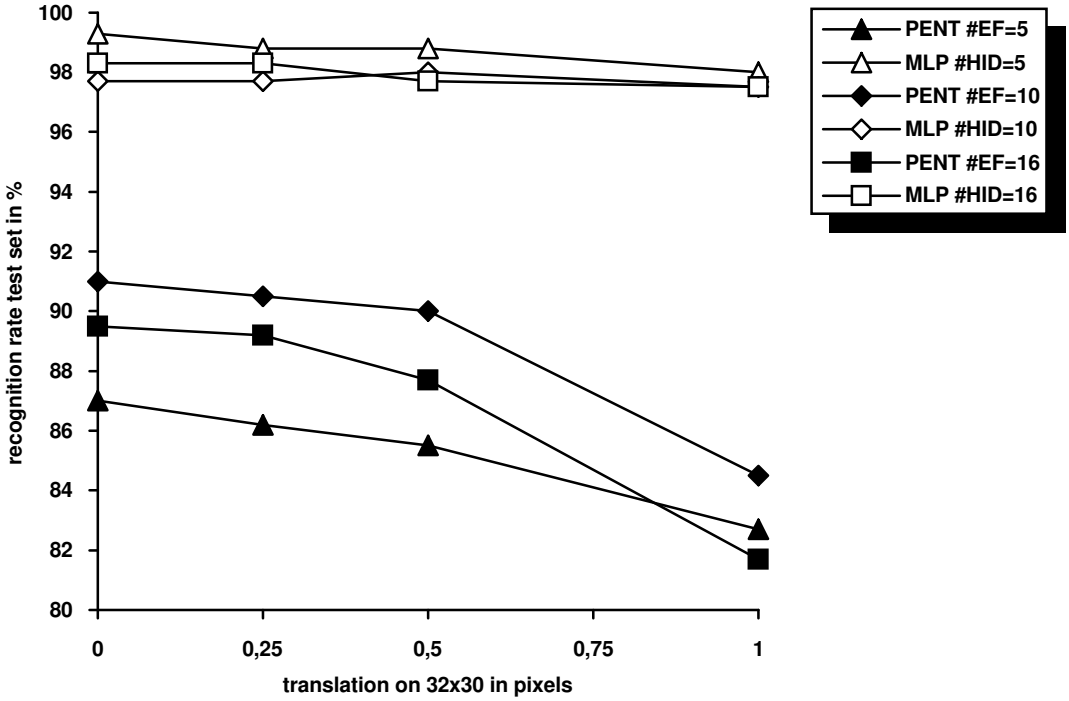

Bild 17: ***Grafischer Vergleich der Testset-Erkennungsraten von Pentland-Software und MLP bei einer Bildgröße von 32x30 unter Variation der Translationsweite und der Anzahl von Eigenfaces bzw. Hidden Units.***

Messung 5:

Das Ziel dieser Messung ist, die Reaktion der beiden Ansätze auf völlig fremde Gesichter zu beobachten, indem neun der sechszehn Gesichter aus der Datenbasis das Trainingsset bilden und mit den restlichen sieben Gesichtern getestet wird (Tabelle 11).
Auch hier ist das MLP der Pentland-Software durchgehend überlegen, was die hohe Generalisierungsfähigkeit von neuralen Netzen bestätigt.

PENTLAND versus MLP

recognition rates	**# eigenfaces resp. hidden units**		
test set (%)	**5**	**10**	**16**
PENTLAND	78.3 (97.0)	80.2 (91.0)	81.3 (89.5)
MLP	94.0 (99.3)	93.2 (97.7)	95.6 (98.3)

Tabelle 11: Vergleich der Testset-Erkennungsraten von Pentland-Software und MLP bei einer Bildgröße von 32x30. Trainingsset = Datenbasis aus 9 Gesichtern, Testset = Datenbasis aus restl. 7 Gesichtern (in Klammern stehen die Meßwerte der gemischten Datenbasis aus 16 Gesichtern).

Beschreibung des „Faceness"-Moduls

Aufgrund der Meßergebnisse ist meine Entscheidung für den Ansatz zur Bestimmung der „Faceness" auf ein MLP-Netzwerk gefallen.
Dieses wird aus drei Schichten aufgebaut. Zusätzliche MLP-Auswertungen mit direkt am PC laufender Bildverarbeitung ergaben, daß ein Input von 16x15 (also 240 Input-Units) und acht bis zwölf Hidden-Units am besten für die Aufgabe geeignet sind. Der Output-Layer besteht aus zwei Units, die die Eingabebilder nach Gesicht und Nicht-Gesicht klassifizieren (siehe Bild 13 rechts). Über einen Schwellwert Θ_{face} , der die Mindest-Differenz der Output-Werte angibt, kann man innerhalb des Netzpotentials die Klassifikationsgenauigkeit, d.h. die Menge der als Gesicht erkannten Bilder, variieren.
Welche Bildausschnitte einer „Faceness"-Klassifikation unterzogen werden, wird im Face Location-Modul bestimmt. Jene Bildausschnitte, die als Gesicht klassifiziert sind, werden über ein virtuelles Netzlaufwerk an die Workstation übertragen, wo die Identifikation der Gesichter stattfindet.

Face Location

Aufgrund der hohen Anforderungen an die Verarbeitungsgeschwindigkeit wäre es unsinnig in der Face Location einen völlig anderen Ansatz als im „Faceness"-Modul zu implementieren. Der Einsatz von **Template Matching**, **Eigenfaces** oder **Symmetrie-operatoren** ist in diesem Modul daher nicht anzuraten.
Folglich wurde von mir eine **Regelbasierte Methode** implementiert, die die Gesetz-mäßigkeiten menschlicher Körper berücksichtigt. Folgende einfach gehaltene Regeln werden auf die aktiven Bereiche, die die Motion Detection liefert, angewendet :

- Suche nach Gesichtern von oben nach unten
- In einem bestimmten Bereich unterhalb eines Gesichtes (Körper) kann kein anderes Gesicht mehr sein
- Nach Auffinden einer bestimmten Anzahl von Gesichtern wird die Suche in diesem Rahmen abgebrochen (Zeitlimitierung)

Außerdem erfolgt die Abtastung der aktiven Bereiche in zwei (bzw. drei) Pyramiden-Ebenen, um das Finden von Gesichtern in verschiedenen Entfernungen von 1.5m bis 5m (bzw. 10m) zu ermöglichen. Eine 3x3/4-Gauß-Pyramide des Kamerabildes auf dem Matrox-Board ermöglicht es, daß das Abtastfenster je nach Ebene einen unterschiedlich großen Teil des Bildes abdeckt, und trotzdem immer 16x15 Pixel für den Input des „Faceness"-MLP bereitgestellt werden. Die Suche wird mit dem größten Abtastfenster gestartet.

Face Identification

Für die Identifikation auf der Workstation wird der **Eigenface-Ansatz** aus [TP91] verwendet. Die Gründe dafür sind, daß die Software für dieses Verfahren am Institut vorhanden ist, und daß diese Methode laut dem amerikanischen Institut NIST

([WBC94]) die derzeit am meisten ausgetestete (mit einer Datenbasis von tausenden verschiedenen Bildern) am Sektor der Gesichtserkennung ist. Die Gesichter werden durch die Nähe zu einer bestimmten Gesichtsklasse einer bekannten (zuvor erlernten) Person zugeordnet.
Später soll dieses Modul durch eine erweiterte Form des Eigenface-Ansatzes, die Hraby [Hra94] im Zuge ihrer Diplomarbeit entwickelt, ersetzt werden.

Die vorhandene Software wurde nur für das Training nach dem Eigenface-Ansatz genutzt. Einige kleine Erweiterungen waren notwendig, um nach dem Trainingsvorgang die für das Identifikationsmodul notwendigen Werte in Dateien abzuspeichern.
Zur Identifikation von PC-Bildern entwickelte ich ein eigenes Programm auf der Workstation, das die vom PC übertragenen Gesichtsbilder einliest und mit Hilfe der zuvor errechneten Eigenfaces für jedes empfangene Gesicht Γ durch Projektion auf den Face Space einen Mustervektor Ω berechnet. Dieser wird mit den Vektoren Ω_n aller gespeicherten Gesichter verglichen. Wenn die kleinste Distanz ε_{min} zu einem Gesicht Γ_n innerhalb des Face Space kleiner als ein festgelegter Threshold Θ_{dist} ist, so kann das Gesichtsbild eindeutig einem Namen zugeordnet werden (siehe auch Gleichungen 2).
Sowohl die Trainingsbilder Γ_n als auch die zu identifizierenden Gesichter Γ werden mit einer Gaußmaske gewichtet, damit der Einfluß des Hintergrundes minimiert wird. Um mit dieser Maske bei jeder Gesichtsgröße eine optimale Wirkung zu erzielen, muß für jede der drei Skalierungen s=0,1,2 eines Gesichtes eine eigene Gaußmaske, die den Hintergrund fast gänzlich ausblendet, verwendet werden. Bild 18 zeigt für jede Skalierung s die sieben Ansichten eines Gesichtes der Datenbasis nach Anwendung der zugehörigen Gaußmaske. Durch die unterschiediche Anzahl an wertbestimmenden Punkten der Eingabebilder wird es notwendig für jede Skalierung s einen eigenen Face Space zu erzeugen. Die Verwendung mehrerer skalierungsbedingter Face Spaces wird auch bei den „View-Based Eigenspaces“ in [PMS94] praktiziert. Bei der Identifikation des Gesichtes Γ werden die Abstände $\varepsilon(s)$ zu allen drei Face Spaces und die Minimaldistanzen $\varepsilon_{min}(s)$ zu deren Gesichtsklassen unter Verwendung der passenden Maske berechnet. Man erhält so für jede Skalierung s eine Differenz $\Delta dist(s)$ von jeweiliger Minimal-Distanz $\varepsilon_n(s)$ und einem skalierungsabhängigen Schwellwert $\Theta_{dist}(s)$. Vergleicht man die Differenzen $\Delta dist(s)$ miteinander, so kann die Skalierung s ermittelt werden, die das dem geprüften Gesicht Γ am nächsten liegende Gesicht Γ_n enthält.
Ein weiteres Problem stellen die translationsbedingten Performanceeinbußen des Eigenface-Ansatzes dar. Das flexiblere MLP liefert nicht exakt zentrierte Bilder, was zu einer ungenauen Korellation mit den Trainingsbildern führt. Um dieses Verhalten auszugleichen wird von meinem Modul der Rand jedes Gesichtsbildes um jeweils zwei „Overscan-Linien“, die eine Kopie der Ausgangsrandpunkte darstellen, erweitert. In diesem vergrößerten Bild kann nun für jede Skalierung eine Minimumsuche nach $\varepsilon(s)$ durchgeführt werden. Vom jeweils gesichtsähnlichsten Ausschnitt werden dann die minimalen Abstände $\varepsilon_{min}(s)$ zu den Gesichtsklassen bestimmt.

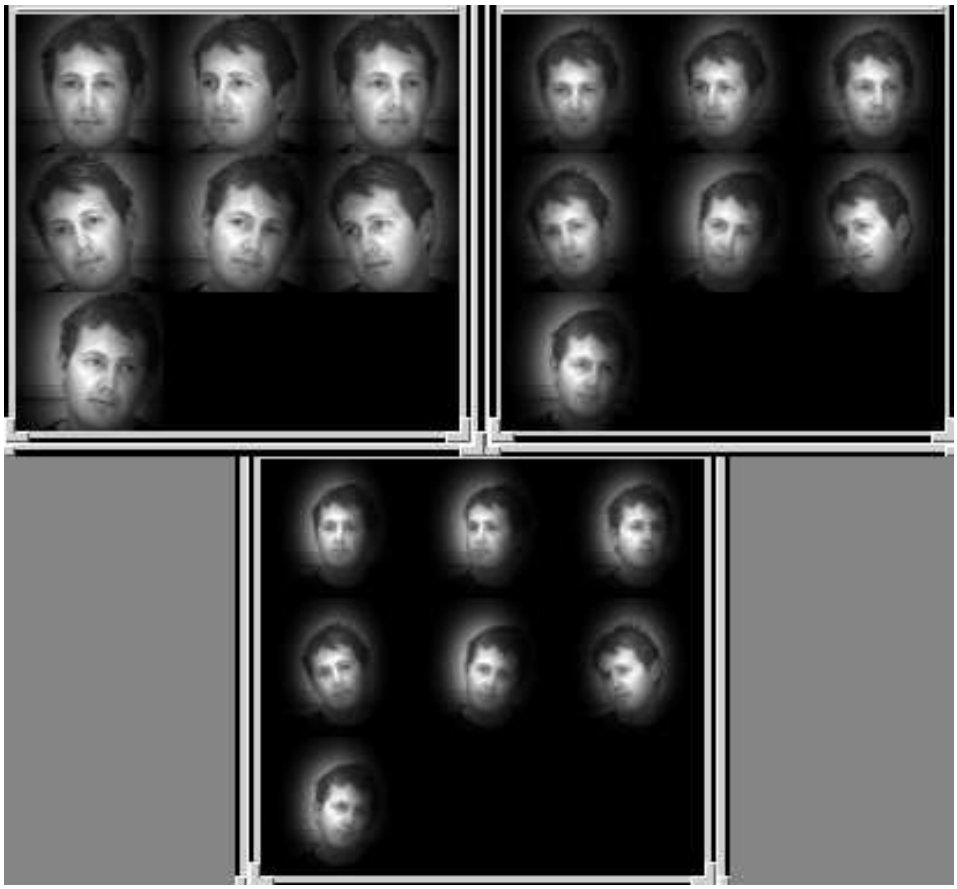

Bild 18: ***Jede der drei Skalierungen s=0,1,2 eines Gesichtes aus der neuen Datenbasis müssen mit einer eigenen Gaußmaske überlagert werden, damit der Hintergrund optimal ausgeblendet wird.***

Robot Control

Dieses Modul ist in dieser Arbeit sehr einfach gehalten und wird hier nur kurz erläutert. Der Roboterarm wird, wenn längere Zeit keine Bewegung detektiert wird, in eine andere Richtung bewegt. Dadurch wird ein anderer Teil des Raumes von der Kamera aufgenommen. Die Anzahl der vorbestimmten Stellungen ist dabei begrenzt.

Schnittstellen zwischen den Modulen

Im folgenden wird der Informationsaustausch, der zwischen den implementierten Modulen stattfindet, kurz erläutert.

Robot Control ↔ Motion Detection

← Keine Bewegung detektiert
→ Rückmeldung: Kamera auf neuer Position

Die Motion Detection veranlaßt nach einer gewissen Zeit ohne Bewegungsdetektion den Roboter die Kamera in eine andere Aufnahmeposition zu bewegen. Nach Beendigung dieser Aktion erhält die Motion Detection eine Rückmeldung.

Motion Detection ↔ Face Location

→ Aktive Bildbereiche
← Rückmeldung: Alle aktiven Bildbereiche geprüft

Wurde von der Motion Detection eine Bewegung detektiert, so werden die entsprechenden aktiven Bereiche an die Face Location übergeben. Dort werden die aktiven Bereiche in einer Schleife regelbasiert (siehe Seite 33) verarbeitet. Danach gibt das Modul die Kontrolle wieder an die Motion Detection zurück.

Face Location ↔ „Faceness“

→ Bildausschnitt mit 16x15 Pixel aus aktiven Bereich
← Rückmeldung: Gesicht oder Nicht-Gesicht

Aus der Face Location wird für jeden aktiven Bereich das „Faceness“-Modul aufgerufen. Dieses verwendet einen Bildausschnitt von 16x15 Pixel Größe, der dem jeweiligen aktiven Bereich entspricht, zur Klassifikation nach Gesicht oder Nicht-Gesicht. Das Ergebnis wird an die Face Location gesendet.

„Faceness“ (PC) → (WS) Face Identification

→ Gesichtsbild mit 32x30 Pixel korrespondierend zu Face Location Bild (16x15)

Wird in der „Faceness“ ein 16x15 Bildauschnitt als Gesicht klassifiziert, so wird das dazugehörige 32x30 Pixel große Gesichtsbild (durch Verwendung der 3x3/4-Gauß-Pyramide, die im Face Location-Modul aufgebaut wurde - siehe Seite 33) über ein Netzlaufwerk an die Workstation gesendet.

6. EVALUIERUNG DES FACE-RECOGNITION-SYSTEMS

In diesem Kapitel werden die Erzeugung der neuen Datenbasis und das anschließende Training des „Faceness"- und des Identification-Moduls beschrieben. Außerdem soll die Performance des Gesamtsystems durch Messungen an den einzelnen Modulen im „Echtzeit"-Betrieb bestimmt werden.
Die Hauptbeleuchtung im Überwachungsraum besteht aus Leuchtstoffröhren an der Decke des Raumes und zwei zusätzlichen Halogen-Scheinwerfern, die sich hinter der Objektivebene der Kamera befinden. Diese Mini-SW-CCD-Kamera (interlaced) ist dabei schon fix auf dem Roboterarm montiert. Die Personen, die im Live-Betrieb des Gesichtserkennungssystems den Überwachungsraum betreten, sollen in einem Bereich, der zwischen 1.5 und 5 Meter von der Kamera entfernt ist, erfaßt werden (siehe Bild 19).

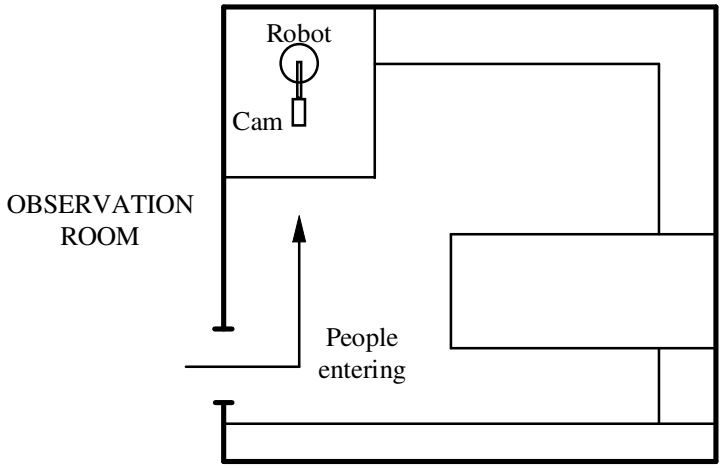

Bild 19: Anordnung des Face-Recognition-Systems im Überwachungsraum.

Aufnahme der neuen Datenbasis

Den ersten Teil der neuen Datenbasis bilden Aufnahmen von Angehörigen des Instituts. Mit Hilfe eines eigens für diesen Zweck kreierten Aufnahmemoduls kann über die Tastatur der Roboter so gesteuert werden, daß das Gesichtsbild vertikal (in Augenhöhe) und horizontal (zwischen den Augen) über ein am Bildschirm sichtbares Fadenkreuz zentriert werden kann. In drei Abständen (1.5, 2 und 2.5 Meter) von der Kamera sind fixe Bodenmarkierungen, auf denen die Personen während des Aufnahmevorganges stehen, angebracht, um die Gesichtsgrößen zwischen jeweils zwei Pyramidenebenen abzudecken (multiscaling, siehe Bild 20). An jeder Position werden sieben Ansichten der Person aufgenommen (frontal und seitlich gedreht,

gekippt, gedreht & gekippt jeweils rechts und links - siehe auch Bild 21). Insgesamt wurden 441 Aufnahmen der Größe 256x240 von 18 verschiedenen Personen gemacht.

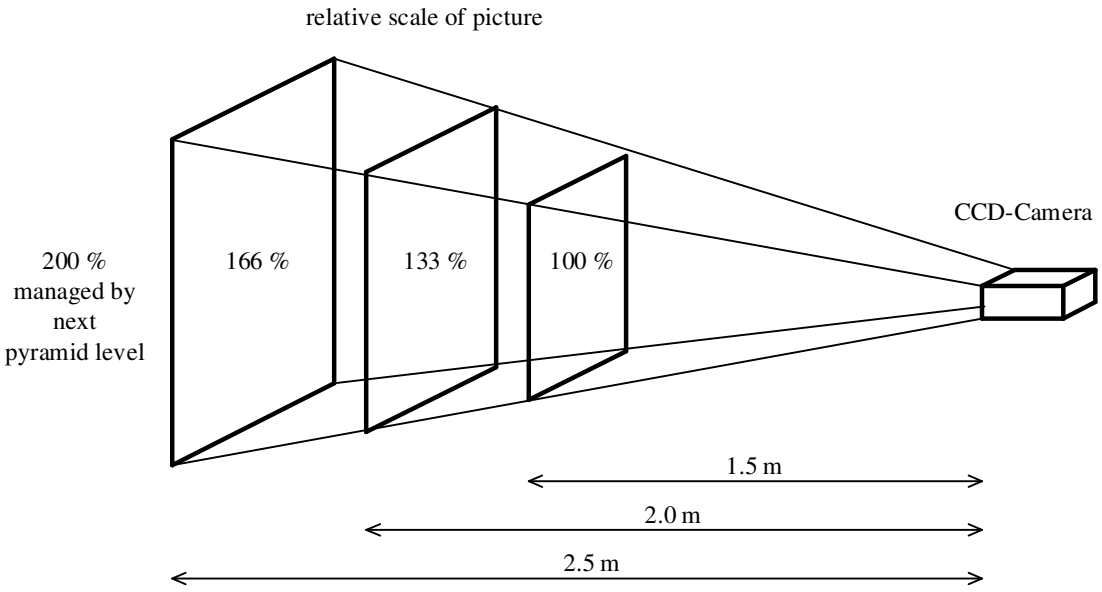

Bild 20: ***Aufnahmegeometrie bei der Erstellung von Gesichtsbildern für die neue Datenbasis. Die relativen Abmessungen der Bildbereiche sind in % angegeben. Bildbereiche, die mehr als doppelt so groß wie der Bezugsbereich (100 %) sind, werden in der nächstgrößeren Pyramidenebene (mit kleinerem Abtastfenster) verarbeitet.***

Zusätzlich zu den vorhandenen Nicht-Gesichtern aus den Vergleichstests (siehe Seite 24 ff) werden 120 weitere, die aus verschiedenen Szenen außerhalb des Raumes stammen, aufgenommen. Durch das Ausschneiden und Vergrößern von vier Teilbildern aus jeder Aufnahme (siehe Bild 22) stehen für die Datenbasis in Summe 728 Nicht-Gesichter zur Verfügung. Infolge des höheren Anteils von Nicht-Gesichtern (als in den Vergleichstests) ist die Streuung der Bildstatistiken größer, und die Klassifikationsgenauigkeit des MLP wird dadurch verbessert.

Training der Module „Faceness“ & Identification

Die generierten Bilder der neuen Datenbasis wurden unter Verwendung einer 3x3/4-Gauß-Pyramide für den Trainingsvorgang auf eine Auflösung von 32x30 (Identification) bzw. 16x15 („Faceness“) reduziert (siehe Bild 23).
Das „Faceness“-Modul wurde mit Hilfe des Xerion-Netzwerk-Simulators trainiert. Als Trainingsset dienten die reduzierten Gesichter und Nicht-Gesichter der neuen Datenbasis. Das dreischichtige Netzwerk besteht aus 16x15 Input-Units, acht Hidden-Units und zwei Output-Units. Nach einer Trainingsdauer des Netzwerkes von 20

Minuten betrug die Erkennungsrate des Trainingssets 99.3 % (wobei alle 441 Gesichter richtig klassifiziert wurden).

Bild 21: Alle Ansichten und Skalierungen zweier Gesichter aus der erzeugten Datenbasis.

Mit der von mir adaptierten Pentland-Software konnte das Identifizieren von Gesichtern trainiert werden. Als Trainingsset dienten ausschließlich die Gesichtsbilder der neuen Datenbasis. Der Input-Vektor besteht aus 32x30 Intensitätswerten. Es werden jeweils die ersten 50 Eigenfaces für jede Skalierung von Gesichtsbildern bestimmt. Damit ist eine sehr gute Repräsentation der Datenbasis gegeben. Die RMS-Pixel-Fehler bei der Rekonstruktion der Gesichter aus den ersten 50 Eigenfaces betragen durchschnittlich über alle drei Skalierungen nur 3.5 % (Bild 24 und Bild 25, siehe auch [SK87]).
Nach Abschluß der Berechnungen erzeugt das adaptierte Pentland-Programm für jede Skalierung drei Dateien, die die Eigenfaces, das Mittelwertbild und die Projektionsvektoren jedes Gesichtes enthalten. Diese neun Dateien werden von meinem entwickelten Identification-Modul dazu benötigt seine eigene lineare Netzstruktur zu erzeugen.

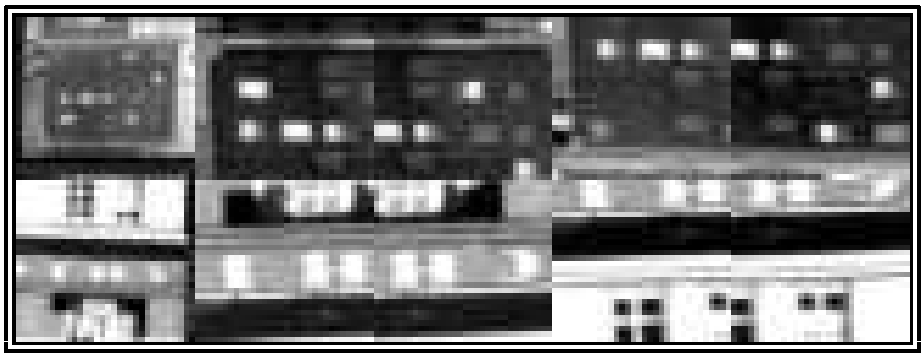

Bild 22: ***Zwei Beispiele für Nicht-Gesichtsbilder, die außerhalb des Überwachungsraumes aufgenommen wurden. Aus jedem Original (links) werden vier Teilauschnitte vergrößert.***

Bild 23: ***Anwendung einer 3x3/4-Gauß-Pyramide auf ein Gesicht aus der neuen Datenbasis. Das aufgenommene Bild wird von 256x240 auf 16x15 Pixel Größe reduziert.***

Performance der einzelnen Module und des Gesamtsystems

Erkennungsraten der einzelnen Module

Es werden Messungen an den Modulen Motion Detection, „Faceness“ und Identification durchgeführt. Um die Performance der einzelnen Module messen zu können, muß sichergestellt sein, daß die davor liegenden Module keine False Positives erzeugen. Diese Voraussetzung ist im Systembetrieb zwar nicht realisierbar, jedoch

kann man sie dadurch simulieren, indem Falschantworten manuell aussortiert werden, bevor das nächste Modul die Daten weiterverarbeitet.

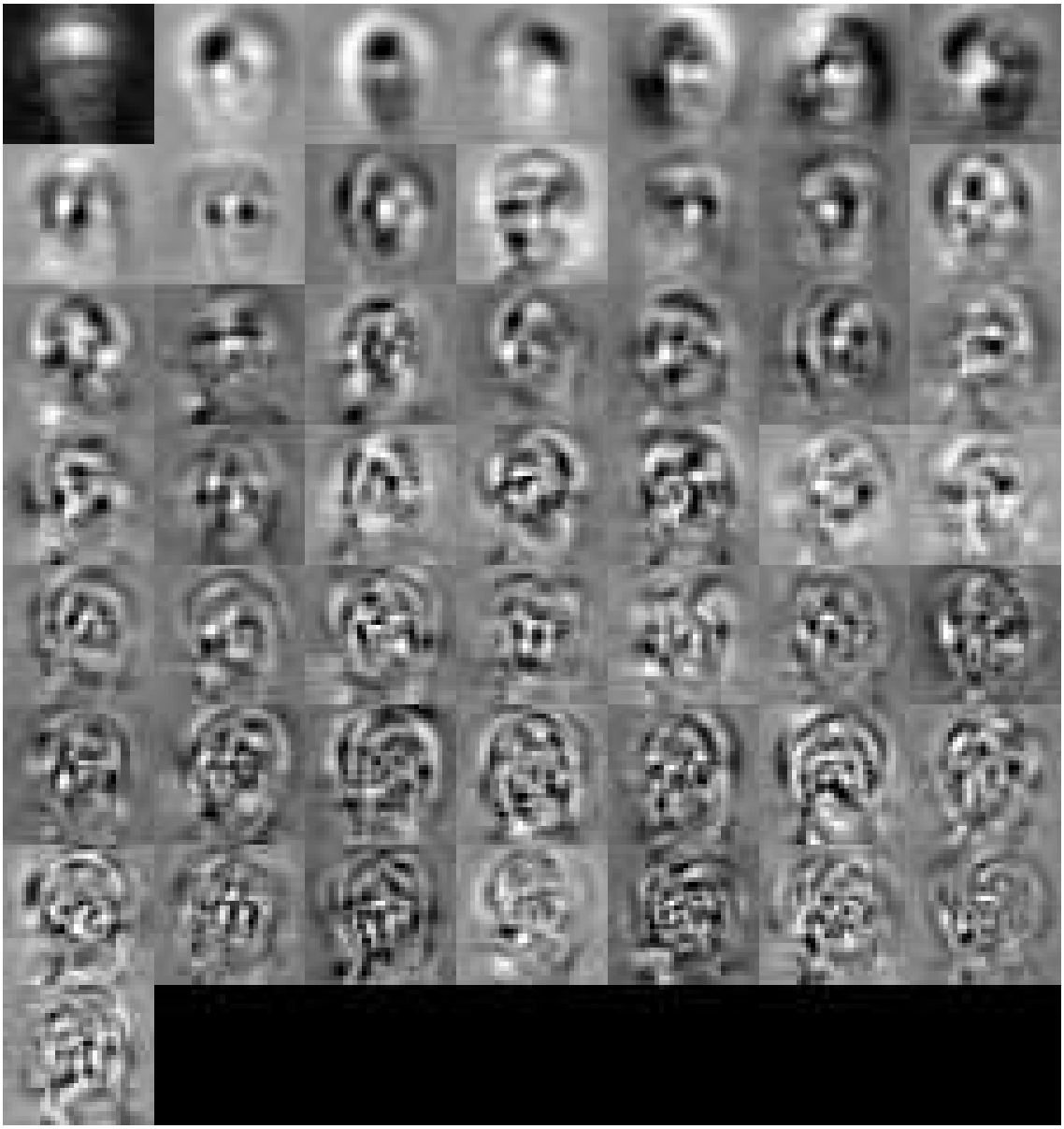

Bild 24: ***Die ersten 50 Eigenfaces nach dem Training des Identification-Moduls mit (am größten skalierten) Gesichtern der neuen Datenbasis. Man erkennt deutlich, daß die Eigenfaces der unteren Reihen weniger Information enthalten.***

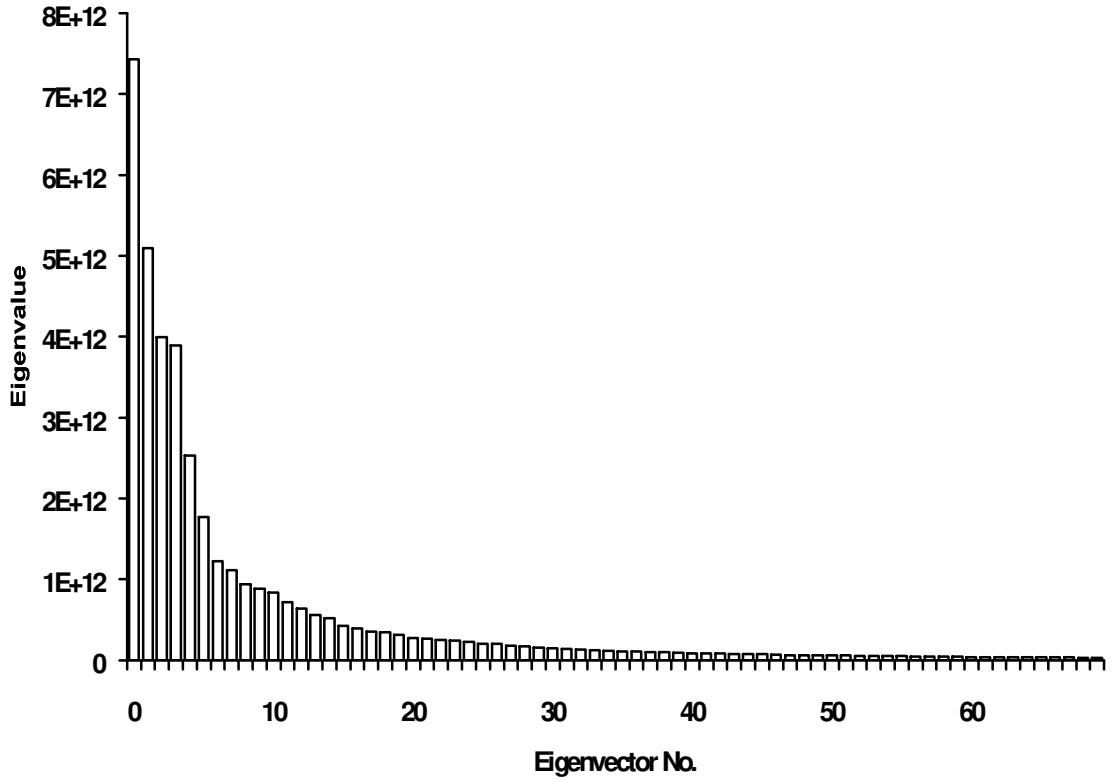

Bild 25: ***Die ersten 70 Eigenwerte der Eigenfaces aus den Gesichtern der größten Skalierung (Bild 24). Durch Verwendung der ersten 50 Eigenfaces wird bei der Rekonstruktion der Bilder nur ein RMS-Pixel-Fehler von 4.4 % gemacht (bei den beiden kleineren Skalierungen sogar nur 3.4 % bzw. 2.7 %). Durch Hinzunahme von z.B. der nächsten 20 Eigenfaces könnte nur eine Verbesserung des Fehlers um 2.2 % (bei größter Skalierung) erreicht werden.***

Motion Detection

Während des Ablaufes der Gesichtserkennung betreten verschiedene Testpersonen aus der Datenbasis einzeln den Überwachungsraum und gehen auf die Kamera zu. Die von der Kamera erfaßten Frames werden von der Motion Detection auf aktive Bereiche geprüft. Dieses Modul ist so konzipiert, daß es solange Frames in Echtzeit (15 Bilder/s) aufnimmt, bis eine Bewegung detektiert wird. Durch die Maximum-Pyramide wird das Differenzbild bis zu einer Auflösung von 16x16 reduziert, wobei einzelne False Negatives des Ausgangsbildes von Nachbardetektionen soweit ausgeglichen werden, daß im reduzierten Bild keine False Negatives mehr meßbar sind. Für die Messung sind daher nur die False Positives bei einer bestimmten Anzahl von richtig erkannten aktiven Bereichen ausschlaggebend. Tabelle 12 zeigt die Anzahl der gemessenen Frames, sowie eine Gegenüberstellung der False Positives mit der Anzahl der „echten“ aktiven Bereiche.

Face Location

Danach werden die False Positives manuell ausgeschieden (in diesem Fall keine), sodaß das Face Location-Modul von der Motion Detecton nur richtig erkannte aktive Bereiche erhält. Beim Lokalisieren der Gesichter werden durch einfache Regeln bestimmte aktive Bereiche für die Weiterleitung an das „Faceness“-Modul ausgewählt. Da bei der Selektion keine False Positives entstehen können und die Regeln so geartet sind, daß auch die False Negatives sehr niedrig sind (< 0.001 %), kann in diesem Modul auf Messungen verzichtet werden. Mit diesen einfachen Regeln können immerhin 10-15 % der aktiven Bereiche ausgeschieden werden.

„Faceness“

Die von „Faceness“-Modul als Gesicht klassifizierten Bilder setzen sich aus richtig erkannten Gesichtern und False Positives zusammen, die auf das Netzlaufwerk übertragen werden. Zusätzlich wird noch die False Negative Rate ermittelt. In Tabelle 13 sind die Anzahlen aller geprüften Bereiche und der darin inkludierten Gesichter, sowie die verschiedenen Erkennungsraten dargestellt. Auffallend ist die große Anzahl der False Positives, die sich aus der Mehrfach-Detektion eines Gesichtes in einem Frame ergibt (siehe Bild 26 und Bild 27). Durch die hohe Flexibilität des MLP in Bezug auf Translationen (siehe Tabelle 10) werden auch nicht zentrierte Gesichtsbilder, die zwar das Gesicht beinhalten, aber trotzdem als False Positives gezählt werden (das Identification-Modul braucht zentrierte Bilder), vom „Faceness“-Modul als Gesicht klassifiziert.

Motion Detection (PC-Matrox)	
# frames	100
# true active areas	2756
# False Positives	0

Tabelle 12: Erkennungswerte am Motion Detection-Modul (am Matrox-Board des PC implementiert).

„Faceness“ (PC)	
# active areas	2756
# detected faces	406
# true faces	65
recognition rate faces	16 %
# False Positives	341
False Positive rate	84 %
# False Negatives	27
False Negative rate	1.1 %

Tabelle 13: Erkennungsraten am Modul „Faceness“ (am PC implementiert). Von der Motion Detection falsch erkannte aktive Bereiche wurden vorher aussortiert.

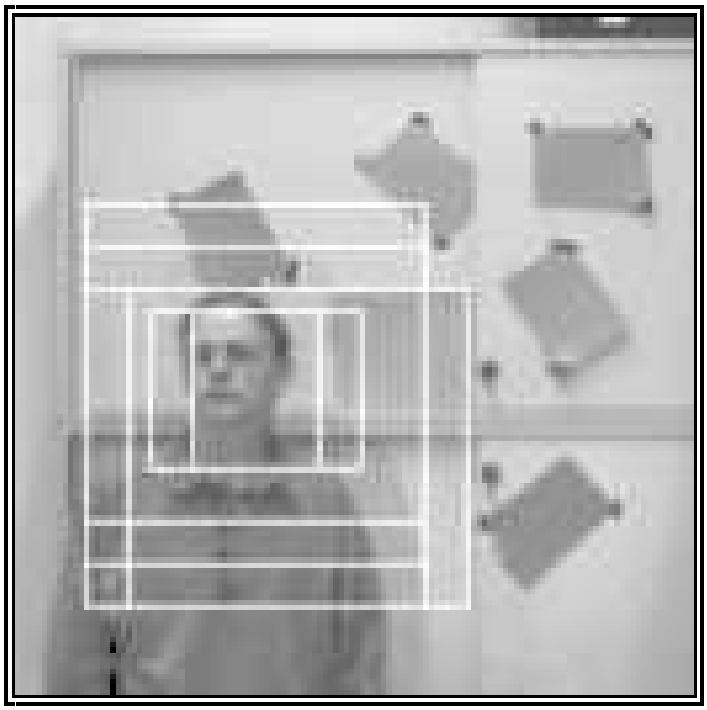

Bild 26: ***Erstes Beispiel für Mehrfach-Detektionen des „Faceness“-Moduls.***

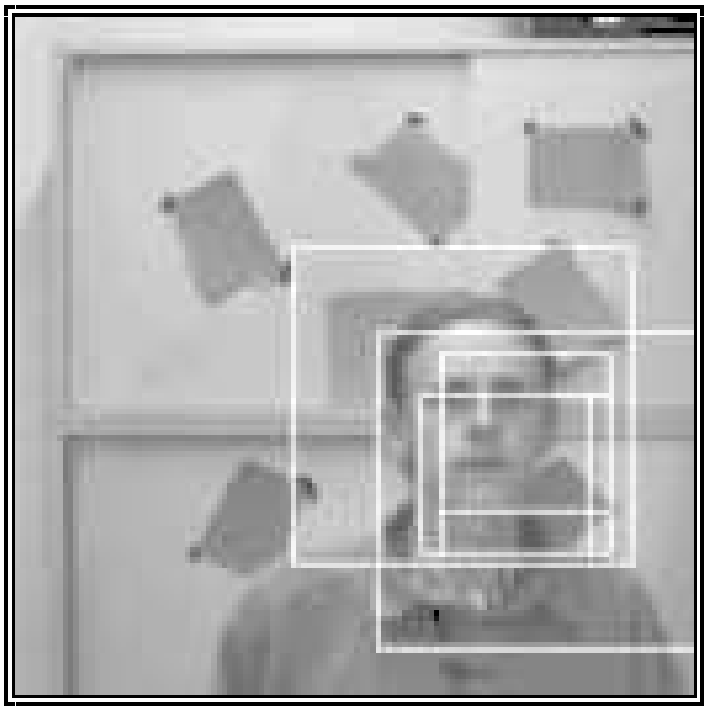

Bild 27: ***Zweites Beispiel für Mehrfach-Detektionen des „Faceness“-Moduls.***

Identification

Bevor die Face Identification gestartet wird, werden die Falsch-Gesichter händisch vom Netzlaufwerk entfernt. Mit den übrigen Gesichtern kann das Identification-Modul ausgewertet werden. Tabelle 14 zeigt die Anzahl der zu identifizierenden Gesichter, sowie die Werte der dazugehörigen Erkennungsraten. Die relativ schlechte Verhältnis zwischen richtig und falsch identifizierten Gesichtern (18/9) kommt wahrscheinlich durch die manuelle Aussortierung (es ist schwierig, exakt zentrierte Bilder händisch zu bestimmen) zustande. Im Falle des Identification-Moduls ist daher die gemessene Erkennungsrate beim Test des Gesamtsystems (siehe nächster Abschnitt), die ohne manuellen Eingriff zustande kommt, aussagekräftiger.

Identification (WS)	
# true faces	65
# identifications	27
# true identified faces	18
# wrong identified faces	9
# identification rate	66.7%

Tabelle 14: Darstellung der Erkennungsraten am Identification-Modul, das auf der Workstation ausgeführt wird. Es ist sichergestellt, daß dieses Modul nur „echte" Gesichter zur Verarbeitung erhält.

Erkennungsrate und Geschwindigkeit des Gesamtsystems

Das Gesichtserkennungssystem wird ohne manuellen Eingriff in Betrieb genommen. Wieder gehen verschiedene Personen einzeln auf die Kamera zu. Gemessen wird die Anzahl der aufgenommenen Frames (jeder Frame enthält nur eine Aufnahme eines Gesichtes) und die resultierende Erkennungsrate. Die Werte sind in Tabelle 15 aufgelistet. Außerdem werden noch Laufzeitmessungen am PC und an der Workstation durchgeführt, die in Tabelle 16 dargestellt sind.
Aus den Messungen folgt, daß , wenn sich eine Person ca. 10 Sekunden im Überwachungsraum aufhält, die identifizierten Gesichter mit einer Wahrscheinlichkeit von 82.4 % richtig sind (und in den restlichen 17,6 % der Fälle ein falscher Name geliefert wird). Durch Ausdehnung der Aufenthaltszeit der Person im Überwachungsbereich oder durch Steigerung der Framerate (derzeit ca. 1 Frame/s) kann die Anzahl der Identifikationen (richtige und falsche) erhöht werden. Werden dann etwa 3 aufeinanderfolgende identifizierte Gesichter betrachtet, so kann man mit einer Mehrheitsentscheidung (2 aus 3) die Identifikationsrate auf 91.8 % erhöhen.
Es fällt auf, daß die Workstation zur Verarbeitung der Gesichter aus den 100 Frames ungefähr doppelt so lange braucht wie der PC. Die Dateien sind mit einer laufenden Nummer (die sich wiederholt) kodiert, sodaß bei längerem Aufenthalt einer Person im Überwachungsraum die ältesten Bilder wieder überschrieben werden und so kein

Überlauf auf dem Netzlaufwerk entstehen kann. Dieses Problem tritt aber nur deshalb auf, weil derzeit sehr viele Gesichter vom „Faceness“-Modul detektiert werden. Hier sind sicher einige Verbesserungen möglich, die ich im nächsten Abschnitt besprechen möchte.

Face Recognition System - Recognition Performance	
#frames = # different face sites	100
# identifications	17
# true identified faces	14
# wrong identified faces	3
identification rate per frame	82.4 %

Tabelle 15: Erkennungsrate des Gesichtserkennungssystems im Live-Betrieb.

Face Recognition System - Time Measures	
# frames	100
Time - PC(+Matrox) (s)	120
Time / frame (s)	1.2
# received faces	367
Time - Workstation (s)	242
Time / received face (s)	0.66

Tabelle 16: Laufzeitmessungen am Gesichtserkennungssystem im Live-Betrieb.

Überlegungen zur Steigerung der Performance

Das Gesichtserkennungssystem kann in zweierlei Hinsicht optimiert werden. Auf der einen Seite kann man versuchen die Laufzeiten zu verkürzen (dadurch steigt die Framerate), indem schnellere Verfahren oder bessere Hardware eingesetzt wird und andererseits können Methoden, die zu niedrige Erkennungsraten aufweisen durch andere ausgetauscht werden. Im Folgenden werde ich das System modulweise auf seine Schwächen durchsuchen und einige Lösungsansätze zur Optimierung des Systems aufzeigen.

Motion Detection

Im Bild 28 sind Laufzeitmessungen an den PC-Modulen dargestellt. Wie aus der Auflistung ersichtlich ist, sind die Berechnungen des MLP (_propagate) mit 43 % und das Senden der erkannten Gesichter (_sendFace) mit 21 % an der Gesamtlaufzeit am aufwendigsten. Der Zeitverbrauch des MLP entsteht hauptsächlich durch die hohe

Anzahl von Anfragen an das „Faceness“-Modul (7192 aktive Bereiche). In diesem Fall kann durch eine intelligentere Bewegungserkennung die Menge der detektierten Bereiche stark reduziert werden und somit auch die Anzahl der Aufrufe des MLP.
Die Bewegungserkennung könnte zum Beispiel mit einem **Motion Tracking-Algorithmus** gekoppelt werden, der von der Position eines erkannten Gesichtes zum Zeitpunkt t-1 aus versucht, die Position des gleichen Gesichtes zum Zeitpunkt t zu bestimmen. Die Stelle, an der das Gesicht zum Zeitpunkt t-1 ist, kann entweder mit der herkömmlichen Methode unter Verwendung der Motion Detection & „Faceness“ solange, bis ein Gesicht gefunden wurde (Rückmeldung durch „Faceness“-Modul), bewerkstelligt werden, oder man sucht im Bild nach charakteristischen Blob-konstellationen (Augen und Nase), womit zugleich die ungefähre Größe des Gesichtes bekannt wäre.
Wenn der Tracking-Algorithmus mit dem Roboter gekoppelt wird, hätte dies den weiteren Vorteil, daß sich ein Gesicht länger im Beobachtungsbereich befindet (daraus folgt eine höhere Framerate), uns so die Erkennungswahrscheinlichkeit steigt.

„Faceness“

Der hohe Anteil der Face-Location-Funktion (_FaceLocation) von 21 % an der Gesamtzeit (Bild 28) rührt daher, weil hier das aufgenommene Bild vom Pufferspeicher des Matrox-Boards in den PC-Speicher kopiert wird, damit die Input-Units des MLP darauf zugreifen können. Eigentlich müßte ein Großteil des Zeitaufwandes dieser Funktion zum MLP hinzugerechnet werden, sodaß es über die Hälfte der Gesamt-laufzeit verbraucht.
Eine Möglichkeit dieses Problem zu bekämpfen wäre die Verwendung eines schnelleren PC’s (derzeit 486DX2/66). Die Installation eines Pentium 100 ist bereits in Vorbereitung. Jedoch kann höchstens mit einer Verbesserung um den Faktor 2 gerechnet werden.
Ein besserer Vorschlag ist die Implementierung des MLP auf dem Matrox-Board, da das Vorwärtspropagieren der Inputs durch das Netz nichts anderes als Matrizen-multiplikationen sind, die das Board mit mindestens gleicher oder höherer Geschwindigkeit (15-30 Mhz pixel access rate) als der PC ausführen kann. Der Hauptvorteil würde darin liegen, daß die viel niedrigere Anzahl von erkannten Gesichtern (im Gegensatz zu den aktiven Bereichen) direkt vom Puffer des Matrox-Boards auf das Netzlaufwerk übertragen werden könnten.
Trotzdem sollte, wie schon bei der Motion Detection erwähnt, das vorrangige Ziel sein, die Anzahl der vom MLP zu untersuchenden Bereiche zu reduzieren, wobei sicher ein Geschwindigkeitsvorteil des „Faceness“-Moduls um den Faktor 10 erreicht werden kann.

Betrachtet man die Erkennungsraten des MLP (Tabelle 13), so sticht die False Positive Rate mit einem Wert von 84 % heraus. Durch die hohe Fehlerrate werden viele Bilder unnötigerweise auf das Netzlaufwerk übertragen, wodurch ein Aufwand von 21 % (_sendFace) an der Gesamtlaufzeit entsteht (Bild 28).
Man könnte Versuchen die False Positives des MLP mit einer höheren Anzahl von Hidden-Units oder einem speziellen Netzaufbau zu senken, jedoch führen beide Möglichkeiten zu einer Erhöhung der Laufzeiten.

Eine andere Möglichkeit bietet (unter Zuhilfenahme des Matrox-Boards) das Ausblenden des Hintergrundes mit einer Maske (wie im Identification-Modul), wobei aber die Größe des jeweiligen Gesichtes berücksichtigt werden muß (kann z.B. durch Überlagerung des Motion Energy Detection-Bildes oder durch einen Motion Tracker ermittelt werden).
Empfehlenswert wäre es auch , beim Training des MLP einen Bootstrapping-Algorithmus zu verwenden ([RBK95]), der falsch erkannte Gesichter zu den Nicht-Gesichtern des Trainingssets hinzufügt.

Identification

In Bezug auf den Zeitaufwand ist dieses Modul, wie in Tabelle 16 zu sehen ist, mit 0.66 Sekunden pro Gesichtsidentifikation als zufriedenstellend zu bewerten. Das hohe Meßergebnis der Gesamtlaufzeit auf der Workstation kommt nur durch die hohe False Positive Rate des „Faceness“-Moduls zustande.

Was auf jeden Fall verbessert werden muß, ist die Erkennungsrate der Identifikation, denn derzeit sind noch ein Sechstel der identifizierten Gesichter Falschantworten.
Die verwendete Methode der Multiscaled Eigenspaces könnte auf „echte“ View-Based Eigenspaces ([PMS94]) erweitert werden, indem für jede Skalierung und Ansicht eines Gesichtes ein eigener Eigenspace (in meinem Fall 21) erzeugt wird.
Auch der auf Eigenfaces basierende Ansatz von Sung und Poggio ([SP94]) , bei dem sie auch Nicht-Gesichter in den Trainingsvorgang miteinbringen, kann zu besseren Erkennungsraten führen. Sie verwenden dabei sogenannte „View-Based Face and Non-Face prototype clusters“.
Weiters wird die Diplomarbeit von Hraby [Hra94] zeigen, ob ihr erweiterter Eigenface-Ansatz Verbesserungen für das System bringt.

```
Turbo Profiler  Version 2.1  Mon Sep 11 17:27:43 1995
Program: C:\USR\MATROX\NEMEC\BIN\FACEREC.EXE

Execution Profile
Total time: 126.09 sec
% of total: 14 %
       Run: 5 of 5

    Filter: All
      Show: Time
      Sort: Frequency

_propagate       55.444 sec  43% |*********************************************
_sendFace        26.583 sec  21% |*********************
_FaceLocation    21.310 sec  16% |*****************
_MotionDetection 18.564 sec  14% |***************
_readWeights     2.3955 sec   1% |*
_main            1.7458 sec   1% |*
_net_quest       0.0121 sec  <1% |
_my_exit         0.0091 sec  <1% |
_InitMatrox      0.0043 sec  <1% |
_Surf2FB         0.0004 sec  <1% |
_net_init        0.0000 sec  <1% |


    Filter: All
      Show: Counts
      Sort: Frequency

_propagate             7192  47% |++++++++++++++++++++++++++++++++++++++++++++++++
_net_quest             7192  47% |++++++++++++++++++++++++++++++++++++++++++++++++
_Surf2FB                500   3% |+++
_sendFace               169   1% |+
_MotionDetection        100  <1% |
_FaceLocation           100  <1% |
_readWeights              5  <1% |
_net_init                 5  <1% |
_my_exit                  5  <1% |
_InitMatrox               5  <1% |
_main                     5  <1% |


    Filter: All
      Show: Time per call
      Sort: Frequency

_readWeights     0.4791 sec/call |*********************************************
_main            0.3491 sec/call |*********************************
_FaceLocation    0.2131 sec/call |********************
_MotionDetection 0.1856 sec/call |******************
_sendFace        0.1573 sec/call |***************
_propagate       0.0077 sec/call |
_my_exit         0.0018 sec/call |
_InitMatrox      0.0008 sec/call |
_net_init        0.0000 sec/call |
_net_quest       0.0000 sec/call |
_Surf2FB         0.0000 sec/call |
```

Bild 28: ***Laufzeitmessungen der Module am PC. Für insgesamt 100 Frames sind für jedes Modul die Gesamtzeit (oben), die Anzahl der Aufrufe (mitte) und die Zeit pro Aufruf (unten) dargestellt. Den größten Zeitbedarf haben das MLP (_propagate) mit 43 % und die Übertragung der erkannten Gesichter auf das Netzwerk (_sendFace) mit 21 %.***

7. ZUSAMMENFASSUNG

Das in dieser Diplomarbeit entwickelte Gesichtserkennungssystem zeichnet sich besonders durch seinen modularen Aufbau aus. Somit ist es möglich zukünftige Verbesserungen leicht in das System einzubringen. Es können die bestehenden Module optimiert oder einfach als ganzes durch neue Module, die andere Verfahren verwenden, ersetzt werden.
Weiters wäre es ohne erheblichen Aufwand möglich, dieses System auch für andere Objekte als Gesichter zu verwenden. Es müßte nur das MLP und das Identification Modul mit einer Objektdatenbasis retrainiert werden.

In der jetzigen Konfiguration meines Gesichtserkennungs-Systems verwendet das erste Modul, das die Bilder direkt von einer CCD-Kamera erhält, für die Registrierung von bewegten Objekten den Motion Energy Detection-Algorithmus (mit Bildverarbeitungs-Hardware implementiert). Im nächsten Modul, der Face Location, werden einfache Regeln verwendet, um die Anzahl der aktiven Bereiche zu verringern, die dem „Faceness"-Modul übergeben werden. Hier werden die ausgewählten Bildbereiche einem neuralen Netz präsentiert, das eine Klassifikation nach Gesicht und Nicht-Gesicht durchführt. Bis hierher werden alle Aufgaben auf dem PC ausgeführt. Die vom PC auf die Workstation übertragenen Gesichts-Bilder werden dort mit Hilfe eines eigens entwickelten Programmes, das auf dem Eigenface-Ansatz basiert, identifiziert.

Um die Module „Faceness" und Identification trainieren und testen zu können, wurde eine große Datenbasis, die aus 441 Gesichtern von 18 Institutsangehörigen und 728 Nicht-Gesichtern besteht, aufgenommen.
Im praktischen Betrieb werden Personen des Institutes, die sich im Überwachungsraum befinden, vom Gesichtserkennungs-System identifiziert. Wenn sich die zu identifizierende Person ca. 10 s im Überwachungsraum aufhält, sind 82.4 % der identifizierten Gesichter richtig (bei den restlichen 17.6 % wird ein falscher Name ausgegeben). Die Identifikationsrate ist für diesen Prototypen eines vollautomatischen Gesichtserkennungssystems als relativ gut zu bewerten. Mit den vorgeschlagenen Verbesserungen und einer optimalen Ausnutzung der vorhandenen Hardware ist mit diesem System vor allem im Geschwindigkeitsbereich, aber auch bei der Erkennungsrate, noch eine Leistungssteigerung möglich.

LITERATURVERZEICHNIS

[Bar81] R.J. Baron, „Mechanisms of Human Facial Recognition“, p.137-178, in *International Journal of Man- Machine Studies*, No.15, 1981

[BP93] Roberto Brunelli, Tomaso Poggio, „Face Recognition: Features versus Templates“, p.1042-1052, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.15, No.10, October 1993

[Buh86] R. Buhr, „Analyse und Klassifikation von Gesichtsbildern“, S.245-256, in *ntzArchiv*, Nr.8, 1986

[Bur88] Peter J. Burt, „Attention Mechanisms for Vision in a Dynamic World“, p.977-987, in *IEEE Proceedings 9th International Conference on Pattern Recognition*, Vol.II, Rome, November 1988

[CF90] G.W. Cottrell, M.K. Fleming, „Categorization of Faces using Unsupervised Feature Extraction“, p.65-70, in *Proceedings International Journal Conference on Neural Networks*, Vol.II, San Diego, 1990

[CL93] Dmitry Chetverikov, Attila Lerch, „Multiresolution Face Detection“, in: R.Klette, W.G. Kropatsch, Eds., „*Theoretical Foundations of Computer Vision*“, Series: Mathematical Research, Vol.69, Berlin 1993

[DC88] J.H. Duncan, T. Chou, „Temporal edges: The detection of motion and the computation of optical flow“, p.374-382, in *Proc. Second Int. Conf. Comput. Vision*, Tampa, FL, Dec. 1988

[GHL71] A. Goldstein, L. Harmon, B. Lesk, „Identification of Human Faces“, p.748-760, in *Proceedings of the IEEE*, 59, May 1971

[GSS90] V. Govindaraju, S.N. Srihari, D.B. Sher, „A Computational Model for Face Location“, p.718-721, in *Proceedings 3rd International Conference on Computer Vision*, 1990

[Hra94] Karin Hraby, „*Identifizieren von Gesichtern durch Steuerung der visuellen Aufmerksamkeit*“, Aufgabenstellung der Diplomarbeit in der Abt. für Mustererkennung und Bildverarbeitung, Institut für Automation, PRIP-TR-30, TU Wien, März 1994

[HWA93] T.S. Huang, J.J. Weng, N. Ahuja, „Learning Recognition and Segmentation of 3D Objects from 2D Images“, p.121-128, in *Proc. IEEE Int. Conf. on Computer Vision*, 1993

[Köh29] W. Köhler, „*Gestalt Psychology*“, Liveright, New York, 1929

[Koh88] T. Kohonen, „*Self-Organization and Associative Memory*“, Springer Verlag, Berlin, 1988

[Kro91] Walter G. Kropatsch, „*Digitales Sehen mit Bildpyramiden*“, Technischer Report, Abt. für Mustererkennung und Bildverarbeitung, Institut für Automation, PRIP-TR-1, TU Wien, März 1991

[KS90] M. Kirby, L. Sirovich, „Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces“, p.103-108, in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.12, No.1, 1990

[LMN92] R. Linggard, D:J. Myers, C. Nightingale, „*Neural Networks for Vision, Speech an Natural Language*“, London, 1992

[LVB93] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C.v.d.Malsburg, R. Wurtz, p.300-311, „Distortion Invariant Object Recognition in the Dynamic Link Architecture“, in *IEEE Trans. Computers*, Vol.42, 1993

[MB94] Don Murray, Anup Basu, „Motion Tracking with an Active Camera“, p.449-459, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.16, No.5, May 1994

[MP94] B. Moghaddam, A. Pentland, „Face Recognition using View-Based and Modular Eigenspaces“ (MIT Report No. 301), also in *Automatic Systems for the Identification and Inspection of Humans*, SPIE Vol. 2277, July 1994

[NMM91] O. Nakamura, S. Mathur, T. Minami, „Identifikation of Human Faces Based on Isodensity Maps“, p.263-272, in *Pattern Recognition*, Vol.24, 1991

[PMP86] Perrett, Mistlin, Potter, Smith, Head, Chitty, Broennimann, Milner, Jeeves, „Functional Organization of Visual Neurones processing Face Identity“, p.187-198, in: H.D. Ellis, M.A. Jeeves, F. Newcombe, A. Young, Eds., „*Aspects of Face Processing*“, Dordrecht 1986

[PMS94] A. Pentland, B. Moghaddam, T. Starner, „View-based and Modular Eigenspaces for Face Recognition“ (MIT Report No. 245), also in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 1994

[RBK95] H.A. Rowley, S. Baluja, T. Kanade, „*Human Face Detection in Visual Scenes*“ (CMU-CS-95-158), School of Computer Science, Pittsburgh, July 1995

[RHW86] D.E. Rumelhart, G.E. Hinton, R.J. Williams, „Learning Internal Representations by Error Propagation“, in „*Parallel Distributed Processing: Explorations in the Microstructure of Cognition*“, chapter 8, MIT Press, England, 1986

[Sam91] Ashok Samal, „Minimum Resolution for Human Face Detection and Identification“, in *Proceedings SPIE/SPSE Symposium Electronic Imaging*, January 1991

[Ser86] Justine Sergent, „Microgenesis of Face Perception“, p.17-33, in: H.D. Ellis, M.A. Jeeves, F. Newcombe, A. Young, Eds., „*Aspects of Face Processing*“, Dordrecht 1986

[SI92] Ashok Samal, Prasana A. Iyengar, „Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey“, p.65-77, in *Pattern Recognition*, Vol.25, No.1, 1992

[SK87] L. Sirovich, M. Kirby, „Low-dimensional procedure for the characterization of human faces“, p. 519-524, in *Journal of the Optical Society of America A*, 4(3), 1987

[SP94] K. Sung, T. Poggio, „*Example-based Learning for View-based Human Face Detection*“, MIT - A.I. Memo No. 1521, Dec.1994

[TP91] Matthew Turk, Alex Pentland, „Eigenfaces for Recognition“, p.71-86, in *Journal of Cognitive Neuroscience*, Vol.4, No.1, 1991

[WBC94] C.L.Wilson, C.S.Barnes, R.Chellappa, S.A.Sirohey, „*Face Recognition Technology for Law Enforcement Applications*“, U.S. Dept. of Commerce, National Inst.of Standards and Technology (NIST), NISTIR 5465, July 1994

[YRW92] Y. Yeshurun, D. Reisfeld, H. Wolfson, „Symmetry: a context free Cue for Foveated Vision“, p.477-491, in „*Neural Networks for Perception*“, Vol.1, 1992