



Internal Report 2004-01

## Finden von Punktkorrespondenzen zwischen Gesichtsbildern

*by*

Marco Müller

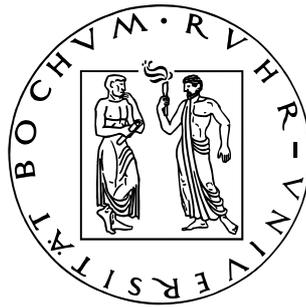
Ruhr-Universität Bochum  
Institut für Neuroinformatik  
44780 Bochum



IR-INI 2004-01  
March 2004  
ISSN 0943-2752



# Finden von Punktkorrespondenzen zwischen Gesichtsbildern



Diplomarbeit  
von  
Marco Müller<sup>1</sup>

Fakultät für Physik und Astronomie  
Ruhr-Universität Bochum

durchgeführt am  
Institut für Neuroinformatik  
Ruhr-Universität Bochum

Februar 2004

---

<sup>1</sup>Email: [email@marcomueller.de](mailto:email@marcomueller.de)

## Zusammenfassung

Ziel dieser Arbeit ist das Finden von Punktkorrespondenzen zwischen Gesichtsbildern. Dazu werden ausgehend von der Methode der Graphen Anpassung verschiedene Verfahren untersucht, mit denen eine genauere Landmarkenpositionierung erreicht werden soll. Diese sind eine Bündelgraphenteilung, eine veränderte Ähnlichkeitsfunktion und das PCA-Matching.

Für die Beurteilung der Positionierung werden eine Vielzahl künstlicher Gesichter mit exakt bekannter Landmarkenposition verwendet. Die Auswirkungen dieser Verfahren auf das Gesichterfinden und die Leistung bei der Gesichtserkennung werden auf künstlichen wie auf natürlichen Bildern untersucht.

Mit dem Verfahren des PCA-Matching wurde eine deutliche Verringerung des Positionierungsfehlers erreicht. Auf künstlichen wie auf natürlichen Bildern führte dies zu einer verbesserten Erkennungsleistung.

## Abstract

Objective of this thesis is finding point correspondences between face images. Starting from the method of graphmatching, several methods are analysed to achieve a more exact landmark positioning. The considered methods are a bunchgraph partition, a modified similarity function and PCA-matching.

For the evaluation of the positioning, lots of computer generated faces with exactly known landmark position are used. The effects of these methods to face finding and recognition are analysed on artificial as well as on natural images.

With the method of PCA-matching a marked reduce of the positioning error has been achieved. On generated, as well as on natural images, this has resulted in an improved recognition performance.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Motivation . . . . .	4
1.2	Problemstellung . . . . .	4
<b>2</b>	<b>Bisherige Arbeiten</b>	<b>6</b>
2.1	Neuronale Netze . . . . .	6
2.2	Gesichterfinden mit neuronalen Netzen . . . . .	7
2.3	Anpassung von Graphen . . . . .	8
2.4	Hauptkomponentenanalyse (PCA) . . . . .	13
2.5	Eigengesichter (Eigenfaces) . . . . .	15
<b>3</b>	<b>Verwendete Daten und Vorverarbeitung</b>	<b>17</b>
3.1	Künstliche Face-Gen Gesichter . . . . .	17
3.2	Variation der Face-Gen Gesichter . . . . .	20
3.3	Natürliche Gesichter der Kodak-Datenbank . . . . .	20
3.4	Natürliche Gesichter . . . . .	21
<b>4</b>	<b>Verfahren zur Bestimmung der Landmarkenpositionen</b>	<b>23</b>
4.1	Teilung des Bündelgraphen . . . . .	23
4.2	Veränderte Ähnlichkeitsfunktion . . . . .	26
4.3	PCA-Matching . . . . .	31
4.4	Skala und Orientierung, Normierung . . . . .	36
<b>5</b>	<b>Gesichterfinden und Erkennen</b>	<b>37</b>
5.1	Leistung des Gesichtsfinders . . . . .	37
5.2	Erkennungsleistung . . . . .	39
<b>6</b>	<b>Diskussion und Ausblick</b>	<b>41</b>
<b>A</b>	<b>Move-Parameter</b>	<b>42</b>
A.1	Parameterdatei 1 . . . . .	42
A.2	Parameterdatei 2 . . . . .	43

# Kapitel 1

## Einleitung

### 1.1 Motivation

Der Mensch besitzt die Fähigkeit, mühelos Gesichter zu sehen und zu erkennen. Dabei ist es erstaunlich, wie sehr die Entfernung der Person, die Orientierung des Gesichtes oder die Lichtverhältnisse variieren können. Diese Veränderungen führen in einem Bild zu einer völlig anderen Helligkeitsverteilung der Pixel. Dennoch ist der Mensch in der Lage, das Gesicht im Bild richtig zu lokalisieren und die Person zu erkennen. Auch Veränderungen des Gesichtes selbst, ein neuer Bart, ein veränderter Gesichtsausdruck, eine Sonnenbrille stellen den Menschen in der Regel vor keine größeren Schwierigkeiten. Er ist fähig, selbst bei sehr unterschiedlich aussehenden Personen korrespondierende Punkte, wie zum Beispiel die Nasenspitze oder die Pupillen, zu finden und diese zu vergleichen.

Computersysteme hingegen sind bisher nur sehr eingeschränkt in der Lage, Erkennungsaufgaben zu übernehmen. In einigen speziellen Aufgaben, zum Beispiel in der Medizin, können sie erfolgreich eingesetzt werden, aber in alltäglichen Situationen, in denen visuelle Informationen verarbeitet werden müssen, ist die Leistung menschlicher Gehirne noch unerreicht. Ein Grund ist sicherlich die Fähigkeit des Menschen, enorme Datenmengen parallel verarbeiten zu können. Ein weiterer mag die enorme Menge an Kontextwissen sein, die Menschen bei der Erschließung einer Szene einsetzen können. So weiß man zum Beispiel, wo man nach einem Gesicht suchen muß, wenn man schon einen Oberkörper oder einen Hut gefunden hat.

Mit fortschreitender Entwicklung der Computertechnik besteht allerdings die Hoffnung, daß diese Probleme zufriedenstellend gelöst werden können. Anwendungsbeispiele für eine bloße robuste Gesichtserkennung sind zahlreich. Die politisch unstrittigste Anwendung ist sicherlich die Personenidentifikation bei Zugangskontrollen. Da man die Randbedingungen hierbei relativ gut kontrollieren kann, sind Systeme hierfür auch schon erfolgreich im Einsatz.

Das automatische Identifizieren und Erkennen von Gesichtern in Massenszenen ist noch eine sehr schwierige Aufgabe, da hierbei alle Randbedingungen wie Größe der Gesichter, Beleuchtung, Pose und so weiter stark variieren. Bei einem Einsatz solcher Verfahren auf Kameras im öffentlichen Raum sind natürliche Aspekte des Datenschutzes zu beachten.

### 1.2 Problemstellung

Für das Erkennen von Gesichtern wurde am Institut für Neuroinformatik die Methode der Graphenanpassung entwickelt, bei der das Umfeld markanter Punkte verglichen wird. Diese Punkte werden Landmarken genannt und in einem Graphen zusammengefasst.

So wird unter anderem die Nasenspitze einer zu erkennenden Person mit den gespeicherten Nasenspitzen von Personen verglichen, die in einer Datenbank vorhanden sind. Voraussetzung für einen sinnvollen Vergleich dieser Art ist allerdings, daß man zunächst einmal die Position der zu vergleichenden Landmarke im neuen Bild findet. Die korrespondierenden Punkte zwischen den Bildern in einer Datenbank und in einem neuen Bild zu finden, ist äußerst schwierig, aber notwendig, um,

sprichwörtlich gesagt, nicht Äpfel mit Birnen zu vergleichen. Das Computersystem soll in die Lage versetzt werden, aufgrund von gespeicherten Nasenspitzen die Position einer anders aussehenden Nasenspitze exakt zu lokalisieren.

Eine genaue Positionierung der Landmarken ist nicht nur für den Vergleich nötig. Sie ist auch für eine automatische Aufnahme von neuen Gesichtern in eine Datenbank, bei der kein Mensch mehr von Hand die Positionen der Landmarken setzen muß, wünschenswert. Generell besteht bei dem automatischen Aufbau einer Datenbank das Problem, daß sich mit zunehmender Anzahl von Gesichtern die Datenbasis immer mehr verbreitert. Wenn bei jedem neuen Bild kleine Ungenauigkeiten in der Position der Landmarken auftreten, so können die Fehler bei der nächsten Positionsbestimmung weiter wachsen, bis keine sinnvolle Graphenanpassung mehr möglich ist.

Für den Vergleich zweier Gesichter durch den Vergleich vieler markanter Punkte besteht die naheliegende Vermutung, daß auch hier eine möglichst genaue Landmarkenpositionierung für eine höhere Erkennungsleistung nötig sein könnte, was zu überprüfen wäre.

Daher sollen in dieser Arbeit, ausgehend von dem Verfahren der Graphenanpassung, Methoden zur Verbesserung der Landmarkenpositionierung gefunden und untersucht werden. Mit einem erfolgreichen Verfahren können so bei Anwendung auf zwei neue Gesichtsbilder die korrespondierenden Punkte besser gefunden werden.

# Kapitel 2

## Bisherige Arbeiten

### 2.1 Neuronale Netze

Kanade hat ein System zum Gesichterfinden entwickelt, das neuronale Netze verwendet. Um die Funktionsweise dieses Systems besser verstehen zu können, folgt hier zunächst eine kurze Einführung zur prinzipiellen Idee der neuronalen Netze. (Für ein grundlegendes Verständnis ist Literatur wie zum Beispiel [1] geeignet.)

Die menschliche Intelligenz und die Rechenleistung des Gehirns steckt im wesentlichen in der Hirnrinde, dem Sitz der Nervenzellen. Die Informationsverarbeitung im Gehirn findet durch die Weiterleitung und Verarbeitung von elektrischen Aktionspotentialen statt. Betrachtet man eine einzelne Nervenzelle, so lassen sich drei Hauptstrukturen ausmachen, der Dendritenbaum, der Zellkörper (Soma) und das Axon. Diesen Strukturen könnte man die Funktionen Eingabe, Verarbeitung und Ausgabe zuordnen. Der Dendritenbaum sammelt Ausgabesignale anderer Neuronen in Form eines elektrischen Potentials, das dem Zellkörper zugeleitet wird. Bei Überschreitung eines Schwellwertes erzeugt der Zellkörper einen elektrischen Impuls. Das Axon leitet diesen Impuls über die Synapsen an die Dendriten oder auch direkt an andere Zellkörper weiter. Dabei regelt der Zustand der Synapse die Stärke der Weiterleitung. Die sowohl erregend als auch hemmend wirkenden Synapsen ermöglichen Lernprozesse durch die Veränderung dieser Weiterleitungsstärke.

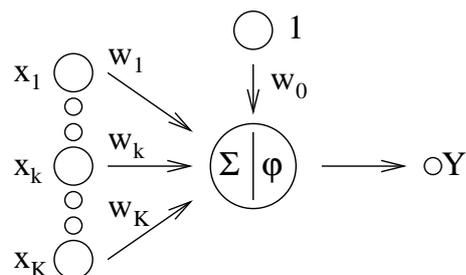


Abb. 2.1: Ein formales Neuron

Der erste Vorschlag zur Modellierung eines Neurons kam von McCulloch und Pitts 1943 [2]. Hierbei war ein Eingang  $x_i$  entweder 0 oder 1, die Synapsenstärke betrug  $\pm 1$ . Wurde so ein Schwellwert  $s$  überschritten, so feuerte das Neuron ( $Y = 1$ ), andernfalls nicht ( $Y = 0$ ).

$$Y = \varphi \left( \sum_i w_i x_i - s \right) \tag{2.1}$$

Da man als Spezialfälle von 2.1 sowohl UND-Gatter als auch NOT-Gatter erhalten kann, lassen sich mit diesem Neuron beliebige logische Schaltungen realisieren.

Das von Rosenblatt [3] vorgeschlagene Perzeptron ist in der Lage, Klassifizierungsaufgaben zu lösen. Das Muster, das zu einer von  $N$  verschiedenen Musterklassen gehört, wird hierzu als Eingabedaten  $x = (x_1, x_2, \dots, x_L)$  in das neuronale Netz gegeben. Nach der Trainingsphase soll nur dasjenige der  $N$  Neurone feuern, das dem Muster zugeordnet ist.

$$Y_r = \varphi \left( \sum_{i=1}^K w_{ri} x_i \right) \quad (2.2)$$

Liefert eines der Neurone  $Y_r$  nicht den korrekten Ausgabewert  $\delta_r$  bei der Präsentation eines Musterbeispiels, so werden die Gewichte wie folgt angepasst:

$$\Delta w_{ri} = \epsilon_r \cdot (\delta_r - Y_r) \cdot x_i \quad (2.3)$$

$$\epsilon_r = \frac{\sum_i w_{ri} x_i}{\sum_i x_i^2} \quad (2.4)$$

Diese Regel ist als Perzeptron-Lernalgorithmus bekannt [1].

Hiermit sollte nur kurz eine Idee davon vermittelt werden, wie prinzipiell mit neuronalen Netzen verschiedene Muster unterschiedlichen Klassen zugeordnet werden können.

## 2.2 Gesichterfinden mit neuronalen Netzen

Kanade hat in [4] ein System vorgestellt, mit dem Gesichter in einem Bild gefunden werden, indem nach einer Vorverarbeitung alle Bildbereiche einem neuronalen Netz präsentiert werden, das dann entscheidet, ob es sich bei diesem Bildausschnitt, auch Template genannt, um ein Gesicht handelt oder nicht.

Zunächst werden alle 20 mal 20 Pixel großen, sich überschneidenden Teilbilder aus einem Bild betrachtet. Dann wird das Gesamtbild verkleinert, es werden wieder alle Teilbilder betrachtet und so weiter. Auf diese Weise sind alle potentiellen Gesichter in der richtigen Größe in einem 20 mal 20 Pixel - Template erfasst. Die Vorverarbeitung passt nun die Gesamthelligkeit der Templates an und führt eine Histogrammangleichung durch.

Die so erhaltenen Bilder sind die Eingabe für ein neuronales Netzwerk, das entscheidet, ob es sich um ein Gesicht handelt oder nicht. Die versteckten Elemente des Netzwerkes betrachten dabei verschiedene rezeptive Felder, und so erhalten die Neurone Teilbilder des Templates als Eingabe. In der Nachbearbeitung wird ein Gesicht nur dann an einer Position als Gesicht erkannt, wenn dort an mehreren Positionen und für verschiedene Skalen das Netz eine positive Ausgabe liefert. So kann die Anzahl falsch positiver Entscheidungen, also die Zahl der fälschlicherweise als Gesicht klassifizierter Bereiche, deutlich reduziert werden.

In [5] stellt Kanade eine Erweiterung dieses Systems vor, mit dem es nun auch möglich ist, im Bild gedrehte Gesichter zu erkennen.

Hierbei wird ein neuronales Netzwerk darauf trainiert, den Winkel zu schätzen, um den das Template, unter der Annahme, daß es ein Gesicht zeigt, gedreht ist. Nun wird das Template um diesen Winkel gedreht und damit ein mögliches Gesicht in eine aufrechte Position gerückt. Damit ist das schon erläuterte System in der Lage zu entscheiden, ob an dieser Position wirklich ein Gesicht zu sehen ist.

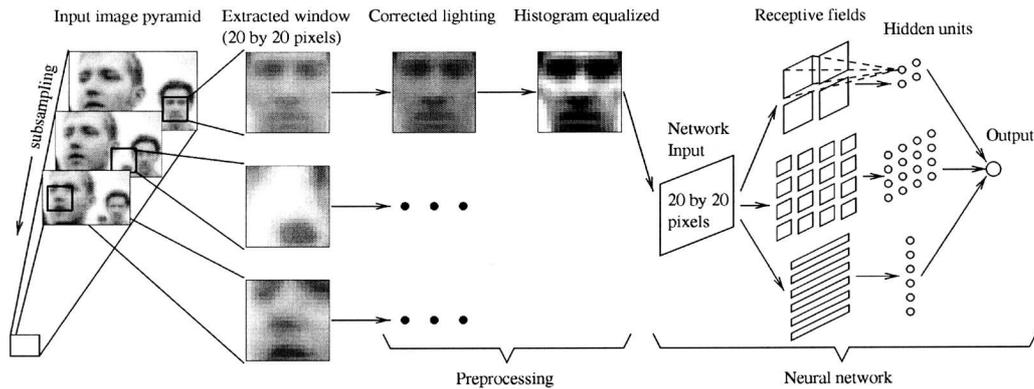


Abb. 2.2: Gesichterfinden mit neuronalen Netzen nach Kanade. Die Abbildung entstammt [4].

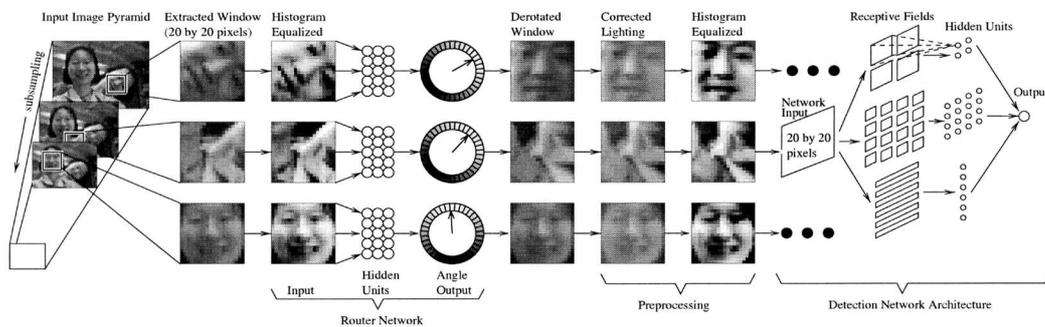


Abb. 2.3: Rotationsinvariantes Gesichterfinden mit neuronalen Netzen nach Kanade. Die Abbildung entstammt [5].

## Bewertung

Das System von Kanade ist in der Lage, die meisten Gesichter in einem Bild zu finden, bei einer sehr geringen Zahl falsch positiver Entscheidungen. Es zeichnet sich aber nicht nur durch diese Robustheit aus, sondern ist dabei auch sehr schnell.

Allerdings liefert es eben nur die Position eines Gesichtes. Mit dem im folgenden dargestellten Verfahren der Graphenanpassung erhält man hingegen nicht nur die Position des Gesichtes, sondern auch die Position markanter Punkte, wie die Augen- oder Mundwinkel oder die Nasenspitze. Damit ist für die Aufgabe der Gesichtserkennung eine ganz wesentliche Arbeit, das Finden von Korrespondenzen, schon geleistet, was das System von Kanade nicht vermag.

## 2.3 Anpassung von Graphen

Wie schon in der Problemstellung angedeutet, wurde am Institut für Neuroinformatik ein biologisch motiviertes Verfahren entwickelt, um visuelle Erkennungsaufgaben zu lösen.

Es wurden Strukturen in der primären Sehrinde entdeckt, die auf dem von den Augen wahrgenommenen Bild eine lokale Merkmalsanalyse durchführen. Diese sogenannten simple cells reagieren auf Objekte, wie z.B. Kanten, und ihre Reaktion ist von der Richtung der Kante abhängig. 1987 fanden Jones und Palmer [6] durch Untersuchungen an Katzenshirnen heraus, daß diese Zellen eine lokale Frequenzanalyse auf dem Bild durchführen. Das Antwortverhalten der simple cells kann durch Gaborfunktionen modelliert werden, die als Filter auf das Bild angewendet werden. Eine Ga-

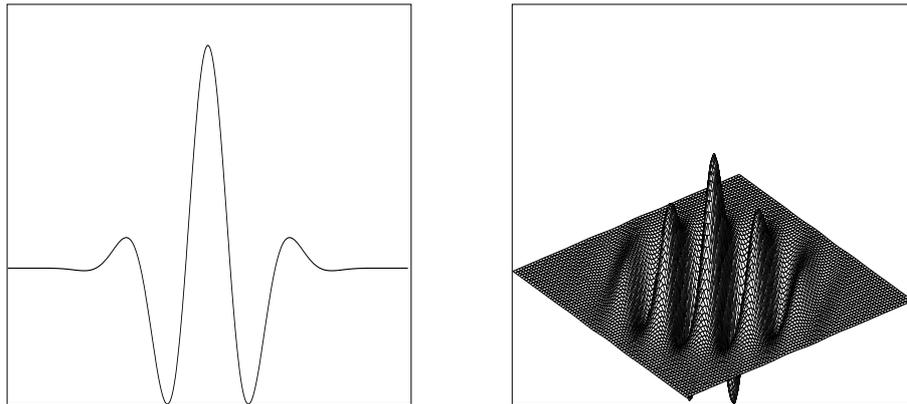


Abb. 2.4: Ein- und zweidimensionale Gaborfunktion

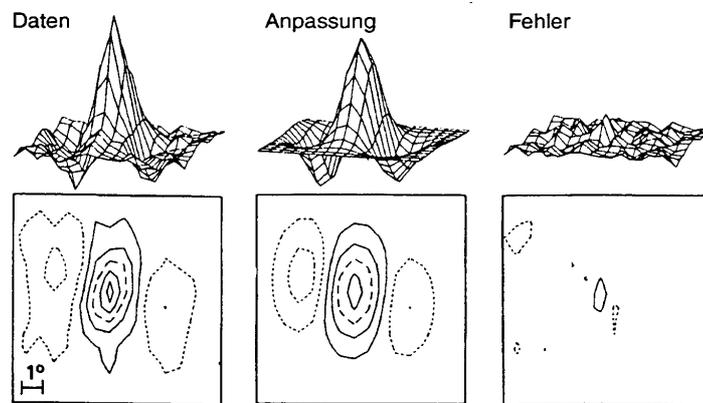


Abb. 2.5: Antwortverhalten der Nervenzellen und der Gaborfunktion [6]

borfunktion ist das Produkt einer ebenen Welle mit einer umhüllenden Gaußfunktion. Gleichung 2.5 zeigt die allgemeine Form.

$$G(\vec{a}) = A \cdot \exp\left(-\frac{1}{2}(\vec{a} - \vec{a}_0) \cdot D_0(\vec{a} - \vec{a}_0)\right) \cdot \exp\left(-i\vec{k}_0 \cdot \vec{a}\right) \quad (2.5)$$

Hierbei ist  $D_0$  eine positiv definite Matrix.  $A$ ,  $\vec{k}_0$  und  $\vec{a}_0$  sind beliebige Parameter. Die Antwort des Gaborfilters erhält man durch Faltung der Eingabe mit der Gaborfunktion.

$$O_G(I, \vec{a}_0) = \int I(\vec{a}) \cdot G(\vec{a}_0 - \vec{a}) d\vec{a}^2 \quad (2.6)$$

Anschaulich bedeutet die Faltung, daß man für jeden Ausgabepunkt  $O$  den Faltungskern (die Gaborfunktion) an dieser Stelle über das Bild legt und die Produkte aus Eingabe und Faltungskern für diesen Punkt aufsummiert. Auf diese Weise wird eine lokale Merkmalsanalyse durchgeführt, die Filterantwort reagiert auf Frequenzen, die in Richtung der ebenen Welle liegen und durch die Gaußfunktion lokal begrenzt sind.

Messungen ergeben eine sehr gute Übereinstimmung zwischen Zellenantwort und Gaborfilter [6].

Die Betrachtungsweise von Denes Gabor war allerdings eine ganz andere, nicht biologisch motivierte.

Betrachtet man den  $2N$ -dimensionalen Phasenraum, der durch den  $N$ -dimensionalen Orts- und den  $N$ -dimensionalen Impulsraum gegeben ist, und durch die dazwischen definierte Fouriertransform,

$$\tilde{F}(\vec{k}) = \frac{1}{\sqrt{2\pi}^N} \int F(\vec{a}) e^{i\vec{k}\vec{a}} da^n \quad (2.7)$$

$$F(\vec{a}) = \frac{1}{\sqrt{2\pi}^N} \int \tilde{F}(\vec{k}) e^{-i\vec{k}\vec{a}} dk^n \quad (2.8)$$

so kann man zeigen, daß zwischen Orts- und Frequenzraum eine Unschärferelation besteht, die besagt, daß das Produkt der Varianzen im Orts- und Frequenzraum nie unter einen festen Wert sinken kann.

$$\prod_{i=1}^N \Delta a_i(F) \cdot \Delta k_i(\tilde{F}) \geq \frac{1}{2^N} \quad (2.9)$$

Ein Spezialfall hiervon ist die Unschärferelation der Quantenmechanik, die besagt, daß Ort und Impuls eines Teilchens nicht gleichzeitig exakt messbar sind.

$$\Delta x \cdot \Delta p_x \geq \frac{\hbar}{2} \quad (2.10)$$

Die von Denes Gabor entwickelte Gaborfunktion hat die Eigenschaft, daß für sie das Produkt der Varianzen minimal ist, also in der Unschärferelation das Gleichheitszeichen steht. Damit sind die Gaborfunktionen optimal im Orts- wie im Frequenzraum lokalisiert. Ihre Antworten sind komplexe Zahlen.

Für den Einsatz in der Bildverarbeitung werden die Parameter speziell gewählt.

D. J. Field hat festgestellt [7], daß in der Fourierertransformierten von natürlichen Bildern das Amplitudenquadrat, also so etwas wie der Informationsgehalt, proportional mit dem Kehrwert des Quadrates der Frequenz abnimmt.

Um eine optimale Kodierung der Information zu erreichen und um Mittelwertfreiheit zu gewährleisten, hat die am Institut eingesetzte Funktion folgende Form:

$$G(\vec{a}) = \frac{\vec{k}^2}{\sigma^2} \exp\left(-\frac{\vec{k}^2(\vec{a} - \vec{a}_0)^2}{2\sigma^2}\right) \cdot \left(\exp(-i\vec{k}(\vec{a} - \vec{a}_0)) - \exp\left(-\frac{\sigma^2}{2}\right)\right) \quad (2.11)$$

Dabei ist  $\vec{a}_0$  der Mittelpunkt des Wellenzuges und  $\vec{a}$  ein Vektor des Argumentbereichs.

Geht man davon aus, daß es eine Grenzfrequenz für die im Bild auftretenden Strukturen gibt, so reicht für die Darstellung eines Bildmerkmals eine endliche diskrete Zahl an Skalen (und Orientierungen) aus. Im technischen Einsatz wird ein Satz von Filtern erzeugt, der im allgemeinen aus 5 verschiedenen Skalen mit je 8 verschiedenen Orientierungen besteht. Hierbei gilt  $\sigma = 2\pi$  und für  $\vec{k}$  sind die Vektoren für die Skalen und Orientierungen wie folgt:

$$\vec{k}_i = \begin{pmatrix} k_\nu \cos \theta_\mu \\ k_\nu \sin \theta_\mu \end{pmatrix} \quad (2.12)$$

$$i = \nu + (\nu_{\max} + 1)\mu \quad (2.13)$$

$$k_\nu = 2^{-\frac{\nu+2}{2}} \pi, \quad \text{mit } \nu = 0, \dots, \nu_{\max} \quad (2.14)$$

$$\theta_\nu = \mu \frac{\pi}{\mu_{\max} + 1}, \quad \text{mit } \mu = 0, \dots, \mu_{\max} \quad (2.15)$$

Was Skala und Orientierung des Gaborkerns im Ortsraum sind, nämlich Frequenz und Richtung der Welle, sind im Frequenzraum der Abstand des Mittelpunkts der Gaußglocke von dem Nullpunkt und der Winkel von einer Frequenzachse.

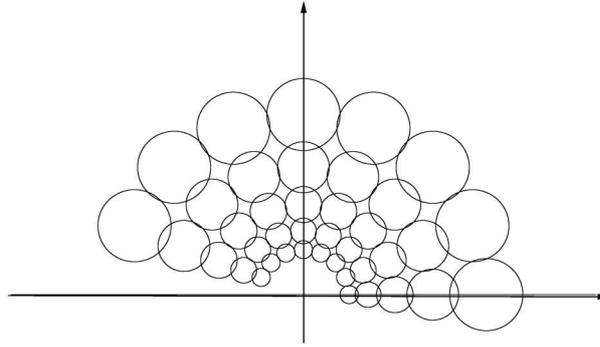


Abb. 2.6: Gaußglocken für 5 Skalen und 8 Orientierungen im Frequenzraum. Die Achsen sind die Frequenzachsen  $\omega_1$  und  $\omega_2$ .

Diese Antworten des Filters kann man zu einem 40dimensionalen Vektor zusammenfassen, im folgenden FeaSt (Feature-Set) genannt. Normiert man diesen Vektor, so ist ein Vergleich zweier FeaSts über ein einfaches Skalarprodukt der Vektoren möglich.

Für die Berechnung der Ähnlichkeit zweier Vektoren  $\vec{b}$  und  $\vec{b}'$ , bei denen sowohl der Absolutwert als auch die Phaseninformation berücksichtigt wird, im folgenden AbsPhaseFeaSt genannt, gilt die Ähnlichkeitsfunktion:

$$s_{\text{AbsPhase}} = \frac{\sum_{\vec{k}} b_{\vec{k}} b'_{\vec{k}} \cos(\varphi_{\vec{k}} - \varphi'_{\vec{k}})}{\sqrt{\sum_{\vec{k}} b_{\vec{k}}^2 \sum_{\vec{k}} b'^2_{\vec{k}}}} \quad (2.16)$$

Will man nur die Absolutwerte für die Berechnung der Ähnlichkeiten benutzen, fällt der cos-Term weg.

$$s_{\text{Abs}} = \frac{\sum_{\vec{k}} b_{\vec{k}} b'_{\vec{k}}}{\sqrt{\sum_{\vec{k}} b_{\vec{k}}^2 \sum_{\vec{k}} b'^2_{\vec{k}}}} \quad (2.17)$$

Bei erwähnter Normierung fällt bei diesen Termen dann auch der Nenner weg.

Für die Darstellung eines Gesichtes betrachtet man die FeaSts an besonders markanten Punkten, die man leicht im Gesicht wiederfinden kann, wie Pupillen oder Nasenspitze. Diese Punkte werden Landmarken genannt. In dieser Arbeit wurden 48 Landmarken auf natürlichen und 52 auf künstlichen Gesichtern verwendet. Die Erhöhung der Landmarkenzahl auf den künstlichen Gesichtern führte dazu, daß bei Posenvariationen mehr Landmarken sichtbar blieben.

Um Informationen über die räumliche Anordnung der als Knoten bezeichneten Punkte auszunutzen (daß z.B. bei aufrechten Gesichtern die Augen über der Nase sitzen usw.), werden die Knoten mittels Kanten verbunden und so zu einer Graphenstruktur, im weiteren Modellgraph [8] genannt, zusammengefasst. Ein Bild eines solchen Graphen auf einem Gesicht ist im Kapitel über die verwendeten Daten in Abb. 3.2 zu sehen.

Um mehrere Gesichter in einer Datenstruktur zusammenzufassen, gibt es das Konstrukt des Bündelgraphen oder Bunchgraphen [9]. In ihm hängen an jedem Knoten die Vielzahl aller FeaSts der jeweiligen Gesichter. Abb. 2.7 veranschaulicht dies graphisch. Abb. a zeigt einen Gaborfilter, mit dem ein Eintrag des FeaSts erzeugt wird, in b sieht man schematisch die Zusammenfassung zu einem FeaSt und in c ist dann die Gruppierung verschiedener FeaSts zu einem Bündelgraphen dargestellt. Besteht ein Bündelgraph mit 52 Knoten zum Beispiel nur aus 5 Graphen, so lassen sich damit  $5^{52}$  verschiedene Gesichter kombinieren.

Für die Identifikation eines Gesichtes in einem Bild und die Erkennung dieser Person durch Vergleich mit einer Datenbank wird das Verfahren der Bündelgraphenanpassung oder des Bunchgraphmatchings angewendet.

Dabei wird der Bündelgraph über das Bild bewegt und die Ähnlichkeit der im Bild befindlichen

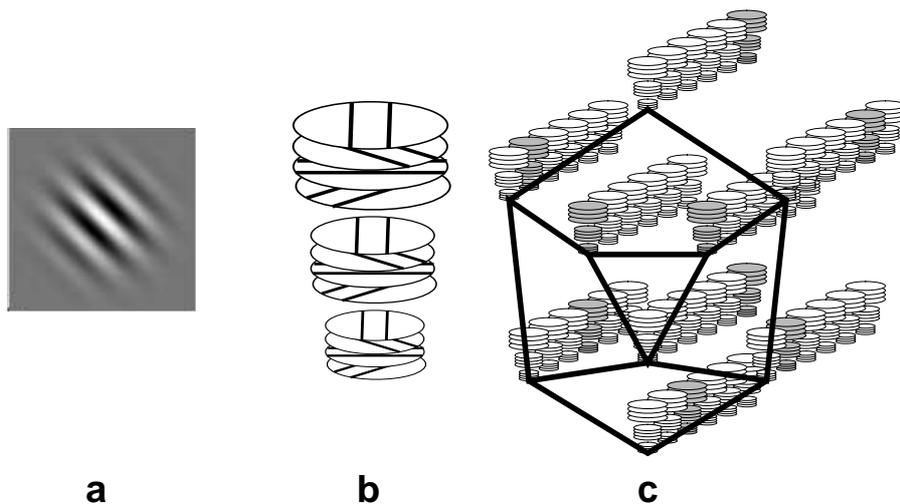


Abb. 2.7: Darstellung des Bündelgraphen

FeaSts mit denen im Bündelgraphen verglichen und aus den ähnlichsten FeaSts die Gesamtähnlichkeit berechnet. Ist ein Schwellwert überschritten, so wurde an dieser Stelle ein Gesicht gefunden. Sucht man nur nach einem Gesicht, so wählt man die Position mit der höchsten Ähnlichkeit als Position des Gesichtes.

Bisher wurde der Graph nur starr über das Bild bewegt. Für eine genaue Positionierung der Landmarken gibt es mehrere Bewegungsmodi (moves), die über ein Parameterfile die Bewegung der Knoten steuern.

#### ScanGlobalMove

In dem schon beschriebenen, meist ersten, Move wird der Graph starr über das Bild bewegt. Dabei ist die Einstellung der Schrittweite und die Angabe einer Region möglich, um so das Bild zunächst grob abzutasten, und dann in einem passenden Bereich genauer nach einem Gesicht zu suchen.

#### ScanScaleXYMove

Hier wird der gesamte Graph in X- und Y-Richtung skaliert, hauptsächlich um sich an unterschiedlich große Gesichter anzupassen. In begrenztem Maße ist es aber so auch möglich, sich einer Posenvariation anzupassen. Da für ein leicht zur Seite weggedrehtes Gesicht die Merkmale in x-Richtung näher aneinanderrücken, kann man den Graphen durch ein unabhängiges Skalieren der x- und y-Richtung dieser Posenvariation anpassen.

#### ScanLocalMove

Hier wird jeder Knoten einzeln in seinem Umfeld bewegt. Da hierbei der Graph deformiert wird, gibt es einen topographischen Faktor, der bei immer stärkerer Deformation den Ähnlichkeitswert verringert und so ein zu starkes Wandern der Knoten verhindert. Schließlich sinkt mit stärker werdender Deformation des Graphen die Wahrscheinlichkeit, daß die Landmarken immer noch richtig liegen.

#### DiparityMove

Bei dem DisparityMove wird über die Phaseninformation die Verschiebung der Knoten geschätzt und der Knoten dann in diese Richtung bewegt. Dies bringt gegenüber dem LocalMove im wesentlichen Geschwindigkeitsvorteile. Der DisparityMove wurde in dieser Arbeit jedoch nicht verwendet.

## Bewertung

Auch wenn das Prinzip der Graphenanpassung für das reine Gesichterfinden nicht ganz die Robustheit und Schnelligkeit des von Kanade entwickelten Systems hat, so bietet es doch einen entscheidenden Vorteil. Hat man ein Gesicht gefunden, so sind zur gleichen Zeit die Position der Korrespondenzen bekannt, und man kann sehr schnell die Person mit einer Datenbank mittels eines einfachen Skalarproduktes vergleichen.

Eine Möglichkeit, das Suchen dieser Korrespondenzen auf der Basis eines sehr großen Bündelgraphen wesentlich zu beschleunigen, stellt das in Kapitel 4.3 beschriebene PCA-Matching dar.

## 2.4 Hauptkomponentenanalyse (PCA)

Da das PCA-Matching in dieser Arbeit von zentraler Bedeutung ist, folgt nun eine kurze Einführung in die Hauptkomponentenanalyse, im Englischen Principal Component Analysis (PCA) genannt.

Die Hauptkomponentenanalyse ist eine Methode, Variablen in einem hochdimensionalen Raum durch eine kleinere Anzahl unkorrelierter Variablen zu ersetzen, die im wesentlichen die gesamte Varianz der Ursprungsdaten erklären. Auf diese Weise ist es möglich, die effektive Dimension der Daten zu reduzieren.

Um ein einfaches, idealisiertes Beispiel zu geben, kann man sich vorstellen, man habe Daten im zweidimensionalen Raum, die alle auf einer Geraden liegen. Dann liegt der Vektor der ersten Hauptkomponente in Richtung der Geraden und erklärt die gesamte Varianz. Die Hauptkomponentenanalyse hat die Dimensionalität auf eins reduziert.

Bei Aufgaben im hochdimensionalen Raum werden nacheinander die Hauptkomponenten gesucht, die zueinander orthogonal liegen und die größte Varianz der Daten liefern.

Zur Erinnerung sei hier noch einmal die Definition von Varianz und Kovarianz angeführt. Mit den Zufallsvariablen  $X$ ,  $Y$  und dem Erwartungswert  $E$  ist

$$\begin{aligned}\text{Var}(X) &= E((X - E(X))^2) \\ \text{Cov}(X, Y) &= E((X - E(X))(Y - E(Y)))\end{aligned}$$

Die Kovarianzmatrix  $\Sigma$  einer mehrdimensionalen Zufallsvariablen  $\vec{X} = (X_1, \dots, X_m)$  ist

$$\Sigma_{ij} = \text{Cov}(X_i, X_j) = \begin{pmatrix} \text{Cov}(X_1, X_1) & \dots & \text{Cov}(X_1, X_m) \\ \vdots & \ddots & \vdots \\ \text{Cov}(X_m, X_1) & \dots & \text{Cov}(X_m, X_m) \end{pmatrix}$$

Sei  $\vec{y} = (y_1, \dots, y_m)$  ein  $m$ -dimensionaler Vektor der Ursprungsdaten, der Erwartungswertvektor der Daten sei  $\vec{\mu}$  und die Kovarianzmatrix  $\Sigma$ . So besteht nun die Aufgabe, neue Variablen  $z_1, \dots, z_m$  zu finden, die unkorreliert sind und deren Varianzen mit steigendem Index  $j = 1, \dots, m$  fallen. Dabei ist  $z_j$  eine Linearkombination der  $y_j$ :

$$z_j = a_{1j}y_1 + a_{2j}y_2 + \dots + a_{mj}y_m = \vec{a}_j \vec{y}$$

Dabei ist der Vektor  $\vec{a}_j$  die  $j$ -te Hauptkomponente. Für die Vektoren  $\vec{a}_j$  gilt die Orthonormalität

$$\vec{a}_i \vec{a}_j = \delta_{ij} \tag{2.18}$$

wodurch der euklidische Abstand erhalten bleibt. Für die erste Hauptkomponente soll gelten, daß die Daten in dieser Richtung die maximale Varianz aufweisen, daß also

$$\text{Var}(z_1) = \text{Var}(\vec{a}_1 \vec{y}) = \vec{a}_1 \Sigma \vec{a}_1 \tag{2.19}$$

maximal wird. Um die Zielfunktion  $\vec{a}_j \Sigma \vec{a}_j$  unter der Nebenbedingung der Normierung  $\vec{a}_j \vec{a}_j = 1$  zu maximieren, kann man die Methode der Lagrange-Multiplikatoren verwenden.

Sei  $f(y_1, \dots, y_m)$  eine unter der Nebenbedingung  $g(y_1, \dots, y_m) = c$  zu maximierende differenzierbare Funktion, so existiert ein Lagrangemultiplikator  $\lambda$ , so daß in den stationären Punkten gilt:

$$\frac{\partial f}{\partial y_i} - \lambda \frac{\partial g}{\partial y_i} = 0 \quad i = 1, \dots, m \quad (2.20)$$

Damit ist eine Bestimmung der Koordinaten der stationären Punkte und der Werte von  $\lambda$  möglich. Für einen stationären Punkt  $\vec{a}$  einer differenzierbaren Funktion sind alle partiellen Ableitungen gleich Null, also

$$\frac{\partial f}{\partial \vec{y}} = \text{grad} f(\vec{a}) = 0$$

$\frac{\partial f}{\partial \vec{y}}$  sei hier als Spaltenvektor, bestehend aus den partiellen Ableitungen  $\frac{\partial f}{\partial y_i}$ , definiert. In der neu definierten, um die Nebenbedingung erweiterten Lagrangefunktion

$$L(\vec{y}, \lambda) = f(\vec{y}) - \lambda[g(\vec{y}) - c]$$

ist der Term in der eckigen Klammer wegen der Nebenbedingung gleich null. Damit wird Gleichung 2.20 zu

$$\frac{\partial L}{\partial \vec{y}} = 0$$

Für dieses Problem sieht die Lagrangefunktion wie folgt aus:

$$L(\vec{a}_1) = \vec{a}_1 \Sigma \vec{a}_1 - \lambda(\vec{a}_1 \vec{a}_1 - 1)$$

Da

$$\frac{\partial \vec{a}_1 \Sigma \vec{a}_1}{\partial \vec{a}_1} = 2 \Sigma \vec{a}_1$$

folgt

$$\frac{\partial L}{\partial \vec{a}_1} = 2 \Sigma \vec{a}_1 - 2 \lambda \vec{a}_1$$

und gleich null gesetzt erhält man:

$$(\Sigma - \lambda I) \vec{a}_1 = 0 \quad (2.21)$$

Dabei ist  $I$  die Einheitsmatrix. Diese Gleichung hat genau dann eine vom Nullvektor verschiedene Lösung, wenn

$$\det(\Sigma - \lambda I) = 0$$

oder anders gesagt, wenn  $\lambda$  ein Eigenwert der Matrix  $\Sigma$  ist. Die Lösung der Gleichung 2.21 führt im allgemeinen zu  $m$  von Null verschiedenen Eigenwerten  $\lambda_j$ . Diese sind nicht negativ, weil  $\Sigma$  positiv semidefinit ist.

Mit Gleichung 2.21 gilt nun

$$\text{Var}(\vec{a}_1 \vec{y}) = \vec{a}_1 \Sigma \vec{a}_1 = \vec{a}_1 \lambda I \vec{a}_1 = \lambda \vec{a}_1 I \vec{a}_1 = \lambda \vec{a}_1 \vec{a}_1 = \lambda$$

Damit wählt man als 1. Hauptkomponente den zum größten Eigenwert  $\lambda_j$  gehörenden Eigenvektor  $\vec{a}_1$ .

Auf der Suche nach der 2. Hauptkomponente gilt als zusätzliche Nebenbedingung, daß  $\text{Cov}(z_2, z_1) = 0$ , also

$$\text{Cov}(z_2, z_1) = \text{Cov}(\vec{a}_2 \vec{y}, \vec{a}_1 \vec{y}) = \vec{a}_2 \Sigma \vec{a}_1 = \vec{a}_2 \lambda_1 \vec{a}_1 = 0$$

Damit muß

$$\vec{a}_2 \vec{a}_1 = 0$$

Es sind daher 2 Lagrangemultiplikatoren  $\lambda$  und  $\delta$  zu wählen und als Lagrangefunktion ergibt sich

$$L(\vec{a}_2) = \vec{a}_2 \Sigma \vec{a}_2 - \lambda(\vec{a}_2 \vec{a}_2) - \delta \vec{a}_2 \vec{a}_1$$

und erhalten

$$\frac{\partial L}{\partial \vec{a}_2} = 2(\Sigma - \lambda I) \vec{a}_2 - \delta \vec{a}_1 = 0 \quad (2.22)$$

Mit  $\vec{a}_1$  von links multipliziert ergibt sich

$$2\vec{a}_1 \Sigma \vec{a}_2 - 2\lambda \vec{a}_1 \vec{a}_2 - \delta \vec{a}_1 \vec{a}_1 = 2\vec{a}_1 \Sigma \vec{a}_2 - \delta = 0$$

Weil  $\text{Var}(z_2, z_1) = \vec{a}_1 \Sigma \vec{a}_2 = 0$ , muß  $\delta = 0$  sein. Damit wird Gleichung 2.22 zu

$$\det(\Sigma - \lambda I) \vec{a}_2 = 0$$

Analog zur 1. Hauptkomponente wählt man für die 2. Hauptkomponente den zweitgrößten Eigenwert  $\lambda_j$  und den dazugehörigen Eigenvektor  $\vec{a}_j$ . Bildet man die Matrix der Hauptkomponenten  $A = [\vec{a}_1, \dots, \vec{a}_m]$ , so gilt

$$\vec{z} = A^\dagger \vec{y}$$

Die Kovarianzmatrix  $\Lambda$  von  $\vec{z}$  besteht aus den Eigenwerten  $\lambda_j$  auf der Hauptdiagonalen. Damit gilt als Beziehung zwischen Kovarianzmatrix von  $\vec{y}$  und den Hauptkomponenten:

$$\Sigma = A^\dagger \Lambda A$$

Das wichtige Ergebnis für das PCA-Matching ist, daß die Hauptkomponentenanalyse es möglich macht, sehr viele hochdimensionale Merkmalsvektoren zu einem Durchschnittsvektor und den Richtungsvektoren zusammenzufassen, die, nach Größe sortiert, die Varianz der Daten erklären. Hat man zum Beispiel eine Datenmenge von 300 80-dimensionalen Augen-FeaSts aus dem schon erwähnten 40 dimensionalen FeaSt-Vektor durch Aufspaltung in Real- und Imaginärteil gewonnen, so kann man die Verteilung dieser Daten durch einen Durchschnittsvektor und 80 Hauptkomponenten mit den zugehörigen Varianzen darstellen. Wie die Hauptkomponentenanalyse für die Graphenanpassung genutzt wird, wird in Kapitel 4.3 erklärt.

## 2.5 Eigengesichter (Eigenfaces)

Eine alternative Möglichkeit, die Hauptkomponentenanalyse für die Gesichtserkennung zu nutzen, ist das von Turk und Pentland [10] entwickelte Verfahren der Eigengesichter. Hierbei wird ein Bild als Vektor der Dimension Höhe mal Breite mit den Helligkeitswerten als Einträge aufgefasst. Nach Berechnung des Durchschnittsgesichtes werden die Abstände eines jeden der  $n$  Gesichter von dem Durchschnittsgesicht berechnet und als Spaltenvektoren in einer Matrix  $X$  der Dimension (Höhe mal Breite)  $\times n$  gespeichert. Um bei der Hauptkomponentenanalyse nicht die Kovarianzmatrix von  $XX^\dagger$  diagonalisieren zu müssen, welche die Dimension Höhe mal Breite hätte, berechnet man statt dessen die Matrix  $X^\dagger X$ , die nur die Dimension  $n^2$  hat. Aus den Eigenvektoren  $E_i$  dieser Matrix lassen sich dann die Eigengesichter über die Vorschrift  $F_i = X E_i$  berechnen.

Da hierbei die wesentlichen Informationen in den ersten Eigengesichtern enthalten sind, werden nur die Hauptkomponenten bis zu einer Schwelle betrachtet.

Für ein zu speicherndes Gesicht wird zunächst das Differenzbild zum Durchschnittsgesicht gebildet. Nach Projektion in den Gesichtsraum können die Koeffizienten der Linearkombinationen mit den Eigengesichtern abgespeichert werden. Diese Koeffizienten werden dann auch für Erkennungsaufgaben verglichen.

**Bewertung**

Das Verfahren der Eigengesichter eignet sich gut für die Erkennung von Gesichtern, weniger für das Finden von Gesichtern. Da es ein pixelbasiertes Verfahren ist, muß ein Gesicht in passender Größe und Pose vorliegen, nur dann werden hohe Erkennungsleistungen erreicht. Hierzu ist meistens ein vorgeschaltetes Verfahren nötig. Da dann allerdings das gesamte Bild verglichen wird, erhält man hierbei nicht die einzelnen Korrespondenzen.

## Kapitel 3

# Verwendete Daten und Vorverarbeitung

### 3.1 Künstliche Face-Gen Gesichter

Da in dieser Arbeit verschiedene Verfahren daraufhin untersucht werden, wie exakt sie das Korrespondenzproblem lösen, stellt sich zunächst die Frage, wie man dieses überhaupt bewerten kann. Sicherlich könnte man die vom Computer gefundenen Korrespondenzen mit den Positionen vergleichen, die ein Mensch gesetzt hätte, aber dabei stellt sich neben dem Problem der Subjektivität - jeder positioniert die Landmarken z. B. auf den Wangenknochen geringfügig anders - noch das Problem der Genauigkeit. Wenn sich die Fehler beim Finden der Position im Rahmen von 1 bis 3 Pixeln bewegen, so macht eine solche Aussage nur Sinn, wenn die wahre Position bekannt ist. Das ist aber bei Photos von Personen nicht gegeben.

Ein Programm der Firma Singular Inversions Inc. macht es möglich, künstliche Gesichter zu erzeugen, die in einem 3D-Modell vorliegen. Beim Rendern, bei dem Beleuchtung und Pose für das Bild der Person festgelegt werden, ist es durch eine Programm-Modifikation von Ingo Wundrich möglich, die exakte Position von vorher definierten Landmarken im Ergebnisbild anzugeben. Damit liegt nicht nur ein Bild der Person vor, sondern auch die dazugehörigen exakten Landmarkenpositionen, mit denen man nun den Erfolg einer Korrespondenzsuche bewerten kann.

Abbildung 3.1 zeigt ein mit FaceGen erzeugtes Gesicht. Die dort eingezeichneten Knotenpositionen sind in allen Versuchen dieser Arbeit verwendet worden. Sie decken das Gesicht und dabei besonders die strukturreichen Orte gut ab, und stellen auch sicher, daß bei leichten Posenvariationen alle Punkte noch sichtbar sind.

Diese Landmarken waren in allen Experimenten dieser Arbeit wie in Abbildung 3.2 gezeigt zu einem Graphen verbunden. Wie schon in Kapitel 2.3 erwähnt, spielen die Kanten hauptsächlich bei der Deformation des Graphen durch lokale Bewegungen einzelner Knoten eine Rolle. Dabei führt eine Längenänderung der Kanten zu einer langsamen Abnahme der Ähnlichkeit, weil mit zunehmender Deformation die Wahrscheinlichkeit abnimmt, ein Gesicht richtig zu finden.

Für die in dieser Arbeit durchgeführten Experimente wurden verschiedene Datensätze mit künstlichen Gesichtern erzeugt, abhängig von der Art des Experiments, aber auch von den technischen Möglichkeiten, die zu diesem Zeitpunkt vorhanden waren.

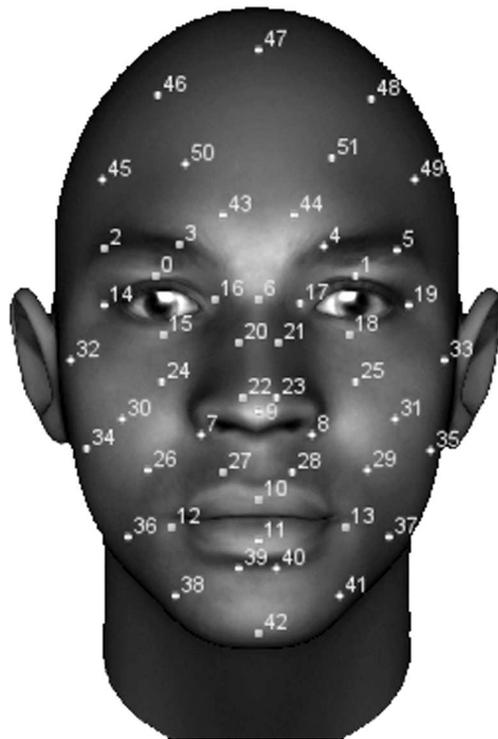


Abb. 3.1: Landmarkenpositionen bei einem FaceGen-Gesicht

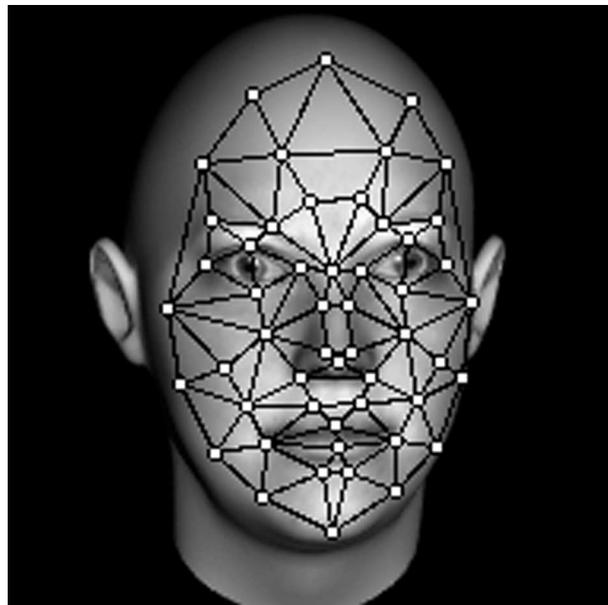


Abb. 3.2: Graphenstruktur

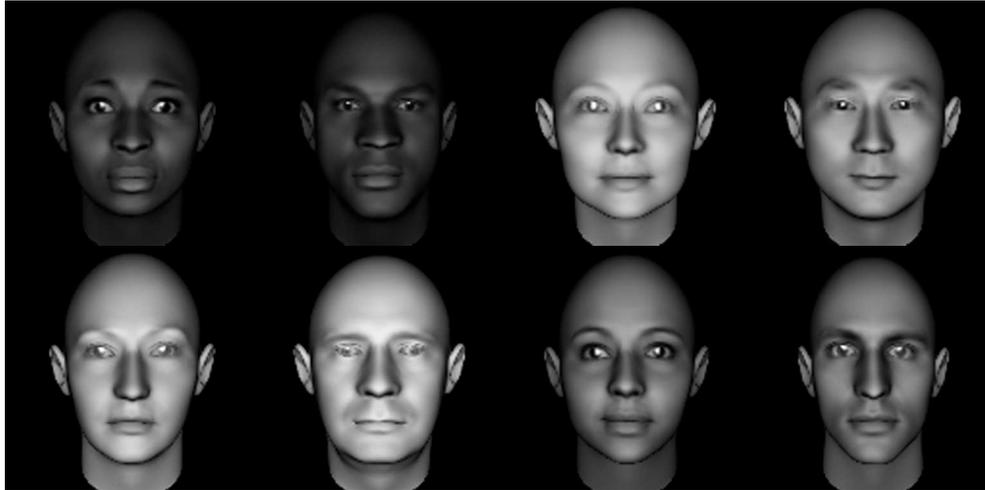


Abb. 3.3: Beispielbilder des Datensatzes FG101

In einem ersten Schritt wurde über eine graphische Benutzeroberfläche von Hand ein Satz von 101 Gesichtern erzeugt. Aufgrund der manuellen Erstellung sind diese Bilder einigermaßen gleich verteilt über Geschlecht und ethnische Abstammung und sehen zudem auch relativ realistisch aus. Dabei ist auf eine Posenvarianz verzichtet worden. Die Auflösung der Bilder in den Versuchen war  $128 \cdot 128$  Pixel. Diese Datenbank wird im Folgenden FG101 genannt.

Die Veränderung eines Perl-Skriptes zur Ansteuerung des FaceGenModellers ermöglichte es, nacheinander alle Textur- und Formparameter des Programms zu variieren und so per Script eine größere Datenbasis zu erzeugen. Da FaceGen aber bei Veränderung eines Parameters die übrigen Parameter automatisch anpassen mußte, um weiter ein gültiges Gesicht zu liefern, sahen die Gesichter weniger realistisch aus, als die vorher von Hand erstellten. Die Posen dieses Datensatzes (FG500) variierten von um  $5^\circ$  in x- und y-Richtung, und die Auflösung in den Versuchen betrug  $192 \cdot 192$  Pixel.

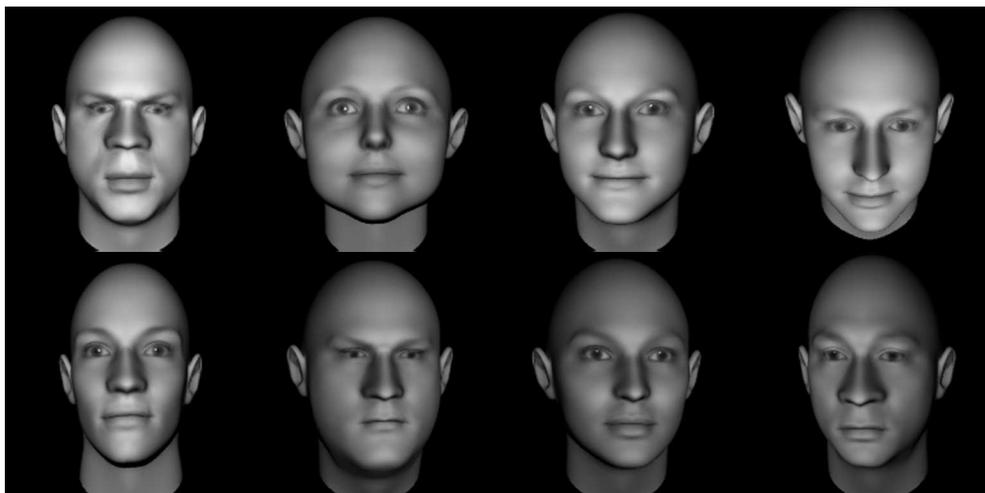


Abb. 3.4: Beispielbilder des Datensatzes FG500

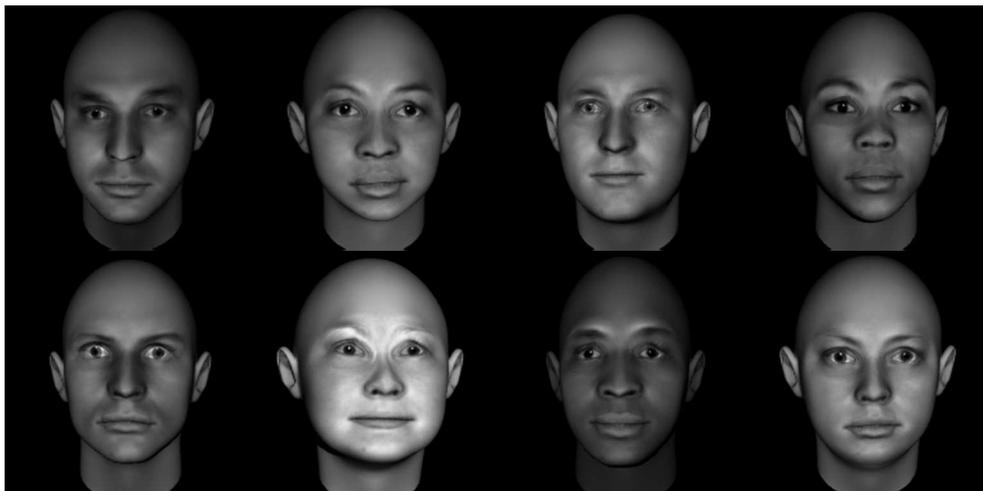


Abb. 3.5: Beispielbilder des Datensatzes FG1000

Eine neue Version von FaceGen erlaubt es, direkt per Scriptaufruf ein zufälliges Gesicht zu erzeugen, das dann in ähnlicher Qualität vorliegt, wie die zuerst von Hand erzeugten. Eine stärkere Variation der Texturparameter läßt diese Gesichter sogar noch realistischer erscheinen. Die entscheidenden Experimente in dieser Arbeit erfolgten mit einem so geschaffenen Satz von 1000 Gesichtern, die eine Posenvarianz von  $5^\circ$  aufweisen bei einer Auflösung von in der Regel  $192 \cdot 192$  Pixel (FG1000). Da in zahlreichen Anwendungen nicht nur das Finden eines Gesichtes und der Landmarken gefordert ist, sondern auch der Vergleich zweier Gesichter oder der Vergleich eines Gesichtes mit denen in in einer Datenbank, ist es sinnvoll, die Auswirkungen einer verbesserten Landmarkenpositionierung auf die Erkennungsleistung zu untersuchen. Zur Bewertung der Erkennungsleistung wurden zwei neue Galerien in der Art FG1000 von Gesichtern gleicher Personen in unterschiedlicher Pose verwendet. Die Posen variierten wieder im Bereich von  $5^\circ$ , die Auflösung hierbei betrug  $128 \cdot 128$  (FG2000).

## 3.2 Variation der Face-Gen Gesichter

Um eine Beurteilung der Ergebnisse der einzelnen Anpassungsvorgänge zu ermöglichen, ist es sinnvoll, die Variation der Gesichter zu kennen. Sollten die Fehler bei der Graphenanpassung im Bereich der Variation liegen, so könnte man zum Finden von Korrespondenzen zwischen zwei Gesichtsbildern auch einfach die identischen Pixelpositionen annehmen. Ein geeignetes Verfahren muß also in jedem Fall besser sein. Ein Kreuzvergleich aller Knotenpositionen untereinander auf dem Datensatz FG 500 lieferte eine Variation von 7,32 Pixeln (bei einer Bildgröße von 192 mal 192 Pixel). Dabei lieferten die einzelnen Knoten Werte im Bereich von 4,72 (Knoten 33, Ohr) bis 11,88 (Knoten 9, Nasenspitze). Die Unterschiede liegen an der Posenvariation, bei der sich die Nasenspitze am meisten bewegt und Punkte, die eher an den Seiten des Gesichtes liegen, sich kaum bewegen.

## 3.3 Natürliche Gesichter der Kodak-Datenbank

Für Tests zum Gesichterfinden steht am Institut für Neuroinformatik die Kodak-Datenbank [11] zur Verfügung. In ihr liegen Photos vor, die jeweils eine unterschiedliche Zahl von Gesichtern enthalten. Im Wesentlichen handelt es sich dabei um Urlaubsbilder. Die Bilder sind sehr unterschiedlich in ihrer Auflösung und Qualität, und auch die Größe der Gesichter variiert sehr stark. Da die Gesichter

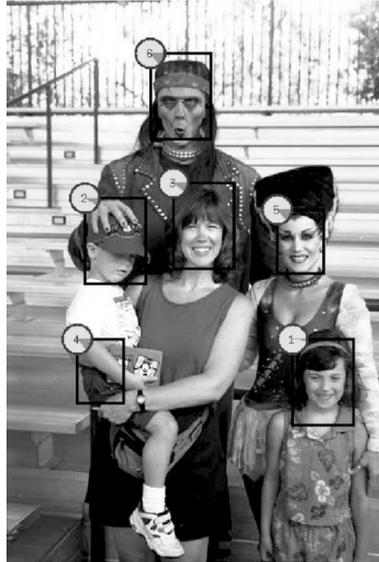


Abb. 3.6: Beispiel eines Bildes aus der Kodak-Datenbank mit vom FaceFinder durch Rechtecke markierten Gesichtshypothesen

teilweise verdeckt sind, eine starke Posenvariation aufweisen, oft auch Sonnenbrillen oder Mützen getragen werden, gilt diese Datenbank als durchaus schwierig. Das Beispielbild zeigt sehr gut die Schwierigkeiten, die bei den Bildern dieser Datenbank auftreten können. Während die Gesichter der Frauen noch deutlich zu sehen sind, ist es offensichtlich, daß es schwierig ist, das Gesicht des geschminkten Mannes zu erkennen oder besonders das des Kindes, welches zur Seite weggedreht und durch die Mütze verdeckt ist. Als Mensch ist man sich der Schwierigkeit oft nicht bewußt, da man selbst für diese Aufgabe nicht nur die Pixelwerte des Gesichtes zu Rate zieht, sondern ein immenses Kontextwissen anwendet: Man versteht die Situation, erkennt den Körper des kleinen Jungen auf dem Arm und weiß natürlich, daß sich unter der Mütze das Gesicht befinden muß. Für das Gesichterfinden auf diesen Bildern wird ein Programm mit dem Namen FaceFinder verwendet, das nach dem Verfahren der Graphenanpassung die Photos mit unterschiedlich großen Graphen absucht und bei entsprechender Ähnlichkeit einen Rahmen um die Region zeichnet, in der ein Gesicht vermutet wird. Mit einem kleinen Analyseprogramm, dem die wahren Hand-generierten Positionen aller Gesichter bekannt sind, läßt sich dann auswerten, wie erfolgreich die Art der Graphenanpassung für das Gesichterfinden war.

### 3.4 Natürliche Gesichter der FERET- und AR-Datenbank

Um die Erkennungsleistung auch auf natürlichen Bildern beurteilen zu können, benötigt man Bilder der gleichen Person in zwei Varianten, sei es, daß sie sich in der Pose unterscheiden, dem Gesichtsausdruck oder der Frisur. Im folgenden werden die ersten Bilder der Personen mit „Album“, die zweiten Bilder derselben Personen mit „Galerie“ bezeichnet. Für die Bewertung der Erkennungsleistung existiert ein Programm mit dem Namen Eagle. Es führt die Graphenanpassung auf den Album-, sowie auf den Galeriebildern durch und vergleicht die gefundenen Merkmale. Eine Person des Albums gilt als richtig erkannt, wenn die ähnlichste Person in der Galeriedatenbank dieselbe Person ist.



Abb. 3.7: Beispiel zweier Bilder aus der AR-Datenbank [12]

In dieser Arbeit kamen zwei solcher Datenbanken zum Einsatz, die FERET- [13] und die AR-Datenbank [12]. Beide enthielten verschiedene Personen in zwei Varianten, die relativ frontal in die Kamera schauten. Bei den FERET-Bildern unterschieden sich Album- und Galeriebild hauptsächlich durch eine leichte Änderung der Mimik, während bei den AR-Bildern zum Teil zwischen den Aufnahmen mehrere Tage vergingen, so daß sich zum Teil die Frisuren der Personen sehr unterschieden. Von daher kann man sagen, daß die AR-Bilder die schwierigere Aufgabe darstellen. Die Anzahl der Personen betrug 298 für FERET und 117 für AR.

## Kapitel 4

# Verfahren zur Bestimmung der Landmarkenpositionen

### 4.1 Teilung des Bündelgraphen

Ein erstes Experiment, bei dem die Genauigkeit der Landmarkenpositionierung gemessen wurde, vergleicht verschiedene Anpassungsvorgänge miteinander. Zunächst wurde die Bündelgraphenanpassung (BGM als Abkürzung für Bunchgraphmatching) untersucht, wie sie am Institut existiert und in Kapitel 2.3 beschrieben ist. Der dazu benötigte Bündelgraph entstand aus 54 der künstlichen Gesichter. Für diese Gesichter wird die Position der Landmarken aus den Daten des 3D-Modells berechnet, daher sind die Korrespondenzen der Punkte optimal. Diese Landmarkenpositionen werden als Ground-Truth-Positionen bezeichnet. An diesen Punkten werden dann durch Anwenden der Gaborfilter die FeaSts berechnet, die dann zu einem Bündelgraphen zusammengefasst werden. Danach blieben 47 Bilder übrig, die nicht im Bündelgraphen enthalten sind und auf denen das Verfahren der Bündelgraphenanpassung getestet wurde.

Anstelle der Bündelgraphenanpassung, bei der für jede Landmarke einzeln die größte Übereinstimmung mit einem FeaSt aus dem gesamten Bündel gesucht wird, kann man aber auch mit einem einzelnen Graphen eine Anpassung durchführen (matchen), der dann nur ein Gesicht repräsentiert (MGM für Modelgraphmatching, im folgenden Modellgraphenanpassung). Das ist äquivalent zu einem BGM mit nur einem Eintrag im Bündel. Auf diese Weise landet der Graph auf den Positionen, bei denen die größte Übereinstimmung mit der Person vorliegt, die dem Modellgraphen zugrunde liegt. Nun kann man das für jeden der Graphen durchführen, die bei der Bündelgraphenanpassung benutzt wurden, und man erhält so für jedes der 47 Bilder 54 Positionen, von denen man die Durchschnittsposition oder den Median der Position betrachten kann.

So kann man für folgende Verfahren die Positionen der Landmarken bestimmen und dank der Ground-Truth-Positionen den durchschnittlichen Fehler der Landmarkenpositionierung berechnen:

#### 1. MGM Durchschnitt

Es wird bei jedem Galeriebild eine Modellgraphenanpassung durchgeführt und anschließend die Graphenposition aus den beiden Durchschnittswerten der x- und y-Koordinate berechnet. Dieser Durchschnitt wird für jeden Knoten einzeln über die 54 Graphen des Bündels berechnet. Dieses Anpassungsverfahren wird für jedes der 47 Bilder durchgeführt und aus diesen 47 Bildern wird dann der Durchschnittsfehler über alle Knoten und alle Bilder berechnet.

#### 2. MGM Median

Bei diesem Anpassungsvorgang wird für jeden Knoten einzeln über die 54 Graphen des Bündels anstelle des Durchschnitts der Median der jeweiligen Koordinate gebildet. Damit erhält man wieder für jedes der 47 Bilder die Graphenpositionen, für die dann wieder der Durchschnittsfehler berechnet werden kann.

#### 3. BGM

Hier wird die Bündelgraphenanpassung verwendet, wie sie im Kapitel 2.3 dargestellt ist, und ebenfalls der Durchschnittsfehler berechnet.

Vergleich MGM vs. BGM FG(101)		
	Fehler [Pixel]	
	AbsFeaSt	AbsPhaseFeaSt
MGM Durchschnitt	2,447	1,490
MGM Median	2,170	1,076
BGM	1,547	0,974

Tabelle 4.1: Vergleich verschiedener Anpassungs-Methoden

Da für die Berechnung der Durchschnittswerte nur 47 Anpassungsvorgänge betrachtet wurden, ist die Genauigkeit und Aussagekraft der Werte natürlich fraglich. Zwei erste Ergebnisse, von denen eines zu der Entscheidung führte, die Bündelgraphenanpassung zu verbessern, kann man aber sicherlich festhalten:

1. Die Bündelgraphenanpassung ist geeigneter zum Landmarkenfinden als eine Einzelgraphenanpassung mit anschließender Median- oder Mittelwertberechnung.
2. Zur genauen Landmarkenpositionierung ist die Phaseninformation wichtig.

Die Wiederholung des Experimentes zu einem späteren Zeitpunkt mit dem Datensatz FG1000 bestätigt die Richtigkeit der hier getroffenen Entscheidung.

Vergleich MGM vs. BGM (FG1000)		
	Fehler [Pixel]	
	AbsFeaSt	AbsPhaseFeaSt
MGM Durchschnitt	3,381	2,051
MGM Median	3,237	1,426
BGM	2,342	1,352

Tabelle 4.2: Vergleich verschiedener Anpassungs-Methoden mit FG1000

Die Tendenzen dieser Experimente entsprechen genau denen des ersten Versuches. Das schlechtere Gesamtergebnis kommt durch die Posenvarianz der Bilder zustande. Hier ist das Finden der Landmarken natürlich wesentlich schwieriger als in dem sehr einfachen Fall gleicher Frontalpose. Das bessere Ergebnis der Medianbildung gegenüber der Durchschnittsbildung läßt sich dadurch erklären, daß bei der Medianbildung einzelne Graphen, die sehr falsch liegen, nicht so sehr ins Gewicht fallen wie bei der Durchschnittsbildung. Dieser positive Effekt der Medianbildung führte zu dem Verfahren der Bündelgraphenteilung. Die Untersuchung dieses Verfahrens fand mit dem Datensatz FG500 statt. Es entstand ein Programm, das den 70 Graphen großen Bündelgraphen zunächst dreiteilte, mit den so entstandenen Bündelgraphen eine klassische Bündelgraphenanpassung auf den 430 Galeriebildern durchführte und anschließend für die Knotenposition den Median der drei Positionen wählte, wobei nach Koordinaten getrennt die mittlere x- und die mittlere y-Koordinate die neue Landmarkenposition bildeten.

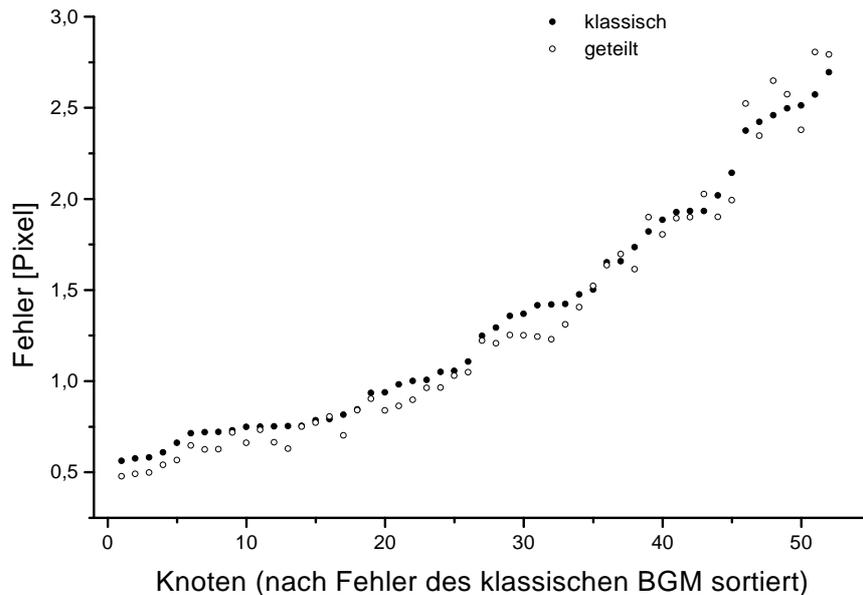


Abb. 4.1: BGM-Teilung: Fehler über Knoten

BGM klassisch vs. dreigeteilt	
Fehler [Pixel]	
BGM (klassisch)	BGM (dreigeteilt)
AbsPhaseFeaSt	
1,339	1,294

Tabelle 4.3: Vergleich Bündelgraphenteilung vs. klassisch

Hierbei zeigt sich eine leichte Verbesserung gegenüber der klassischen Variante. Wenn auch nur eine geringe Verbesserung auftritt, so ist diese dennoch aussagekräftig, da bei den 430 Galeriebildern der Durchschnittsfehler aus  $430 \cdot 52 = 22360$  Einzelmessungen gewonnen wurde, wodurch dieser Wert statistisch aussagekräftig ist. Einen Fehler anzugeben ist dabei nicht möglich, da es sich hierbei nicht um Einzelmessungen des selben Wertes handelt, sondern um einen Durchschnitt vieler unterschiedlicher Werte.

Trägt man den durchschnittlichen Fehler der Knotenpositionierung für jeden der 52 Knoten getrennt auf, also den Durchschnittswert von 430 Bildern, und sortiert man dabei die Knoten nach der Größe des Fehlers für das klassische BGM, so erhält man Diagramm 4.1.

Hier zeigt sich, daß das Verfahren der Bündelgraphenteilung bei den Landmarken die klassisch gut gefunden werden, etwas besser funktioniert, und daß es bei den Landmarken, bei denen die klassische Variante ihre Schwierigkeiten hat, in etwa die gleichen Probleme aufweist.

Trägt man für jedes der 430 Bilder die Durchschnittsfehler (Durchschnitt von 52 Knoten) des klassischen BGM gegen die Durchschnittsfehler der Bündelgraphenteilung, so erhält man Diagramm 4.2.

Die Häufung der Punkte um die Winkelhalbierende herum zeigt, wie schon Diagramm 4.1, die Vergleichbarkeit beider Verfahren. Die Punkte, die rechts davon liegen, zeigen jedoch, daß die Bilder, die für das klassische BGM schwierig sind, relativ einfach für die Methode der Bündelgraphenteilung sind. Umgekehrt zeigen die Punkte oberhalb, daß die Bündelgraphenteilung Probleme mit den Bildern hat, auf denen klassisch die Landmarken relativ gut gefunden werden. Eigentlich er-

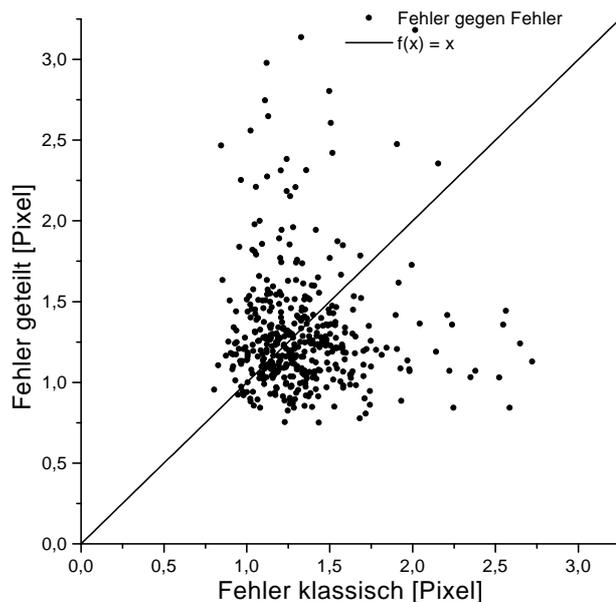


Abb. 4.2: BGM-Teilung: Fehler gegen Fehler bildweise

wartet man ein äquivalentes Verhalten der Verfahren und damit eine Verteilung der Punkte um die Winkelhalbierende herum, mit darunterliegenden Punkten für ein besseres und darüberliegenden für ein schlechteres Verfahren. Das ist aber nicht der Fall, und so zeigt sich hier ein prinzipieller Unterschied beider Verfahren.

## 4.2 Veränderte Ähnlichkeitsfunktion

Bei der Berechnung der Ähnlichkeiten bei der Bündelgraphenanpassung wird die Ähnlichkeit der Landmarke im Galeriebild zum ähnlichsten FeaSt im Bündelgraphen berechnet. Dies geschieht über eine Skalarproduktbildung der beiden FeaStvektoren, wie es in Kapitel 2.3 beschrieben ist. Bei einer Person, die im Bündelgraphen enthalten ist, ist es sicherlich das beste, die höchste Ähnlichkeit im Bündelgraphen zu suchen.

Wenn aber auf einem neuen Bild die Korrespondenzen einer Person gesucht werden, die noch nicht im Bündelgraphen gespeichert ist, so kann es sinnvoll sein, die Landmarke für die Nase dort zu platzieren, wo nicht eine möglichst hohe Übereinstimmung mit einer einzelnen Nase gefunden wurde, sondern wo eine hohe Übereinstimmung mit mindestens zwei, drei oder mehr gespeicherten Nasen auftritt. Dadurch ist es möglich, mit einer etwas allgemeineren Nasenschablone zu suchen.

Um dies zu untersuchen, wurde die Ähnlichkeitsfunktion dahingehend verändert, bei der Ähnlichkeitsberechnung, über ein Parameterfile steuerbar, nicht nur das ähnlichste FeaSt zu benutzen, sondern die ähnlichsten  $n$  FeaSts. Dabei besteht die Möglichkeit, diese Ähnlichkeiten in verschiedener Form zu gewichten. Neben der ganz naheliegenden Variante, alle  $n$  FeaSts gleich zu gewichten, wurden in dieser Arbeit noch verschiedene Varianten untersucht, die die ähnlicheren FeaSts stärker gewichten. Tabelle 4.4 erklärt die unterschiedlichen Varianten konstant, linear, exp Basis 2, exp Basis  $e$ . Bei den Funktionen läuft der Definitionsbereich von 1 in ganzen Zahlen bis zur Anzahl der Parameter. Das gerade noch betrachtete FeaSt mit der geringsten Ähnlichkeit hatte immer die Gewichtung eins, die Gewichtung des ähnlichsten FeaSts war dann von der Anzahl der betrachteten FeaSts abhängig. Für die Normierung mußte zuletzt die Gesamtähnlichkeit durch die Summe der Gewichte dividiert werden.

Koeffizienten der veränderten Ähnlichkeitsfunktion				
Koeffi- zienten	Koeffizientenfunktion			
	konstant	linear	exp 2	exp e
	$f(i) = 1$	$f(i) = i$	$f(i) = 2^{i-1}$	$f(i) = r(e^{i-1})$
1	1	1	1	1
2	1	2	2	3
3	1	3	4	7
4	1	4	8	20
5	1	5	16	55
6	1	6	32	148
7	1	7	64	403
8	1	8	128	1097
9	1	9	256	2981
10	1	10	512	8103
11	1	11	1024	22026
12	1	12	2048	59874

Tabelle 4.4: In der veränderten Ähnlichkeitsfunktion benutzte Koeffizienten

Damit ergeben sich die Formeln für die Berechnung der Gesamtähnlichkeit  $s$  aus den Einzelähnlichkeiten  $s_i$  bei  $n$  betrachteten FeaSts, wobei das ähnlichste FeaSt den größten Index  $i$  hat. Die Funktion  $r$  rundet dabei auf ganze Zahlen.

$$\begin{aligned}
 \text{konstant:} \quad s &= \frac{1}{n} \sum_{i=1}^n s_i \\
 \text{linear:} \quad s &= \frac{1}{\sum_{i=1}^n i} \sum_{i=1}^n i \cdot s_i \\
 \text{exp 2:} \quad s &= \frac{1}{\sum_{i=1}^n 2^{i-1}} \sum_{i=1}^n 2^{i-1} \cdot s_i \\
 \text{exp e:} \quad s &= \frac{1}{\sum_{i=1}^n r(e^{i-1})} \sum_{i=1}^n r(e^{i-1}) \cdot s_i
 \end{aligned}$$

Die ersten 70 Gesichter bildeten den Bündelgraphen für die Anpassungsvorgänge auf den restlichen 930 Bildern des Datensatzes FG1000. Der Fehler, gemessen in Abhängigkeit der Anzahl der betrachteten Koeffizienten und ihrer Art der Gewichtung, ist also ein Durchschnittswert über 930 Bilder mal 52 Knoten. Tabelle 4.5 und die graphische Darstellung in Diagramm 4.3 zeigen diese gemessenen Fehler.

Es zeigt sich, daß sich schon bei Betrachtung der zwei bis drei ähnlichsten FeaSts eine Verbesserung der Knotenpositionierung erreichen läßt. Die Gewichtungsarten konstant und linear erreichen ein Minimum und werden mit zunehmender Koeffizientenzahl wieder schlechter. Die beiden exponentiellen Gewichtungsarten steuern auf ein Minimum zu, dabei lag das Minimum für die die Gewichtung exp Basis 2 unter dem der Gewichtung Basis e. Es ist denkbar, mit einer anderen Basis zwischen 1 und e noch geringere Fehlerwerte zu erreichen.

Fehler der veränderten Ähnlichkeitsfunktion				
Koeffizienten	Koeffizientenfunktion			
	konstant	linear	exp 2	exp e
1	1,3515			
2	1,2991	1,2958		1,2995
3	<b>1,2900</b>	1,2838	1,2845	1,2868
4	1,2949	<b>1,2815</b>	1,2789	1,2836
5	1,3141	1,2857	1,2765	1,2828
6	1,3269	1,2933	1,2744	1,2821
7	1,3348	1,3032	<b>1,2735</b>	1,2814
8	1,3477	1,3091	1,2738	<b>1,2811</b>
9	1,3591	1,3143	1,2736	1,2812
10	1,3645	1,3190	1,2738	1,2814
11	1,3724	1,3263	1,2738	1,2814
12	1,3790	1,3385	1,2737	1,2814

Tabelle 4.5: Mittlerer Pixelfehler der Knotenposition der veränderten Ähnlichkeitsfunktion und Parametersatz 1 (siehe Anhang). Die Minima sind fett gedruckt.

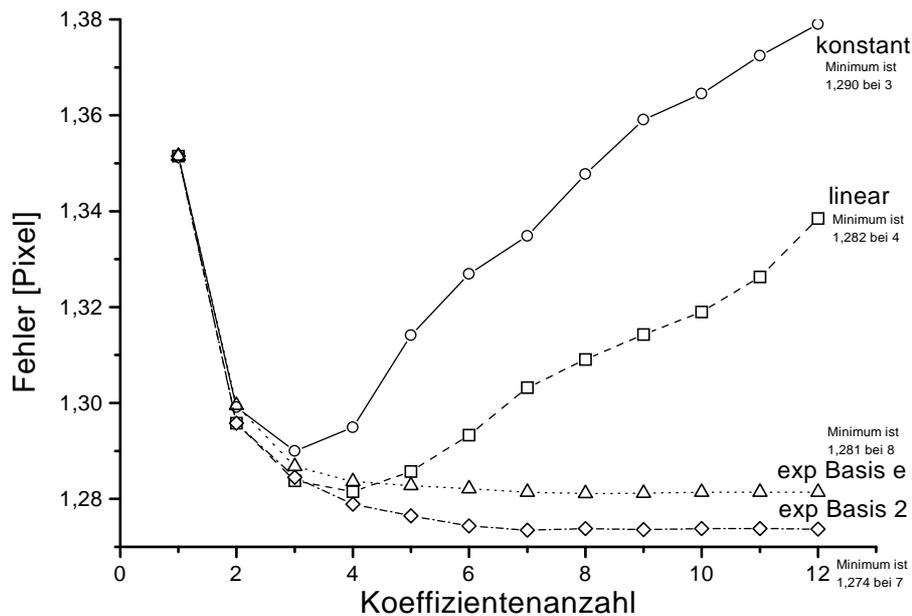


Abb. 4.3: Veränderte Ähnlichkeitsfunktion: Fehler bei Parametersatz 1 (siehe Anhang)

Die Ergebnisse, die mit der Graphen Anpassung erzielt werden, sind auch sehr stark von den Bewegungsparametern abhängig. Diese steuern zum Beispiel die Schrittweite beim ersten Global-Move, die Größe der Verzerrung beim ScanScale-Move oder die Höhe des topographischen Faktors beim Local-Move, mit dem die Verzerrung des Graphen „bestraft“ wird. So lieferte eine leicht modifizierte Parameterdatei bessere Ergebnisse.

Fehler der veränderten Ähnlichkeitsfunktion				
Koeffizienten	Koeffizientenfunktion			
	konstant	linear	exp 2	exp e
1	1,3353			
2	1,2774	1,2838		1,2878
3	<b>1,2639</b>	1,2632	1,2652	1,2717
4	1,2664	1,2555	1,2572	1,2664
5	1,2728	<b>1,2528</b>	1,2548	1,2657
6	1,2845	1,2587	1,2538	1,2646
7	1,2914	1,2635	1,2536	1,2643
8	1,2984	1,2703	1,2538	1,2645
9	1,3097	1,2726	1,2537	1,2642
10	1,3199	1,2805	1,2535	1,2642
11	1,3266	1,2861	1,2535	1,2641
12	1,3335	1,2936	<b>1,2533</b>	<b>1,2641</b>

Tabelle 4.6: Mittlerer Pixelfehler der Knotenposition der veränderten Ähnlichkeitsfunktion und Parametersatz 2 (siehe Anhang). Die Minima sind fett gedruckt.

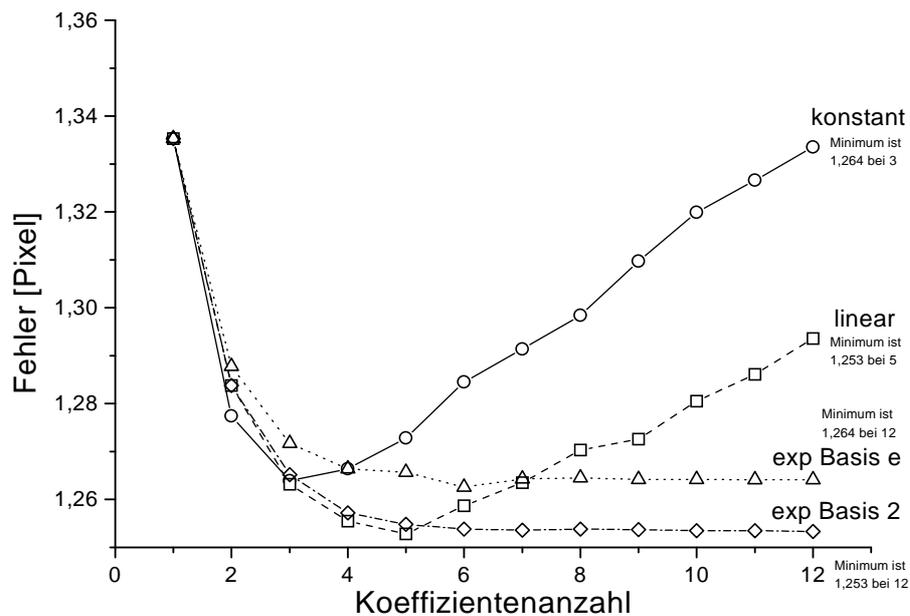


Abb. 4.4: Veränderte Ähnlichkeitsfunktion: Fehler bei Parametersatz 2 (siehe Anhang)

Die Wiederholung dieser Experimente mit dem Parametersatz 2 (siehe Anhang) zeigt, daß die Kurven sich prinzipiell nicht unterscheiden, sondern nur um die durch die anderen Parameter erreichte Verbesserung nach unten verschoben sind. Das unterstreicht noch einmal die allgemeine Gültigkeit der hier gefundenen Ergebnisse.

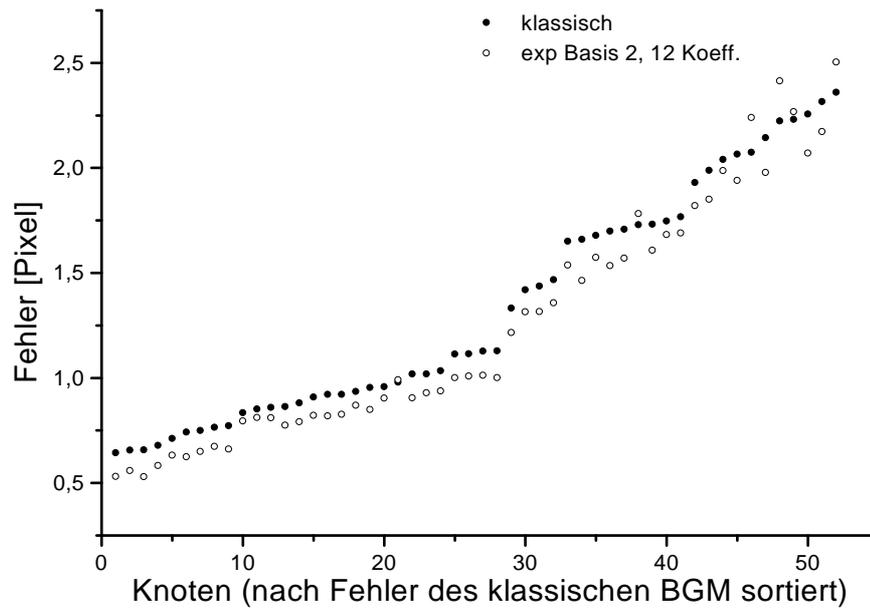


Abb. 4.5: Veränderte Ähnlichkeitsfunktion: Fehler über Knoten (Parametersatz 2)

Trägt man wieder die Fehler beider Verfahren gegen die Knotennummer, sortiert nach der Größe des Fehlers der klassischen Variante, auf, so erhält man ein ähnliches Ergebnis wie schon bei der Bündelgraphenteilung. Die Knoten, die für das klassische BGM einfach zu positionieren sind, werden durch die veränderte Ähnlichkeitsfunktion etwas besser positioniert. Die schwierig zu positionierenden Landmarken sind bei modifizierter Ähnlichkeitsfunktion genauso problematisch.

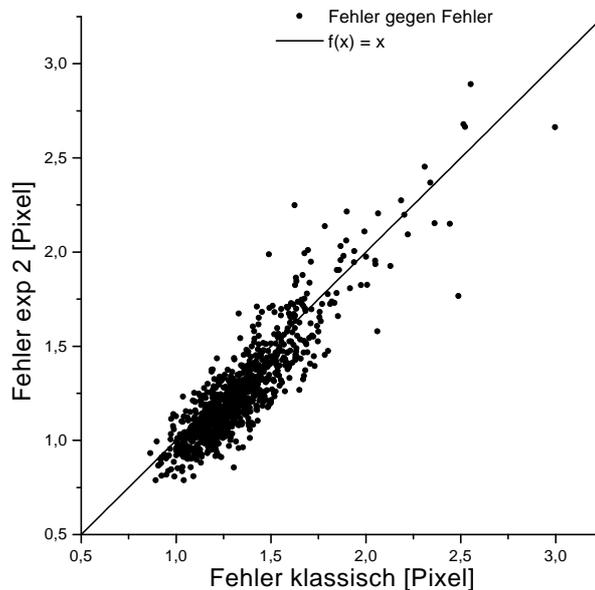


Abb. 4.6: Veränderte Ähnlichkeitsfunktion: Fehler gegen Fehler bildweise (Parametersatz 2)

In dem Diagramm Fehler gegen Fehler, in dem wieder die Durchschnittsfehler der veränderten Ähnlichkeitsfunktion für jeden Graphen gegen die Fehler des klassischen BGM aufgetragen sind, ergibt sich ein völlig anderes Ergebnis als bei der Bündelgraphenteilung. Hier sind nämlich nun die Graphen mit klassisch schlecht zu findenden Korrespondenzen auch die, welche mit der Methode der veränderten Ähnlichkeitsfunktion die größten Fehler aufweisen. Das sieht man daran, daß die Verteilung der Punkte in etwa parallel zur Winkelhalbierenden verläuft.

Den etwas geringeren Fehler in der Positionierung kann man nun auch an dieser Graphik sehen, da die Mehrzahl der Punkte unterhalb der Geraden liegt.

### 4.3 PCA-Matching

Wie schon in Kapitel 2.4 erklärt, lassen sich mit der Hauptkomponentenanalyse viele hochdimensionale Merkmalsvektoren optimal darstellen. Da die Merkmalsvektoren der verschiedenen Landmarken alle etwas anderes kodieren, führt man die Hauptkomponentenanalyse für jede Landmarke separat durch. Will man für die Suche von Landmarken auf neuen Bildern eine Basis von 200 Gesichtern zu grunde legen, so entsteht für jeden Knoten ein Bündel von 200 40-dimensionalen komplexwertigen Vektoren, die zunächst in 80-dimensionale Vektoren, zusammengesetzt aus Real- und Imaginärteil, umgewandelt werden können. Die Hauptkomponentenanalyse liefert damit für jeden Knoten den Schwerpunktsvektor, 80 Hauptkomponenten und 80 Varianzen. Diese Informationen lassen sich in einem sogenannten PCA-Jet zusammenfassen, dem Äquivalent zu einem FeaSt:

$$\text{für Knoten } i: \quad \langle \vec{F}^i \rangle \otimes \{\vec{P}_0^i \dots \vec{P}_N^i\} \otimes \{\sigma_0^i \dots \sigma_N^i\}$$

Genau wie bei der Bündelgraphenanpassung werden auch beim PCA-Matching die Jets zu einem Graphen mit denselben Verbindungen zwischen den Landmarken zusammengefasst. Um den PCA-Graphen wie einen Bündelgraphen zur Graphenanpassung benutzen zu können, benötigt man eine Ähnlichkeitsfunktion, die zwischen den FeaSts eines jeden Bildpunktes und dem PCA-Jet (also der Repräsentation der Landmarke im PCA-Graph) eine Ähnlichkeit berechnet und so den zur Landmarke korrespondierenden Punkt im Bild findet.

Dafür werden die Projektionen des Bild-FeaSts auf die Hauptkomponenten berechnet. Von einer gaußförmigen Verteilung der Merkmalsvektoren ausgehend, erhält man so für jedes Merkmal die Wahrscheinlichkeit, daß es in der Verteilung vorkommt. Man fasst diese Wahrscheinlichkeiten als Einzelähnlichkeiten auf, gewichtet sie mit der gaußschen Normalverteilung und summiert die 80 Einzelähnlichkeiten zu einer Gesamtähnlichkeit.

Eine Betrachtung der Histogramme rechtfertigt diese Annahme für fast alle Hauptkomponenten. Nur die ersten zwei bis drei Hauptkomponenten zeigen eine Verteilung mit einem etwas breiterem Plateau oder sogar einem Ansatz zweier Maxima. Zieht man zur Berechnung der Ähnlichkeit die Hauptkomponenten  $\alpha$  bis  $\beta$  heran, so lautet die Ähnlichkeitsfunktion für den Gaborjet  $j$  damit

$$S_1(\vec{j}, i) = \frac{1}{\beta - \alpha} \sum_{m=\alpha}^{\beta} e^{-\frac{(\langle \vec{j} - \langle \vec{F} \rangle | \vec{P}_m^i \rangle)^2}{2\sigma_m^i}} \quad \text{mit } 0 \leq \alpha < \beta \leq N \quad (4.1)$$

Es wurden auch Experimente mit einer Ähnlichkeitsfunktion durchgeführt, bei der die Einzelähnlichkeiten nicht summiert, sondern multipliziert wurden. Die Begründung hierfür ist, daß es sich bei den einzelnen Ähnlichkeiten um Wahrscheinlichkeiten handelt, die für die Berechnung der Gesamtwahrscheinlichkeit multipliziert werden müssen. Diese Funktion lieferte aber nur sehr schlechte Ergebnisse, auf deren Darstellung ganz verzichtet wird.

Um eine ausreichende Datenbasis für die Hauptkomponentenanalyse zu haben, ist es sinnvoll, mehr als die bisher verwendeten 70 Gesichter des Bündelgraphen zu verwenden. Wenn, wie es bei 70 Gesichtern der Fall ist, die Zahl der Vektoren kleiner ist als die Dimensionalität, dann liefert die Hauptkomponentenanalyse auch nur so viele linear unabhängige Hauptkomponenten wie Vektoren existieren. Schon daher ist klar, daß die Stärken des PCA-Matchings erst bei einer großen Datenbasis zum Tragen kommen.

Zeitvergleich BGM vs. PCA-Matching				
Messung Nr.	t [s]			
	70 FeaSts		500 FeaSts	
	BGM	PCA	BGM	PCA
1	52,46	42,31	241,40	48,13
2	52,65	42,18	241,98	48,78
3	52,34	42,59	241,85	48,17
4	52,39	42,13	241,44	48,71
5	52,57	42,34	241,82	48,13
6	52,57	42,75	241,76	48,11
7	52,47	42,45	241,71	48,59
8	52,49	42,28	241,81	48,03
9	52,34	42,57	241,21	48,90
10	52,52	42,08	241,58	48,48
Durchschnitt	52,48	42,37	241,66	48,60

Tabelle 4.7: Zeitvergleich eines einzelnen Anpassungsvorganges zwischen BGM und PCA-Matching auf einer ®Intel™ XEON CPU 1,80 GHz

Ein Vergleich der Zeit, die für die Graphenanpassung benötigt wird, zeigt, daß es durchaus gerechtfertigt ist, das klassische BGM, das auf einer Basis von 70 Gesichtern arbeitet, mit dem PCA-Matching zu vergleichen, das als Basis 500 Gesichter hat. Die Zeit, die für das PCA-Matching benötigt wird, ist, wenn die Hauptkomponentenanalyse einmalig durchgeführt wurde, ab einer Basis von 80 Gesichtern nicht mehr von der Anzahl der Gesichter abhängig. Die Zeit, die das BGM benötigt, ist durch die Vergleiche mit jedem der Gesichter im Bündelgraphen in etwa proportional der Zahl der Gesichter im Bündel, weil die Zahl der Gesichter gleich der Zahl der FeaSts ist, die an jedem Knoten hängen.

Im ersten Experiment wurden auf dem Datensatz FG1000 die Positionierungsfehler gemessen. Tabelle 4.8 zeigt die Ergebnisse des PCA-Matchings für eine unterschiedliche Zahl in der Ähnlichkeitsfunktion benutzter Hauptkomponenten (Entsprechend Formel 4.1 wurde  $\beta$  variiert bei  $\alpha = 0$ ). Darin sind die Durchschnittsfehler für AbsPhase-FeaSts mit den erwähnten Werten und für Abs-FeaSt gegenübergestellt. Bei AbsFeaSts erhält man nur maximal 40 Hauptkomponenten, entsprechend der Dimensionalität der Vektoren. Diese Fehler wurden für eine 70 und eine 500 Gesichter enthaltende Basis gemessen.

Hierbei zeigt sich, daß beim PCA-Matching für AbsFeaSt keine Verbesserung erzielt werden kann. Bezieht man hingegen die Phaseninformation mit ein, so sinkt der Fehler der Knotenposition von einem sehr hohen Wert bei geringer Zahl betrachteter Hauptkomponenten mit steigender Komponentenzahl deutlich unter den Vergleichswert des klassischen BGM. Bei allen Anpassungsvorgängen mit der PCA ist deshalb nur von AbsPhaseFeaSts ausgegangen worden. Um die Hauptkomponentenanalyse durchführen zu können, wurden die komplexen Werte in Real- und Imaginärteil umgewandelt, da die Phase eine zyklische Variable ist. Diese so entstandenen FeaSts werden im weiteren ImReFeaSts genannt.

Fehler BGM vs. PCA-Matching				
	AbsPhaseFeaSt		AbsFeaSt	
#PCs	70 Bsp.	500 Bsp.	70 Bsp.	500 Bsp.
10	18,172	18,508	4,679	4,689
20	2,558	2,076	3,150	3,188
30	1,430	1,219	2,528	2,603
40	1,256	1,001	2,213	2,329
50	1,163	0,862	-	-
60	1,104	0,750	-	-
69	1,084	-	-	-
70	-	0,678	-	-
80	-	0,625	-	-
<b>BGM</b>	1,335	1,192	2,240	2,144

Tabelle 4.8: Vergleich des Positionierungsfehler [Pixel] zwischen Bündelgraphenanpassung und PCA-Matching (additiv)

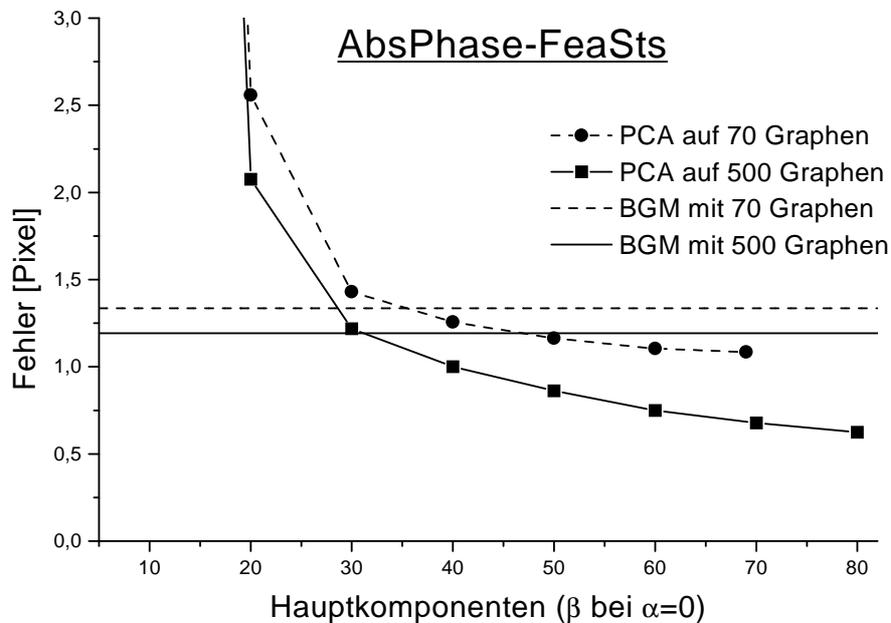


Abb. 4.7: PCA-Matching: Fehler über Anzahl PCs

Vergleicht man auf Grund der vorherigen Zeitüberlegungen den BGM-Wert von 70 Beispielen mit dem PCA-Matching-Wert von 500 Beispielen, so ergibt sich eine Fehlerreduktion von 1,335 Pixeln auf 0,625 Pixel, was einer Fehlerreduktion von 53,2% entspricht.

Als kurze Anmerkung zur technischen Umsetzung der PCA sei noch erwähnt, daß es numerische Probleme geben kann, die zu negativen Varianzen bei den letzten Hauptkomponenten führen, wenn die Zahl der Beispiele gleich der Zahl der berechneten Hauptkomponenten ist. Daher wurden in Tabelle 4.8 bei 70 Beispielen 69 Hauptkomponenten anstelle der eigentlich 70 verwendet.

Diagramm 4.7 stellt das Verhalten graphisch dar. Als Abkürzung für Hauptkomponente wird dort PC (Principal Component) verwendet.

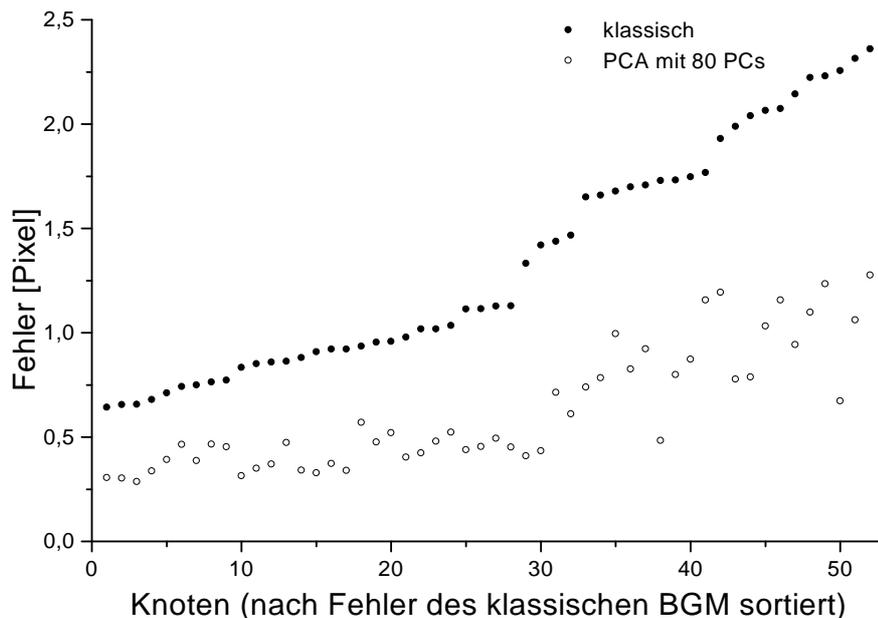


Abb. 4.8: PCA-Matching: Fehler über Knoten

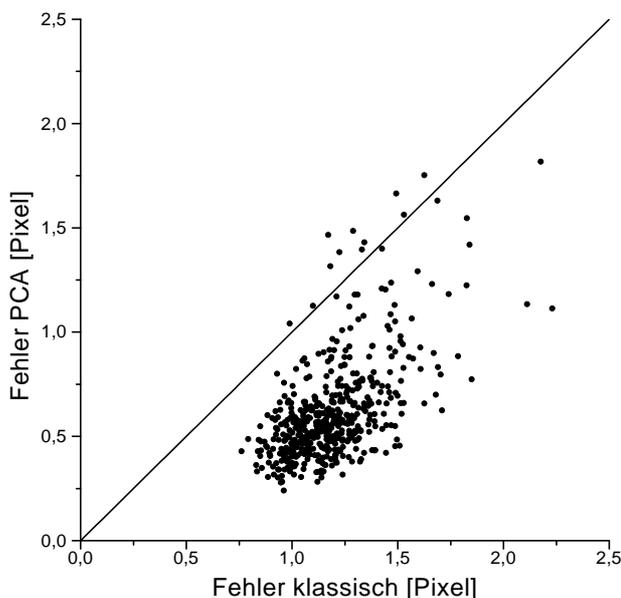


Abb. 4.9: PCA-Matching: Fehler gegen Fehler bildweise

Vergleicht man nun auch das PCA-Matching (500 Gesichter, 80 Hauptkomponenten) mit dem BGM (70 Gesichter), indem man die Fehler knotenweise, nach Fehler des BGM sortiert, aufträgt, so erhält man Diagramm 4.8. Hier zeigt sich deutlich, daß das PCA-Matching in der Lage ist, den Fehler jeder Landmarke im Durchschnitt um die Hälfte zu reduzieren. Sowohl die Bündelgraphenteilung als auch die veränderte Ähnlichkeitsfunktion brachte bei den schwierigeren Landmarken keine Verbesserung. Diese Eigenschaft der PCA, besonders aber die Halbierung des Durchschnitts-

fehlers, macht das PCA-Matching zu dem am besten geeigneten Verfahren für das Finden von Korrespondenzen, was auch im Diagramm 4.9, Fehler gegen Fehler, deutlich wird. Fast alle Punkte liegen unterhalb der Winkelhalbierenden, was zeigt, daß es nur wenige Bilder gibt, auf denen die Korrespondenzen schlechter gefunden werden als bei der klassischen Graphenanpassung, auf fast allen anderen Bildern hingegen werden sie besser gefunden.

Ein zweites Experiment betrachtet nicht die ersten  $n$  Hauptkomponenten, sondern die letzten. Wenn man daran denkt, daß die Verteilung der ersten zwei bis drei Hauptkomponenten nicht genau gaußförmig aussieht, so läßt sich durch das Weglassen der ersten Hauptkomponenten noch eine Verbesserung erwarten. Dies entspricht in Formel 4.1 einer Variation von  $\alpha$  bei  $\beta = 80$ . Die Ergebnisse zeigt Tabelle 4.9 und graphisch dargestellt Diagramm 4.10.

Fehler bei Auslassen der ersten Hauptkomponenten							
PCs ( $\alpha$ bei $\beta = 80$ )	0	1	2	3	4	5	6
Fehler	0,625	0,624	0,626	0,621	0,619	0,623	0,622
PCs ( $\alpha$ bei $\beta = 80$ )	10	15	20	25	30		
Fehler	0,624	0,625	0,633	0,644	0,670		

Tabelle 4.9: Pixelfehler des PCA-Matching bei Auslassen der ersten  $n$  Hauptkomponenten

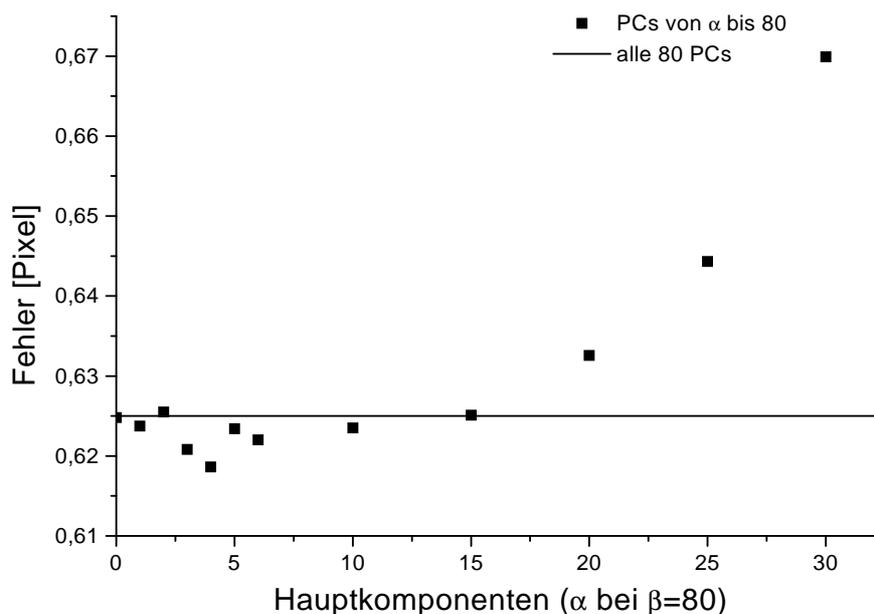


Abb. 4.10: PCA-Matching: Fehler über Anzahl weggelassener PCs

Hier zeigt sich die Vermutung der geringfügigen Verbesserung bestätigt. Inwieweit dieses Ergebnis sich auf natürliche Bilder übertragen läßt, und besonders wie viele Hauptkomponenten dann ausgelassen werden sollten, ist allerdings noch zu untersuchen. Erstaunlich ist jedoch auf jeden Fall, daß bei einem Auslassen der ersten 15 Hauptkomponenten eine gleich gute Landmarkenpositionierung erreicht wird, wie mit allen Hauptkomponenten, besonders wenn man sich in Erinnerung ruft, daß die ersten Hauptkomponenten die größte Varianz der Daten bilden. Ein Erklärungsansatz ist die These, daß in den ersten Hauptkomponenten möglicherweise die Posenvariation und ähnliches kodiert ist.

## 4.4 Skala und Orientierung, Normierung

Auf der Suche nach den optimalen Anpassungsparametern kann man sich natürlich fragen, ob die bisher verwendete Zahl der Skalen und Orientierungen der Gaborfilter für eine bestmögliche Positionierung geeignet ist. Bisher wurde immer eine Zahl von 5 Skalen und 8 Orientierungen verwendet. Diagramm 4.4 enthält die Positionierungsfehler für das BGM mit unterschiedlichen Skalen und Orientierungen. Dies entspricht einer Variation von  $\nu_{\max}$  und  $\mu_{\max}$  in den Formeln 2.12-2.15.

Fehler verschiedener Skalen & Orientierungen			
Skalen ( $\nu_{\max}$ )	Orientierungen ( $\mu_{\max}$ )		
	6	8	12
4	1,6049	1,5820	1,5551
5	1,3810	1,3394	1,3368
6	1,4037	1,3665	1,3676

Tabelle 4.10: Pixelfehler des klassischen BGM bei verschiedenen Skalen und Orientierungen

Es ließ sich keine signifikante Verbesserung feststellen. Die Standardeinstellung ist also durchaus gut gewählt.

### Normierung

In der Regel sind die FeaSt-Vektoren normiert, sie haben also alle die Länge eins und liegen damit auf einer Hyperkugel. Ist eine einzelne Komponente des nicht normierten Vektors sehr groß, so „zieht“ diese Komponente die anderen bei der Normierung auf sehr kleine Werte. Um diesen Effekt zu verhindern, ist es möglich, die FeaSt-Vektoren für jede Skala einzeln zu normieren. Damit wird der Gesamtvektor ebenfalls auf eins normiert, allerdings beeinflussen die unterschiedlichen Skalen sich bei der Normierung nicht gegenseitig.

Beim PCA-Matching wurden die AbsPhaseFeaSts normiert, in ImReFeaSts umgewandelt und dann die Hauptkomponentenanalyse durchgeführt.

Fehler klassisch vs. skalenweise normiert		
Anpassungsverfahren	klassisch	skalenweise
klassisches BGM	1,335	1,316
veränderte Ähnlichkeitsfkt.	1,253	1,179
PCA-Matching	0,625	1,567

Tabelle 4.11: Vergleich klassischer vs. skalenweiser Normierung bei verschiedenen Anpassungsverfahren. Fehler [Pixel]

Es zeigt sich, daß die skalenweise Normierung zu einer leichten Verbesserung führt, allerdings beim PCA-Matching völlig einbricht. Bei der Normierung der FeaSt Vektoren liegen die ursprünglich 80-dimensionalen Vektoren auf einer Hyperkugel, also einem gekrümmten Unterraum der Dimension 79. Daher liegt es nahe, einen Versuch mit nicht normierten Vektoren durchzuführen.

Dabei ergeben sich aber viel höhere Fehlerwerte. Während die Landmarken auf den meisten Bildern gleich gut gefunden werden, wird der Graph auf manchen Gesichtern völlig falsch plazierte. Die Normierung der FeaSts ist also auch für das PCA-Matching notwendig.

## Kapitel 5

# Gesichterfinden und Erkennen

Inhalt dieses Kapitels ist es, die für künstlich erzeugte Gesichter erzielten Verbesserungen auf reale Bilder zu übertragen. Dazu wird die Leistung des Gesichterfindens und die Erkennungsleistung der Verfahren ermittelt.

### 5.1 Leistung des Gesichtsfinders

Eine Anwendung, mit der sich die Auswirkungen der in dieser Arbeit erprobten Verfahren sehr leicht messen lassen, ist das Gesichterfinden. Mit dem FaceFinder [14] steht am Institut ein Programm zur Verfügung, das auf der schon erwähnten Kodak-Datenbank, in der hauptsächlich Urlaubsphotos mit Gesichtern unterschiedlicher Zahl, Größe und Pose enthalten sind, diese Gesichter sucht und anschließend die Hypothesen bewertet. Dazu führt es für unterschiedliche Skalierungen des Bildes eine Bündelgraphenanpassung durch und zeichnet um die Bereiche, in denen die Ähnlichkeit beim BGM einen Schwellwert überschritten hat, einen Rahmen. Befindet sich in dem Bereich, in dem das Programm ein Gesicht vermutet, tatsächlich eines, so wird das als Erfolg gewertet. Fehler sind Bereiche mit Gesichtern, die das Programm nicht findet, aber auch Bereiche, in denen FaceFinder ein Gesicht vermutet, die aber gar keines enthalten.

So kann man durch eine entsprechend niedrig gesetzte Entscheidungsschwelle die Erkennungsrate, also die Anzahl der gefundenen Gesichter geteilt durch die der vorhandenen, erhöhen. Sehr viel aussagekräftiger hingegen ist die Erkennungsrate bei Equal-Error, bei der die Schwelle so gesetzt wird, daß genauso viele Gesichter nicht gefunden wie Nichtgesichter gefunden werden.

Das Programm wurde so modifiziert, daß die Anpassungsvorgänge zum einen mit veränderter Ähnlichkeitsfunktion und zum anderen mit dem PCA-Matching erfolgten.

#### Veränderte Ähnlichkeitsfunktion

Tabelle 5.1 zeigt den Vergleich zwischen klassischem BGM und veränderter Ähnlichkeitsfunktion. Hierbei hat sich in mehreren Versuchen die Verwendung der konstanten Gewichtung mit fünf Koeffizienten als am besten herausgestellt.

Betrachtet man die Erfolgssraten, so kann man schon von einer durchgängigen Verbesserung sprechen, wenn sie auch zum Beispiel bei den unverdeckten Gesichtern bis  $50^\circ$  nur von 83% auf 87% steigt. Es darf allerdings bezweifelt werden, daß diese Verbesserung Folge einer genaueren Positionierung ist. So wird für das Gesichterfinden in den meisten Fällen die Phaseninformation nicht berücksichtigt, die aber für eine möglichst exakte Landmarkenpositionierung wichtig ist. Wenn man an die Funktionsweise der veränderten Ähnlichkeitsfunktion denkt, nämlich die Forderung von Übereinstimmung mit mehr als einem Gesicht, dann kann man sagen, daß die veränderte Ähnlichkeitsfunktion hier die Sicherheit erhöht, mit einer Hypothese richtig zu liegen. Begründen kann man das auch durch das Ergebnis, bei dem die Erfolgssraten bei Equal-Error Rate stärker angestiegen sind als die bei voreingestellter Schwelle.

Erfolgsraten des FaceFinder		
Anpassungsverfahren	Standard	konstant 5
Erfolgsrate bei voreingestellter Schwelle		
Alle Gesichter	87,94%	89,04%
bis 50°	90,49%	91,48%
Unverdeckt bis 50°	92,80%	93,18%
Erfolgsrate bei Equal-Error		
Alle Gesichter	59,10%	62,81%
bis 50°	81,52%	83,87%
Unverdeckt bis 50°	83,27%	86,59%

Tabelle 5.1: Vergleich der Erfolgsraten des FaceFinder auf der Kodak-Datenbank zwischen klassischem BGM und veränderter Ähnlichkeitsfunktion

Erfolgsraten des FaceFinder		
Anpassungsverfahren	Standard	PCA
Erfolgsrate bei voreingestellter Schwelle		
All faces	90,35%	89,91%
bis 50°	91,80%	91,15%
Unverdeckt bis 50°	92,80%	92,05%
Erfolgsrate bei Equal-Error Rate		
All faces	48,06%	47,80%
bis 50°	71,79%	71,58%
Unverdeckt bis 50°	74,29%	74,07%

Tabelle 5.2: Vergleich der Erfolgsraten des FaceFinder auf der Kodak-Datenbank zwischen klassischem und PCA-Matching

### PCA-Matching

Da das PCA-Matching nur dann zu den entsprechenden Verbesserungen in der Landmarkenpositionierung führt, wenn für die Statistik genügend Gesichtsbeispiele vorhanden sind, reichen die vom FaceFinder verwendeten Bündelgraphen in ihrer Größe nicht aus. Hierfür wurde ein neuer Bündelgraph verwendet, der aus 516 Gesichtern realer Personen besteht (BG 516). Versuche, mit Bündelgraphen aus künstlichen Personen Gesichter in natürlichen Bildern zu finden, waren nicht erfolgreich. Durch die Verwendung eines anderen Bündelgraphen ergeben sich erwartungsgemäß andere Erfolgsraten, die mit denen in Tabelle 5.1 nicht mehr verglichen werden können.

Als Ergebnis kann man hiermit die Vermutung bestätigt sehen, daß eine genauere Positionierung der Landmarken für die Leistung eines Gesichtsfinders nicht von Vorteil sind. Allerdings muß man zugeben, daß es sich bei den Modellgraphen, auf denen die PCA berechnet wurde, um Graphen handelt, deren Landmarken durch automatische Graphenanpassung gefunden wurden, bei der keine Kontrolle durch den Benutzer vorgenommen wurde. Dies ist eine mögliche Fehlerquelle. Außerdem konnte beobachtet werden, daß beim Gesichterfinden Gesichter kleinerer Skala besser gefunden wurden.

## 5.2 Erkennungsleistung

Erkennungsraten verschiedener Anpassungsverfahren					
Anpassungsverfahren	Pos.Fehler [Pixel]	Erkennungsrate		rel. Erkenn.-Fehler	
		1000	100	1000	100
BGM (70 Graphen)	1,335	50,7%	69%	7,31%	2,82%
BGM (70 Graphen, ver. Ähnlichkeitsfkt.)	1,264	50,5%	69%	7,68%	2,82%
BGM (500 Graphen)	1,192	50,9%	69%	6,95%	2,82%
PCA (von 500 Graphen, 80 PCs)	0,625	52,6%	70%	3,84%	1,41%
ohne Anpassung, Groundtruth-Posit.	0	54,7%	71%	0%	0%

Tabelle 5.3: Vergleich der Erkennungsraten auf FaceGen-Gesichter bei verschiedenen Anpassungsverfahren

Eine ganz wichtige Aufgabe ist die Gesichtserkennung, bei der überprüft wird, ob es sich bei zwei Bildern um dieselbe Person handelt. Da hierbei direkt die korrespondierenden Punkte miteinander verglichen werden, besteht die starke Vermutung, daß die Erkennungsleistung steigt, wenn man die Korrespondenzen besser findet. Für den Test der Erkennungsleistung existiert ein Programm namens Eagle. Auf zwei Galerien, die die gleichen Personen in jeweils veränderter Pose (und bei natürlichen Bildern auch anderem Gesichtsausdruck) enthalten, wird eine Bündelgraphenanpassung mit einem passenden Bündelgraphen durchgeführt und die Jets an den gefundenen Stellen extrahiert. Anschließend findet ein Kreuzvergleich der Jets heraus, ob das erste Bild in der ersten Galerie auch als ähnlichstes zu dem entsprechenden Bild in der zweiten Galerie gefunden wird und so weiter. Als Wert für die Erkennungsleistung gilt der Prozentsatz der richtig identifizierten Personen. Um sicherstellen zu können, daß auch das PCA-Matching beim Eagle die verbesserte Knotenposition erreichte, wurde der gleiche (lediglich in der Skalierung leicht variierte) Bündelgraph aus 500 künstlichen Gesichtern des Datensatzes FG1000 benutzt. Als Testbilder dienten 1000 neu erzeugte künstliche Gesichter, die in je zwei unterschiedlichen, zufälligen Posen vorlagen (FG2000).

Tabelle 5.3 zeigt die Erkennungsraten für 1000 und für 100 verschiedene Gesichter. Der relative Erkennungsfehler bezieht sich auf den Kreuzvergleich der FeaSts an den Groundtruth-Positionen. Dabei wird davon ausgegangen, die maximal mögliche Erkennungsleistung bei dieser wahren Position zu erreichen, und für sie wird der Fehler auf null gesetzt. Damit ergibt sich der relative Erkennungsfehler als Anzahl der falsch erkannten Personen geteilt durch die Zahl der Personen, die mit perfekten Korrespondenzen erkannt werden.

Als erstes Ergebnis muß man feststellen, daß die Erkennungsleistung für 1000 Gesichter insgesamt sehr gering ist, viel geringer als bei Versuchen mit natürlichen Gesichtern. Das liegt daran, daß sich die von FaceGen erzeugten Gesichter zwar schon unterscheiden, daß sich bei Gesichtern dieser großen Zahl aber doch viele Gesichter sehr ähnlich sehen. So geht dann auch die Erkennungsleistung stark nach oben, wenn man versucht, nur 100 Personen zu unterscheiden.

Betrachtet man den Vergleich klassisches BGM gegen PCA-Matching, so gilt:

**Eine Halbierung der Fehler bei der Landmarkenpositionierung führte auf den künstlichen Gesichtern auch zu einer Halbierung der Fehler der Erkennungsleistung.**

Allerdings bleibt natürlich noch zu überprüfen, ob sich das PCA-Matching auch für natürliche Gesichter eignet. Geht man davon aus, daß der gefundene Zusammenhang im Prinzip auch für natürliche Bilder richtig ist, so hat man nun die Möglichkeit, die Genauigkeit der Landmarkenpositionierung indirekt durch die Erkennungsraten zu ermitteln.

So wurden mit dem Eagle Tests auf zwei Datenbanken, der FERET-Datenbank und der AR-Datenbank, durchgeführt. Beide lieferten im Prinzip dieselben Ergebnisse. Die veränderte Ähnlichkeitsfunktion führte hier nur auf den AR-Daten zu einer Verbesserung, aber nur bei Verwendung

Erkennungsraten auf natürlichen Gesichtern				
Anpassungsverfahren	BG 38		BG 516	
	Fehler	Fehlerrate	Fehler	Fehlerrate
BGM klassisch	30	10,1%	20	6,7%
Ähnlichk.Fkt. exp 2, 5 Koeff.	30	10,1%	28	9,4%
PCA-Matching 80 PCs	-	-	19	6,4%
PCA-Matching PCs 5-80	-	-	19	6,4%

Tabelle 5.4: Vergleich der Erkennungsraten auf 298 natürlichen Gesichtern der FERET-Datenbank bei verschiedenen Anpassungsverfahren mit Zahl der Fehlerkennungen (Fehler) und den Fehlerraten

Erkennungsraten auf natürlichen Gesichtern				
Anpassungsverfahren	BG 38		BG 516	
	Fehler	Fehlerrate	Fehler	Fehlerrate
BGM klassisch	46	39,3%	33	28,2%
Ähnlichk.Fkt. exp 2, 5 Koeff.	38	32,5%	34	29,1%
PCA-Matching 80 PCs	-	-	31	26,5%
PCA-Matching PCs 5-80	-	-	31	26,5%

Tabelle 5.5: Vergleich der Erkennungsraten auf 117 natürlichen Gesichtern der AR-Datenbank bei verschiedenen Anpassungsverfahren mit Zahl der Fehlerkennungen (Fehler) und den Fehlerraten

des kleinen, 38 Gesichter großen Graphen (BG 38).

Das PCA-Matching lieferte hingegen eine konstant höhere Erkennungsleistung, sogar etwas höher als die Bündelgraphenanpassung mit dem großen Graphen. Da man die Ergebnisse aber mit BGM-Werten und kleinem Bündelgraphen vergleichen muß, ergibt sich hier eine deutliche Verringerung des Fehlers von 10% auf 6% bzw. von 39% auf 27%.

Hierbei wurden bis jetzt alle Hauptkomponenten verwendet. Da auf den künstlichen Bildern der Fehler durch Weglassen der ersten Hauptkomponenten noch ein wenig reduziert werden konnte, wurde das PCA-Matching mit unterschiedlich vielen ausgelassenen Hauptkomponenten auf den AR-Bildern wiederholt.

Erkennungsraten auf natürlichen Gesichtern								
PCs ( $\alpha$ bei $\beta = 80$ )	0	2	4	6	8	10	12	14
Fehler	31	31	31	32	32	29	28	25
PCs ( $\alpha$ bei $\beta = 80$ )	16	18	20	22	24	26	28	30
Fehler	25	26	26	26	27	26	27	27

Tabelle 5.6: Vergleich der Erkennungsraten auf 117 natürlichen Gesichtern der AR-Datenbank

Tabelle 5.6 zeigt, daß es bei natürlichen Gesichtern noch wichtiger ist, die ersten Hauptkomponenten auszulassen. Auch die Zahl der nicht zu betrachtenden Komponenten ist höher als bei künstlichen Gesichtern. Während dort das beste Ergebnis bei einem Startwert von 4 erzielt wurde, liegt der optimale Wert nun bei 13 bis 17. Hierbei wurde der Fehler von ursprünglich 39% nicht mehr nur auf 27%, sondern bis auf 21% verringert.

Um die Ergebnisse besser vergleichen zu können, sind immer die gleichen Anpassungsparameter verwendet worden. Mit speziell für das jeweilige Verfahren angepaßten Parametern sind sicher noch weitere Optimierungen in jeder Variante möglich.

## Kapitel 6

# Diskussion und Ausblick

Ausgehend von der geschilderten Problemstellung bestand das Ziel dieser Arbeit darin, die Korrespondenzen zwischen Gesichtsbildern möglichst genau zu finden.

In einem ersten Schritt wurden dazu die bisherigen Arbeiten vorgestellt, die sich mit Gesichtsfinden und -erkennen befassen und daraufhin bewertet, wie gut in ihnen die Korrespondenzen gefunden werden.

Danach sind die Eigenschaften der künstlichen Gesichter, auf denen die Fehler der Landmarkenpositionierung gemessen wurden, ausführlich beschrieben worden. Auch die natürlichen Gesichter, die für das Testen der Leistung des Gesichtsfindens und der Erkennungsleistung benötigt wurden, sind mit Beispielen beschrieben worden.

Daraufhin sind die Verfahren beschrieben worden, die auf eine Verbesserung der Genauigkeit der Landmarkenpositionierung untersucht wurden, und die Ergebnisse wurden dargestellt und erläutert. Es wurden leichte Verbesserungen der Genauigkeit durch das Verfahren der Bündelgraphenteilung und das Verfahren der veränderten Ähnlichkeitsfunktion erzielt. Besonders aber haben die Ergebnisse beim PCA-Matching gezeigt, daß dieses Verfahren in dem Finden von Korrespondenzen allen anderen Verfahren überlegen ist. Hierbei konnte auf den künstlichen Gesichtern eine Fehlerreduktion von 53,2% erreicht werden.

Bei Tests zum Gesichtsfinden konnte keine Verbesserung erzielt werden, da hierbei eine genaue Positionierung einzelner Landmarken nicht vorteilhaft ist. Versuche zur Erkennungsleistung zeigten jedoch, daß auf künstlichen Gesichtern die Erkennungsleistung in dem Maße steigt, in dem eine genauere Positionierung erreicht wurde. Dieses Ergebnis rechtfertigt die in dieser Arbeit vorgenommenen Bemühungen, die Landmarken immer genauer zu plazieren.

Überträgt man diesen Zusammenhang auf natürliche Gesichter, bei denen Experimente eine Fehlerreduktion von 37% (FERET) und 46% (AR) erreichten, so kann man festhalten, daß die Korrespondenzen auf künstlichen wie auf natürlichen Gesichtern durch das Verfahren des PCA-Matchings deutlich besser gefunden werden können.

In Zukunft sind einige Möglichkeiten denkbar, die Genauigkeit beim PCA-Matching noch zu erhöhen. Neben Parametereinstellungen wie die optimale Zahl zu verwendender PCs ist auch denkbar, mit einer Erhöhung der Zahl der Skalen und Orientierungen eine Verbesserung zu erreichen, da beim PCA-Matching die Zahl der Hauptkomponenten gleich dem Produkt aus Skalen und Orientierungen ist.

Erste Versuche, das PCA-Matching nicht nur für das Finden der Korrespondenzen einzusetzen, sondern auch für einen Vergleich zweier Landmarken, in dem nicht die FeaSt-Vektoren, sondern ihre Projektionen auf die Hauptkomponenten verglichen wurden, zeigten eine Verbesserung in der Erkennungsleistung.

Auch besteht über die Lage einzelner Merkmale in der Verteilung die Möglichkeit, jeder Landmarke eine gewisse Außergewöhnlichkeit zuzuordnen und damit eine Signifikanz für die Erkennung zu berechnen. So könnten besondere Merkmale auch in besonderem Maße für die Erkennung berücksichtigt werden, und die Entscheidung, welche das sind, träfe dann automatisch der Computer.

# Anhang A

## Move-Parameter

### A.1 Parameterdatei 1

MatchParam:

number of moves = 4

move 0: ScanGlobalMove

delta in x-direction = 4  
delta in y-direction = 4  
flag for processing whole image = True  
similarity function: AbsPhaseSimFct  
Coefficients of Max Considered =  
number of coefficients = 12  
coefficients:  
12 11 10 9 8 7 6 5 4 3 2 1

move 1: ScanScaleXyMove

connect x-y = false  
parameter of scale:  
number of scales in x-direction = 8  
minimum range of scale in x-direction = 0.70  
maximum range of scale in x-direction = 1.5  
  
number of scales in y-direction = 8  
minimum range of scale in y-direction = 0.70  
maximum range of scale in y-direction = 1.5

numberOfPositions = 17

positionArr:

0[0, 0]  
1[4, 0]  
2[4,2]  
3[4,4]  
4[2,4]  
5[0,4]  
6[-2,4]  
7[-4,4]  
8[-4,2]  
9[-4,0]  
10[-4,-2]  
11[-4,-4]  
12[-2,-4]  
13[0,-4]  
14[2,-4]

```

15[4,-4]
16[4,-2]
similarity function: AbsPhaseSimFct
Coefficients of Max Considered =
  number of coefficients = 12
  coefficients:
    12 11 10 9 8 7 6 5 4 3 2 1

move 2: ScanGlobalMove
  delta in x-direction           = 1
  delta in y-direction           = 1
  flag for processing whole image = False
  distance for search region in x-direction = 6
  distance for search region in y-direction = 6
  similarity function: AbsPhaseSimFct
  Coefficients of Max Considered =
    number of coefficients = 12
    coefficients:
      12 11 10 9 8 7 6 5 4 3 2 1

move 3: ScanLocalMove
  delta in x-direction           = 1
  delta in y-direction           = 1
  distance for search region in x-direction = 8
  distance for search region in y-direction = 8
  number of iterations           = 4
  use current edges              = false
  influence of topography        = 0.004
  similarity function: AbsPhaseSimFct
  Coefficients of Max Considered =
    number of coefficients = 12
    coefficients:
      12 11 10 9 8 7 6 5 4 3 2 1

```

## A.2 Parameterdatei 2

```

MatchParam:
number of moves = 5

move 0: ScanGlobalMove
  delta in x-direction           = 4
  delta in y-direction           = 4
  flag for processing whole image = True
  similarity function: AbsPhaseSimFct
  Coefficients of Max Considered =
    number of coefficients = 12
    coefficients:
      12 11 10 9 8 7 6 5 4 3 2 1

move 1: ScanGlobalMove
  delta in x-direction           = 2
  delta in y-direction           = 2
  flag for processing whole image = False
  distance for search region in x-direction = 8
  distance for search region in y-direction = 8
  similarity function: AbsPhaseSimFct
  Coefficients of Max Considered =
    number of coefficients = 12

```

```

coefficients:
12 11 10 9 8 7 6 5 4 3 2 1

move 2: ScanScaleXyMove
  connect x-y                               = True
  parameter of scale:
  number of scales                          = 3
  minimum range of scale                    = 0.82
  maximum range of scale                    = 1.18
  numberOfPositions = 5
  positionArr:
  0[3, 3]
  1[-3, 3]
  2[3, -3]
  3[-3, -3]
  4[0,0]
  similarity function: AbsPhaseSimFct
  Coefficients of Max Considered =
  number of coefficients = 12
  coefficients:
  12 11 10 9 8 7 6 5 4 3 2 1

move 3: ScanGlobalMove
  delta in x-direction                      = 1
  delta in y-direction                      = 1
  flag for processing whole image           = False
  distance for search region in x-direction = 4
  distance for search region in y-direction = 4
  similarity function: AbsPhaseSimFct
  Coefficients of Max Considered =
  number of coefficients = 12
  coefficients:
  12 11 10 9 8 7 6 5 4 3 2 1

move 4: ScanLocalMove
  delta in x-direction                      = 1
  delta in y-direction                      = 1
  distance for search region in x-direction = 5
  distance for search region in y-direction = 5
  number of iterations                      = 5
  use current edges                         = true
  influence of topography                   = 0.005
  similarity function: AbsPhaseSimFct
  Coefficients of Max Considered =
  number of coefficients = 12
  coefficients:
  12 11 10 9 8 7 6 5 4 3 2 1

```

# Abbildungsverzeichnis

2.1	Ein formales Neuron . . . . .	6
2.2	Gesichterfinden mit neuronalen Netzen nach Kanade. Die Abbildung entstammt [4]. . . . .	8
2.3	Rotationsinvariantes Gesichterfinden mit neuronalen Netzen nach Kanade. Die Abbildung entstammt [5]. . . . .	8
2.4	Ein- und zweidimensionale Gaborfunktion . . . . .	9
2.5	Antwortverhalten der Nervenzellen und der Gaborfunktion [6] . . . . .	9
2.6	Gaußglocken für 5 Skalen und 8 Orientierungen im Frequenzraum. Die Achsen sind die Frequenzachsen $\omega_1$ und $\omega_2$ . . . . .	11
2.7	Darstellung des Bündelgraphen . . . . .	12
3.1	Landmarkenpositionen bei einem FaceGen-Gesicht . . . . .	18
3.2	Graphenstruktur . . . . .	18
3.3	Beispielbilder des Datensatzes FG101 . . . . .	19
3.4	Beispielbilder des Datensatzes FG500 . . . . .	19
3.5	Beispielbilder des Datensatzes FG1000 . . . . .	20
3.6	Beispiel eines Bildes aus der Kodak-Datenbank . . . . .	21
3.7	Beispiel zweier Bilder aus der AR-Datenbank [12] . . . . .	22
4.1	BGM-Teilung: Fehler über Knoten . . . . .	25
4.2	BGM-Teilung: Fehler gegen Fehler bildweise . . . . .	26
4.3	Veränderte Ähnlichkeitsfunktion: Fehler bei Parametersatz 1 (siehe Anhang) . . . . .	28
4.4	Veränderte Ähnlichkeitsfunktion: Fehler bei Parametersatz 2 (siehe Anhang) . . . . .	29
4.5	Veränderte Ähnlichkeitsfunktion: Fehler über Knoten (Parametersatz 2) . . . . .	30
4.6	Veränderte Ähnlichkeitsfunktion: Fehler gegen Fehler bildweise (Parametersatz 2) . . . . .	30
4.7	PCA-Matching: Fehler über Anzahl PCs . . . . .	33
4.8	PCA-Matching: Fehler über Knoten . . . . .	34
4.9	PCA-Matching: Fehler gegen Fehler bildweise . . . . .	34
4.10	PCA-Matching: Fehler über Anzahl weggelassener PCs . . . . .	35

# Tabellenverzeichnis

4.1	Vergleich verschiedener Anpassungs-Methoden . . . . .	24
4.2	Vergleich verschiedener Anpassungs-Methoden mit FG1000 . . . . .	24
4.3	Vergleich Bündelgraphenteilung vs. klassisch . . . . .	25
4.4	In der veränderten Ähnlichkeitsfunktion benutzte Koeffizienten . . . . .	27
4.5	Mittlerer Pixelfehler der Knotenposition der veränderten Ähnlichkeitsfunktion und Parametersatz 1 . . . . .	28
4.6	Mittlerer Pixelfehler der Knotenposition der veränderten Ähnlichkeitsfunktion und Parametersatz 2 . . . . .	29
4.7	Zeitvergleich eines einzelnen Anpassungsvorganges zwischen BGM und PCA-Matching auf einer ®Intel™ XEON CPU 1,80 GHz . . . . .	32
4.8	Vergleich des Positionierungsfehler [Pixel] zwischen Bündelgraphenanpassung und PCA-Matching (additiv) . . . . .	33
4.9	Pixelfehler des PCA-Matching bei Auslassen der ersten n Hauptkomponenten . . . . .	35
4.10	Pixelfehler des klassischen BGM bei verschiedenen Skalen und Orientierungen . . . . .	36
4.11	Vergleich klassischer vs. skalenweiser Normierung bei verschiedenen Anpassungsverfahren. Fehler [Pixel] . . . . .	36
5.1	Vergleich der Erfolgsraten des FaceFinder auf der Kodak-Datenbank zwischen klassischem BGM und veränderter Ähnlichkeitsfunktion . . . . .	38
5.2	Vergleich der Erfolgsraten des FaceFinder auf der Kodak-Datenbank zwischen klassischem und PCA-Matching . . . . .	38
5.3	Vergleich der Erkennungsraten auf FaceGen-Gesichtern bei verschiedenen Anpassungsverfahren . . . . .	39
5.4	Vergleich der Erkennungsraten auf 298 natürlichen Gesichtern der FERET-Datenbank bei verschiedenen Anpassungsverfahren . . . . .	40
5.5	Vergleich der Erkennungsraten auf 117 natürlichen Gesichtern der AR-Datenbank bei verschiedenen Anpassungsverfahren . . . . .	40
5.6	Vergleich der Erkennungsraten auf 117 natürlichen Gesichtern der AR-Datenbank . . . . .	40

# Literaturverzeichnis

- [1] HELGE RITTER, THOMAS MARTINEZ UND KLAUS SCHULTEN: *Neuronale Netze*. Addison-Wesley, 1994.
- [2] McCULLOCH, W. S. und W. PITTS: *A logical calculus of the ideas immanent in nervous activity*. Bull Math Biophys, 5:115-133, 1943.
- [3] ROSENBLATT, F.: *The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain*. Psych Rev, 65:386-408, 1958.
- [4] H. A. ROWLEY, S. BALUJA und T. KANADE: *Neural network-based face detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):23-38, January 1998.
- [5] H. A. ROWLEY, S. BALUJA und T. KANADE: *Rotation invariant neural network-based face detection*. Technical Report 97-201, Computer Science Department, Carnegie Mellon University, January 1998.
- [6] JONES, JUDSON P. und LARRY A. PALMER: *An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex*. Journal of Neurophysiology, 58(6):1233 – 1258, December 1987.
- [7] FIELD, D. J.: *Relations between the statistics of natural images and the response properties of cortical cells*. Optical Society of America, 4(12):2379-2394, 1987.
- [8] LADES, MARTIN, JAN C. VORBRÜGGEN, JOACHIM BUHMANN, J. LANGE, CHRISTOPH VON DER MALS-BURG, ROLF P. WÜRTZ und WOLFGANG KONEN: *Distortion Invariant Object Recognition in the Dynamic Link Architecture*. IEEE Transactions on Computers, 42:300–311, 1993.
- [9] WISKOTT, LAURENZ, JEAN-MARC FELLOUS, NORBERT KRÜGER und CHRISTOPH VON DER MALS-BURG: *Face Recognition by Elastic Bunch Graph Matching*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):775–779, 1997.
- [10] TURK, M. und A. PENTLAND: *Eigenfaces for recognition*. Journal of Cognitive Neuroscience, 3(1), pp 71-86, 1991.
- [11] A. LOUI, C. JUDICE und S. LIU: *An image database for benchmarking of automatic face detection and recognition algorithms*. Proc. IEEE International Conference on Image Processing, 1998.
- [12] MARTINEZ, A.M. und R. BENAVENTE: *The AR Face Database*. CVC Technical Report 24, June 1998.
- [13] P.J. PHILLIPS, P.J. RAUSS und S.Z. DER: *FERET (Face Recognition Technology) Recognition Algorithms*. Proc. Fifth Automatic Target Recognizer System and Technology Symp., 1996.
- [14] WIEGHARDT, JAN und HARTMUT S. LOOS: *Finding Faces in Cluttered Still Images with Few Examples*. In: DORFFNER, GEORG, HORST BISCHOF und KURT HORNIK (Herausgeber): *Artificial Neural Networks – ICANN 2001*, Band 2130 der Reihe *Lecture Notes in Computer Science*, Seiten 1026–1033, Vienna, Austria, 2001. Springer Verlag.

# Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die zum Gelingen dieser Arbeit beigetragen haben. Ein besonderer Dank gilt:

- **Prof. Dr. von der Malsburg** für die Vergabe des interessanten Themas und für die freundliche Aufnahme an seinen Lehrstuhl.
- **Prof. Dr. Wieck** für die Beurteilung dieser Arbeit als Zweitgutachter.
- **Dr. Würtz und Alexander Heinrichs** für die hervorragende Betreuung, Einarbeitung und Zusammenarbeit in dieser Zeit.
- **Frau Schwalm** für ihre Hilfsbereitschaft in allen organisatorischen Fragen
- **Michael Neef und Arno Berg** für den reibungslosen Betrieb des Rechnernetzes und Hilfe bei Problemen.
- **allen Mitarbeitern des Lehrstuhls** für die gute Zusammenarbeit und nette Atmosphäre, sowie für ihre Hilfe bei Fragen.
- **Christof, Daniel, Egmont, Hanno, Jens, Marcus, Markus und Steffi** für ihre Freundschaft und Hilfe während des Studiums.

und nicht zuletzt natürlich **all meinen Freunden** und besonders **meinen Eltern**, die mir mit ihrer Unterstützung dieses Studium erst ermöglicht haben.