

Elastic Bunch Graph Matching für die Objekterkennung

Hauptseminar Medizinische Bildverarbeitung im Sommersemester 2003

von David Rybach

Inhalt

1	Einleitung	2
2	Grundlagen	2
2.1	Jets	2
2.2	Graphen und Bunch Graphs	3
2.3	Vergleich von Bunch Graphs	4
3	Elastic Bunch Graph Matching	5
3.1	Der Matching-Algorithmus	5
3.2	Training	5
3.3	Erkennung	6
4	Anwendungen	6
4.1	Handgesten-Erkennung	6
4.1.1	Problembeschreibung	6
4.1.2	Verfahren	7
4.1.3	Ergebnisse	8
4.2	Gesichtserkennung	8
4.2.1	Problembeschreibung	8
4.2.2	Verfahren	8
4.2.3	Ergebnisse	10
5	Diskussion	10
6	Fazit	11
	Literaturverzeichnis	13

Zusammenfassung

Elastic Bunch Graph Matching ist eine Methode zur Objekterkennung, die auf lokalen und geometrischen Merkmalen basiert. Objekte werden mit objekt-adaptierten Graphen und Gabor-Wavelets modelliert. Aus den Trainingsdaten werden Graphen erstellt, die zu klassenspezifischen Graphen, „Bunch Graphs“, zusammengefasst werden und alle Variationen im Trainingsmaterial abdecken. Elastic Bunch Graph Matching ist invariant zu Helligkeitsänderungen des Bildes, Größen- und Positionsänderungen des Objekts und zum Bildhintergrund. Es werden Anwendungen in der Handgesten- und in der Gesichtserkennung vorgestellt. Diese Seminararbeit basiert auf den Arbeiten von Wiskott et al. [1, 2] und Triesch, von der Malsburg [3, 4].

Keywords: Object Recognition, Graph Matching, Computer Vision, Gabor Wavelets, Face Recognition, Hand Posture Recognition

1 Einleitung

Basis dieser Seminararbeit sind die Arbeiten von Wiskott et al. zur Gesichtserkennung mit Elastic Bunch Graph Matching in [1, 2] und von Triesch und von der Malsburg zur Handgesten-Erkennung in [3, 4].

Elastic Bunch Graph Matching ist eine Methode zur Objekterkennung auf der Grundlage von Bildern, die auf lokalen und geometrischen Merkmalen basiert. Das Verfahren ist theoretisch für die Erkennung aller Objekttypen, die sich hinreichend durch geometrische und textuelle Merkmale unterscheiden, anwendbar. Die biologische Motivation des Elastic Bunch Graph Matching sind Erkenntnisse über das menschliche Sehen und die menschliche Wahrnehmung. Ziel ist es, die Erkennung ohne spezielle Anforderungen an das Bild durchführen zu können und dadurch eine große Anzahl von Variationen des Objekts abzudecken und korrekt zu erkennen. Dazu gehören zum Beispiel Invarianz gegenüber der Helligkeit des Bildes, Größe und Position des Objekts sowie dem Bildhintergrund. Trotzdem soll das Training des Erkenners nicht aufwändig sein, so dass einfach neue Klassen von Objekten berücksichtigt werden können.

Objekte werden für das Elastic Bunch Graph Matching mit beschrifteten Graphen modelliert. Die Graphen sind objekt-adaptiert, also von der Form des Objekts abgeleitet. Besondere lokale Merkmale des Objekts werden als Knotenbeschriftung verwendet. Eine Klasse von Objekten wird ebenfalls durch einen Graphen, einen *Bunch Graph*, repräsentiert. Für die Erkennung eines Objekts, das heißt die Zuordnung des Bildes zu einer Klasse, wird ein Graph auf Basis des Bildes erstellt, der dem Graph der Klasse möglichst ähnlich ist. Die Erstellung und Bewertung eines Graphen wird mit Elastic Bunch Graph Matching realisiert.

Elastic Bunch Graph Matching wird erfolgreich bei der Objekterkennung, der Gesichtserkennung und -detektion und bei der Analyse von komplexen Szenen verwendet. In dieser Arbeit werden die Erkennung von Handgesten und die Gesichtserkennung als Beispiele für die Anwendung von Elastic Bunch Graph Matching gezeigt.

Im Abschnitt 2 werden die Grundlagen, die für das Verstehen der Methode notwendig sind, erklärt. Diese werden dann in Abschnitt 3 verwendet, in dem der Algorithmus für das Matching, das Training für den Erkenner und die Erkennung selbst vorgestellt werden. Abschnitt 4 demonstriert den praktischen Einsatz des Elastic Bunch Graph Matching an zwei Anwendungen. Die Ergebnisse dieser Anwendungen und die grundsätzliche Methode werden in Abschnitt 5 diskutiert. Im letzten Abschnitt werden die Ergebnisse zusammengefasst, sowie mögliche Erweiterungen und die Einordnung in den medizinischen Kontext vorgestellt.

2 Grundlagen

2.1 Jets

Lokale Merkmale eines Bildes werden mit *Gabor Wavelets* extrahiert. Das Gabor Wavelet ist eine ebene Welle, die von einer Gauß-Fensterfunktion begrenzt wird (siehe Abbildung 1). Das Mother-Wavelet wird als

$$\psi_{\mathbf{k}}(\mathbf{x}) = \frac{\mathbf{k}^2}{\sigma^2} \exp\left(-\frac{\mathbf{k}^2 \mathbf{x}^2}{2\sigma^2}\right) \left[\exp(i\mathbf{k}\mathbf{x}) - \exp\left(-\frac{\sigma^2}{2}\right) \right] \quad (1)$$

definiert. Über den Parameter \mathbf{k} werden Wavelets für verschiedene Orientierungen und Frequenzen erzeugt:

$$\mathbf{k}_{\nu\mu} = \begin{pmatrix} k_{\nu} \cos \phi_{\mu} \\ k_{\nu} \sin \phi_{\mu} \end{pmatrix} \quad (2)$$

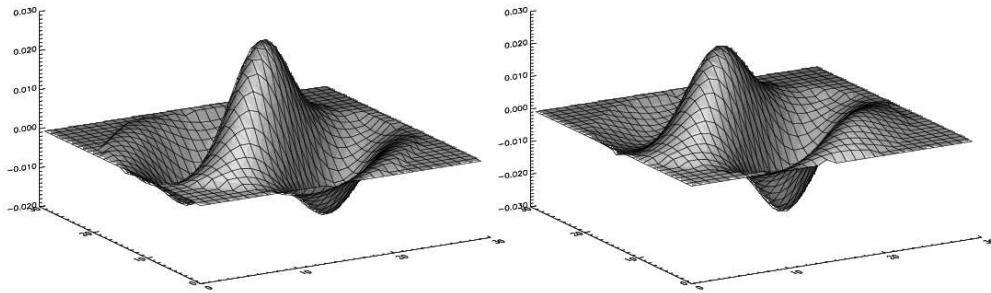


Abbildung 1: Gaborfilter. Links ist der Realteil dargestellt, rechts der Imaginärteil (aus [3])

wobei v der Index für die Frequenz, k_v die dazugehörige Frequenz, μ der Index für die Orientierung und ϕ_μ der entsprechende Winkel ist. Mit σ bzw. $\frac{\sigma}{|\mathbf{k}|}$ wird die Breite der Gaußfunktion gesteuert.

Die Verwendung von Gabor Wavelets ist unter anderem biologisch motiviert. Sie sind ein Modell für das menschliche Sehen, da Gaborfilter ähnlich reagieren wie die Rezeptorfelder des visuellen Cortex (Hirnrinde) [5]. Außerdem liefern Gabor Wavelets in vielen Anwendung im Bereich der Bildverarbeitung und -erkennung gute Ergebnisse.

Gabor Wavelets sind gut für die Merkmalsextraktion geeignet, denn sie sind invariant zu konstanten Grauwertverschiebungen (Helligkeitsänderungen) im Bild (die Wavelets sind DC-free, [1]). Außerdem sind sie, durch die geringe räumliche Ausdehnung, robust gegen Translation und Verzerrung des Bildes. Ein Nachteil der Gabor Wavelets ist die Abhängigkeit vom Hintergrund des Objekts.

Die komplexen Antworten des Filters für ein Pixel $\mathbf{x} = (x, y)$ werden zusammengefasst zu einem *Jet* $\mathcal{J} = \{J_j\}$, wobei die $J_j = a_j \exp(i\phi_j)$ die komplexwertigen Filterantworten in Polar-Darstellung sind. Wenn beispielsweise 4 Frequenzen und 5 Orientierungen untersucht werden, enthält der Jet 20 komplexe Werte. Die Amplitude $a_j(\mathbf{x})$ ändert sich nur langsam mit der Position, wogegen sich die Phase $\phi_j(\mathbf{x})$ teilweise sehr abrupt ändert.

Der Jet kann zu einem *Compound Jet* erweitert werden, wie es in Abschnitt 4.1.2 beschrieben wird.

2.2 Graphen und Bunch Graphs

Um ein Objekt, das auf einem Bild dargestellt ist, zu modellieren, wird ein Graph – *Image Graph* genannt – benutzt. Die Knoten des Graphen werden auf besonderen Punkten (*fiducial Points*) platziert, bei der Erkennung von Gesichtern zum Beispiel auf den Pupillen, den Mundwinkeln und so weiter. Da die Knoten von einem Menschen positioniert werden, ist die Bewertung eines Punkts als fiducial Point subjektiv. Jeder Knoten wird mit einem Jet beschriftet, also mit den lokalen Merkmalen des Punktes, auf dem der Knoten liegt. Die Knoten werden mit Kanten verbunden, welche mit dem Abstand der beiden Knoten beschriftet ist. Ein Image Graph \mathcal{G} für ein Objekt besteht aus N Knoten, die mit E Kanten verbunden sind. Die Knoten werden an Positionen \mathbf{x}_n , $n = 1, \dots, N$ platziert und mit Jets J_n beschriftet. Die Beschriftungen der Kanten sind zweidimensionale Distanzvektoren $\Delta \mathbf{x}_e = \mathbf{x}_n - \mathbf{x}_{n'}$ für eine Kante e , die die Knoten n und n' verbindet. Wie in Abbildung 2 zu sehen ist, ist der Graph nicht vollständig. Es werden nur benachbarte Knoten verbunden, bei denen eine Distanzmessung sinnvoll ist. Außerdem werden so redundante Distanzmessungen vermieden. Der Graph ist *objekt-adaptiert*, weil seine geometrische Struktur von der des Objekts abgeleitet ist.

Um eine Klasse von Objekten zu repräsentieren, werden mehrere Image Graphs zu einem *Bunch Graph* zusammengefasst. Die Image Graphs, die für einen Bunch Graph verwendet werden, müssen alle die gleiche Struktur besitzen. Das heißt, dass die Knoten der Graphen auf gleichen fiducial Points sitzen müssen und die Anordnung der Kanten sich nicht unterscheiden darf. Der Bunch Graph \mathcal{B} wird dann aus M Image Graphs \mathcal{G}^{Bm} konstruiert, indem die Jets der Image Graphs für jeden Knoten zu einer Menge von Jets, genannt *Bunch*, zusammengefasst werden. Der Bunch Graph hat die gleiche Struktur wie die einzelnen Graphen. Die Knoten werden mit Bunches J_n^{Bm} beschriftet, die Kanten mit der durchschnittlichen Distanz $\Delta \mathbf{x}_e^B = \sum_m \Delta \mathbf{x}_e^{Bm} / M$ der Kanten der Image Graphs. Bei der in Abschnitt 4.1 vorgestellten Anwendung in der Handgestenerkennung wird zum Beispiel jede Geste als ein Bunch Graph repräsentiert, der aus Bildern von unterschiedlichen Händen, die alle die gleiche Geste zeigen, erzeugt wird.

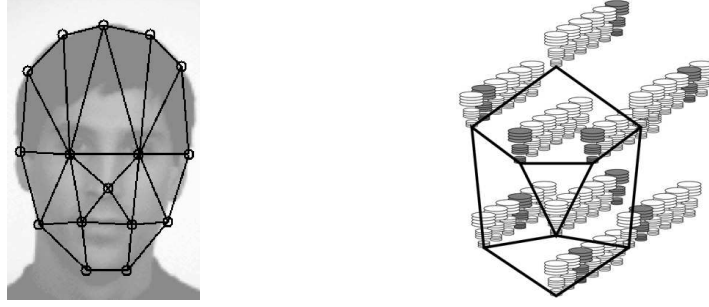


Abbildung 2: Links: Image Graph für die Erkennung des Gesichts auf dem Bild. Rechts: Abstrakte Darstellung eines Bunch Graph (beide aus [1]).

2.3 Vergleich von Bunch Graphs

Für die Objekterkennung müssen Image Graphs verglichen werden bzw. muss ein Image Graph mit einem Bunch Graph verglichen werden. Dazu werden Vergleichsfunktionen eingeführt, die eine Metrik für die Distanz zwischen den Graphen definieren. Die Ähnlichkeit wird sowohl für die Jets als auch für die geometrische Struktur bestimmt. Für Jets gibt es zwei verschiedene Vergleichsfunktionen: Die Funktion \mathcal{S}_a ignoriert die Phase, bei \mathcal{S}_ϕ wird sie berücksichtigt.

$$\mathcal{S}_a(\mathcal{J}, \mathcal{J}') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j a'^2}} \quad (3)$$

ist das normierte Skalarprodukt der absoluten Anteile der beiden Jets \mathcal{J} und \mathcal{J}' , häufig auch als normalisierte Kreuz-Korrelation bezeichnet. Durch die Invarianz zur Norm der Jets ist die Vergleichsfunktion robust gegenüber unterschiedlichen Beleuchtungsverhältnissen auf den Bildern [6].

Die phasensensitive Vergleichsfunktion kann definiert werden als

$$\mathcal{S}_\phi(\mathcal{J}, \mathcal{J}') = \frac{1}{2} \left(1 + \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j)}{\sqrt{\sum_j a_j^2 \sum_j a'^2}} \right) \quad (4)$$

In [1] wird \mathcal{S}_ϕ anders definiert: dort wird noch ein Displacement Vector eingeführt. Dieses Verfahren wird in Abschnitt 4.2 beschrieben. Die Phase hat den Vorteil, dass sie sich sehr schnell mit dem Ort ändert. Darum lässt sich die phasensensitive Vergleichsfunktion gut für die Lokalisierung eines Jets benutzen. Außerdem lassen sich damit Muster unterscheiden, die den gleichen absoluten Anteil haben.

Beim Vergleich eines Image Graphs mit einem Bunch Graph wird sowohl die Distanz zwischen den zusammengehörigen Jets als auch die geometrische Ähnlichkeit berücksichtigt. Für die Jets wird im Bunch Graph der am besten zum Jet im Image Graph passende Jet ausgewählt. Dieser wird als *local Expert* für diesen Knoten bezeichnet. In Abbildung 2 sind die local Experts durch eine graue Schattierung gekennzeichnet. Für einen Bunch Graph \mathcal{B} mit M Jets in jedem Bunch und einen Image Graph \mathcal{G}^I mit Knoten $n = 1, \dots, N$ und Kanten $e = 1, \dots, E$ ist die Vergleichsfunktion definiert als

$$\mathcal{S}_B = \frac{1}{N} \sum_n \max_m \left\{ \mathcal{S}_\phi \left(\mathcal{J}_n^I, \mathcal{J}_n^{Bm} \right) \right\} - \frac{\lambda}{E} \sum_e \frac{(\Delta \mathbf{x}_e^I - \Delta \mathbf{x}_e^B)^2}{(\Delta \mathbf{x}_e^B)^2} \quad (5)$$

λ ist die Gewichtung der geometrischen Ähnlichkeit. \mathcal{J}_n bezeichnet den Jet am Knoten n und $\Delta \mathbf{x}_e$ ist die Beschriftung der Kante e , bzw. deren Distanzvektor. Für alle Knoten n im Image Graph wird der ähnlichste Knoten im Bunch Graph gesucht und der mittlere Ähnlichkeitswert aller Knoten berechnet. Für die räumliche Struktur wird der mittlere quadratische Abstand gemessen, dessen Einfluss auf die gesamte Ähnlichkeit des Graphen mit λ gesteuert werden kann. Je nachdem, in welchem Zusammenhang die Funktion genutzt wird, kann auch die absolute Vergleichsfunktion \mathcal{S}_a benutzt werden. Darauf wird im nächsten Abschnitt näher eingegangen.

3 Elastic Bunch Graph Matching

3.1 Der Matching-Algorithmus

Nachdem ein Maß für die Ähnlichkeit eines Image Graphs mit einem Bunch Graph bekannt ist, kann nun ein Bild mit den Bunch Graphs in der Trainingsdatenbank verglichen und so ein Image Graph aus dem Bild erzeugt werden. Die Methoden, die dafür angewendet werden, unterscheiden sich in einigen Punkten. Deshalb wird an dieser Stelle nur das grundsätzliche Vorgehen beschrieben. In Abschnitt 4 werden dann die in den Anwendungen verwendeten Methoden genauer beschrieben.

Ziel des Matchings ist es, für ein gegebenes Bild einen Image Graph zu berechnen. Das heißt, dass die Knoten möglichst genau positioniert werden und der Graph möglichst wenig deformiert wird. Dazu werden nacheinander alle Bunch Graphs mit dem Bild verglichen und S_B maximiert. Für jeden Bunch Graph wird folgendes Verfahren durchlaufen:

1. Grobe Positionierung: Der Graph wird auf einem groben Raster über das Bild bewegt. In diesem Schritt darf keine Verformung des Graphen auftreten, deswegen wird S_B mit $\lambda \rightarrow \infty$ verwendet. Bei der groben Positionierung wird S_a , statt der phasensensitiven Funktion S_ϕ , als Vergleichsfunktion verwendet. Die großen Schwankungen der Phase sind für diesen Schritt zu empfindlich.
2. Anpassung der Größe und genauere Positionierung: Der Graph wird in seiner Gesamtheit vergrößert und verkleinert. Da auch hier keine Verformung auftritt, weil alle Kanten im gleichen Verhältnis verlängert bzw. verkürzt werden, bleibt $\lambda \rightarrow \infty$. Der Graph kann zudem um einige Pixel in x- und y-Richtung verschoben werden. Außerdem kann der Graph – je nach Anwendung – um seinen Schwerpunkt gedreht werden. An dieser Stelle unterscheiden sich die Verfahren in Bezug auf die Verwendung von S_ϕ und S_a als Vergleichsfunktion.
3. Lokale Verformung: Die einzelnen Knoten werden separat verschoben. Hier wird ein geeignetes λ verwendet, um die Verformung des Graphen einzuschränken.

In jedem Schritt wird für jede Position der Knoten der Abstand zum Bunch Graph berechnet. Die Position mit der höchsten Ähnlichkeit bzw. dem geringsten Abstand wird für den nächsten Schritt verwendet. Der initiale Graph hat die Struktur des Bunch Graph. Das Matching wird für verschiedene Stufen der Erkennung verwendet. Es findet Anwendung in der Segmentierung, der Normalisierung und der eigentlichen Erkennung. Abhängig vom Ziel des Matchings werden unterschiedliche Varianten des Bunch Graphs mit einer unterschiedlichen Anzahl von Knoten verwendet.

Abbildung 3 zeigt das Ergebnis eines Matchings. Wie man sieht, konnten in diesem Fall nicht alle Knoten exakt positioniert werden.

3.2 Training

Um das Matching durchführen zu können, müssen Bunch Graphs vorhanden sein. Diese werden im Training aus Trainingsbildern erstellt. Für jede Klasse von Objekten wird ein Bunch Graph erstellt. Das Problem ist auch hier die Positionierung der Knoten auf den Trainingsbildern. Bei den in Abschnitt 4 vorgestellten Anwendungen werden zwei unterschiedliche Vorgehensweisen vorgestellt.

Die erste Möglichkeit ist, bei allen Bildern die Knoten von Hand zu setzen. An den Positionen der Knoten werden die Jets berechnet und zwischen den Knoten Kanten gezeichnet, die mit der Distanz zwischen den Knoten

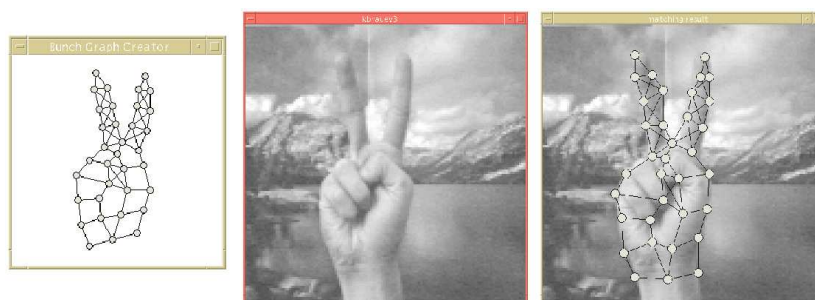


Abbildung 3: Matching eines Graphen mit einem Bild (aus [3]).

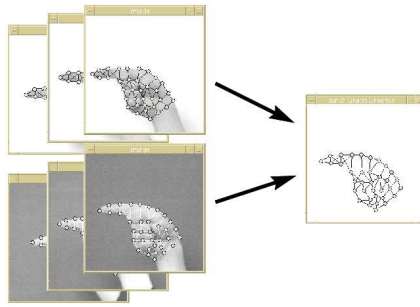


Abbildung 4: Generierung eines Bunch Graphs (aus [3]).

beschriftet werden. Die so entstandenen Image Graphs werden, je nachdem zu welcher Klasse das Objekt auf dem Bild gehört, zu Bunch Graphs zusammengefasst. Dazu werden, wie in Abschnitt 2.2 beschrieben, die Jets zusammengefasst und die Distanzvektoren gemittelt.

Eine andere Variante nutzt das Elastic Bunch Graph Matching auch im Training. Im ersten Bild für jede Klasse müssen die Knoten noch manuell positioniert werden. Für jedes weitere Bild kann der Image Graph mit dem Matching-Algorithmus generiert werden. Der erstellte Image Graph wird dann in den Bunch Graph seiner Klasse eingefügt. Solange der Bunch Graph nur wenige Graphen enthält, ist es unter Umständen nötig, die automatisch erstellten Graphen von Hand zu korrigieren oder anzupassen. Sobald der Bunch Graph aber groß genug ist, können große Mengen von Trainingsdaten verarbeitet werden.

Die manuelle Positionierung der Knoten kann unterstützt werden, indem zwischen bestimmten Knoten in deren Schwerpunkt automatisch ein Knoten gesetzt wird. Dies kann helfen, Knoten an Punkten mit nur schwer zu bestimmenden Merkmalen zu setzen.

Abbildung 4 verdeutlicht die Erstellung eines Bunch Graphs für eine Handgeste. Dort werden sechs Image Graphs, die aus je drei Bildern mit hellem und dunklem Hintergrund erstellt wurden, verwendet.

3.3 Erkennung

Die Erkennung des Objektes auf einem Bild mit Elastic Bunch Graph Matching ist relativ einfach. Das zu erkennende Bild wird als Eingabe für den Matching Algorithmus genommen. Der Bunch Graph mit dem höchsten Ähnlichkeitswert ist der Repräsentant der Klasse. Bleibt der Ähnlichkeitswert für alle Bunch Graphs unter einem bestimmten Schwellwert, kann das Bild als „nicht erkannt“ gelten.

Unter Umständen muss das Bild vor der Erkennung noch normalisiert werden. Dabei wird die Position und die Größe des Objekts im Bild bestimmt und das Bild skaliert und beschnitten.

4 Anwendungen

4.1 Handgesten-Erkennung

4.1.1 Problembeschreibung

Gesten sind in der zwischenmenschlichen Kommunikation tief verankert. Die Gesten der Hand spielen dabei eine entscheidende Rolle. So sind beispielsweise in der Zeichensprache viele Gesten abhängig von der Haltung der Hand. Um eine gestenbasierte Mensch-Maschine-Kommunikation zu realisieren, ist eine Handgesten-Erkennung von großem Nutzen.

Bei vielen Anwendungen ist es wichtig, dass die Erkennung personenunabhängig ist. Es müssen geometrische Verzerrungen, die durch die individuelle Anatomie der Hand und durch unterschiedliche Ausführungen der Geste entstehen, toleriert werden. Zudem muss auch eine Erkennung von Bildern mit einem komplexen Hintergrund, der eine Segmentierung der Hand erschwert, möglich sein.

In [3] und [4] wird ein System vorgestellt, dass die Handgesten-Erkennung mit Elastic Bunch Graph Matching realisiert. In [4] wird das Ziel des Projekts folgendermaßen formuliert:

Our goal was a system that works in relatively unconstrained environments (...). It should not be necessary for subjects to take off rings, etc., before interacting with the system but they should be able to just „come as they are“.

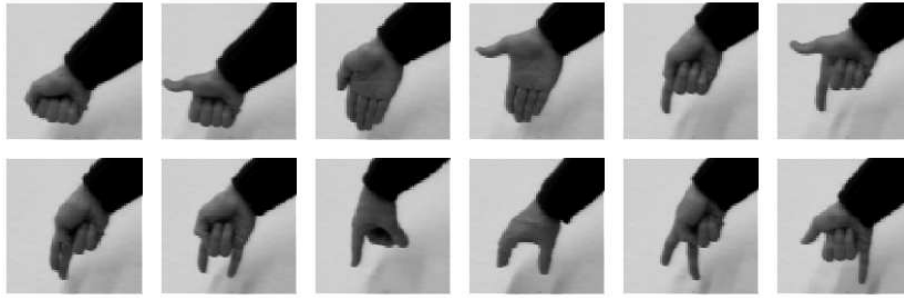


Abbildung 5: Die 12 Handgesten, die erkannt werden sollen (aus [4]).

Das System wurde in ein Gesture Interface für einen autonomen Roboter integriert. Der Roboter wird durch Gesten eines Menschen dazu angeleitet, Gegenstände aufzunehmen und abzulegen.

4.1.2 Verfahren

Eine Besonderheit bei diesem Verfahren ist die Erweiterung der Jets zu *Compound Jets*, in denen zusätzliche Typen von Merkmalen gespeichert werden. Dazu gibt es mehrere Möglichkeiten: Erstens können sich die Merkmalstypen von Knoten zu Knoten unterscheiden. Zum Beispiel können am Rand des Objekts andere Merkmale benutzt werden als im Objektinneren. Zweitens kann eine Kombination von Merkmalstypen gespeichert sein, die an allen Knoten gleich ist. Die dritte Möglichkeit ist, an den Knoten unterschiedliche Kombinationen von Merkmalstypen zu nutzen. In dem hier vorgestellten Verfahren wird die zweite Möglichkeit genutzt.

Ein Bunch Jet für einen Merkmalstyp \mathcal{F} am Knoten n besteht dann aus einer Menge $\mathcal{B}_{\mathcal{F}}^n = \{\mathcal{J}_{\mathcal{F}}^n(1), \dots, \mathcal{J}_{\mathcal{F}}^n(K)\}$. Die Bunch Jets für einen Knoten n werden zusammengefasst zu einem *Compound Bunch Jet* \mathcal{B}^n . Der daraus entstehende Bunch Graph wird *Compound Bunch Graph* genannt. Die Vergleichsfunktion für Jets muss an die unterschiedlichen Merkmalstypen angepasst werden, indem die Vergleichsfunktionen für die einzelnen Merkmalstypen gewichtet gemittelt werden. Dann ergibt sich für zwei Compound Jets \mathcal{J} und \mathcal{J}' und merkmalsbezogene Vergleichsfunktionen $S_{\mathcal{F}}$ die Funktion:

$$S = \sum_{\mathcal{F}} w_{\mathcal{F}} S_{\mathcal{F}}(\mathcal{J}_{\mathcal{F}}, \mathcal{J}'_{\mathcal{F}}), \quad \sum_{\mathcal{F}} w_{\mathcal{F}} = 1 \quad (6)$$

Die Vergleichsfunktion für Compound Bunch Graphs und Image Graphs wird analog angepasst.

Es werden zwei neue Merkmalstypen eingeführt: *Color Average* und *Color Gabor Jet*. Die Farben sind HS-Werte aus dem HSV-Farbraum (Hue, Saturation, Value). Die Vergleichsfunktion für Color Average vergleicht den Durchschnitt von 3×3 Pixel Regionen. Die Color Gabor Jets sind Gabor Jets, die auf Bildern arbeiten, deren Pixel die Ähnlichkeit des Original-Pixels zur Hautfarbe ausdrücken. Die unterschiedlichen Merkmale und deren Zusammenfassung werden in Abbildung 6 anschaulich dargestellt.

Im Training wurden für jede Geste Bilder von drei Personen verwendet, die jeweils einmal vor hellem und vor dunklem Hintergrund (einheitlich) aufgenommen wurden. Insgesamt wurden sechs Bilder pro Geste, bei zwölf Gesten (siehe Abbildung 5), also 72 Bilder ausgewertet. Aus jedem Bild wurden drei Image Graphs erstellt (für jeden Merkmalstyp einer) die aus 15 Knoten bestehen. Die resultierenden 216 Image Graphs wurden zu zwölf Compound Bunch Graphs zusammengefasst. Alle Knoten wurden auf allen Trainingsbildern manuell auf anatomisch besonderen Punkten der Hand positioniert.

Das Matching gliedert sich in vier Schritte:

1. Der Graph wird in Schritten von fünf Pixeln in x- und y-Richtung bewegt.
2. Der Graph wird um seinen Schwerpunkt in sieben Schritten um bis zu 15° gedreht. Außerdem wird er in zwei Schritten um je drei Pixel in x- und y-Richtung verschoben.
3. Vergrößerung um bis zu 20% und Verkleinerung um bis zu 10% (in fünf Stufen). Zusätzlich darf sich der Graph in zwei Schritten um vier Pixel verschieben.
4. Die Knoten werden um maximal ein Pixel verschoben, um geometrische Verzerrungen auszugleichen.

In allen Schritten wird S_a als Vergleichsfunktion für Gabor Jets und Color Average benutzt; S_ϕ für die Color Gabor Jets. In den ersten drei Schritten bleibt der Graph starr. In den Schritten 2 bis 4 wird jeweils das beste Ergebnis des vorhergehenden Schrittes verwendet.

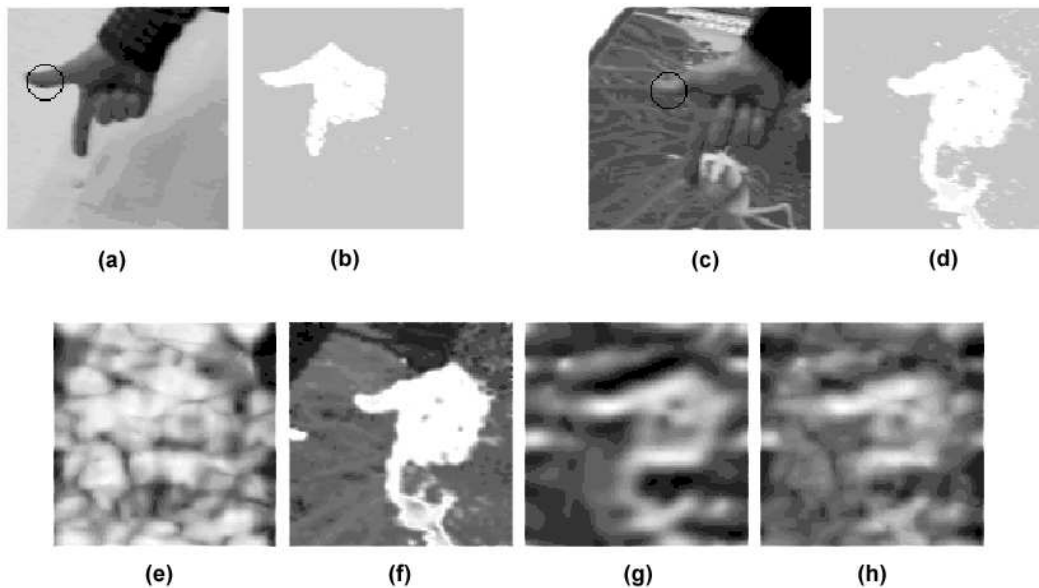


Abbildung 6: Vergleichsfunktionen zwischen zwei Bildern (a) und (c). (b), (d) sind die dazugehörigen Differenzbilder zur Hautfarbe. (e) bis (h) zeigen Ähnlichkeitsfunktionen für unterschiedliche Merkmalstypen: (e) Gabor Jets, (f) Color Average, (g) Color Gabor Jets und (h) Compound Jets für alle drei Merkmalstypen. Je heller ein Pixel ist, desto ähnlicher ist die entsprechende Region in (c) zur Region in (a) (aus [4]).

4.1.3 Ergebnisse

Es wurden 604 Bilder mit einheitlichem und 338 Bilder mit komplexem Hintergrund getestet. Die optimale Gewichtung der Merkmalstypen betrug $w_{Gabor} = w_{GaborColor} = 25\%$, $w_{Color} = 50\%$. Die Erkennungsrate für Bilder mit einheitlichem Hintergrund betrug 92,9%, für komplexe Hintergründe lag sie bei 85,8%. Praktisch alle Fehler traten bei Bildern mit starken geometrischen Verzerrungen durch unterschiedliche Ausführung der Geste, unterschiedliche Anatomie der Hand, Tiefenrotation oder Kombinationen aus diesen auf.

4.2 Gesichtserkennung

4.2.1 Problembeschreibung

Die Gesichtserkennung findet in vielen Bereichen Anwendung. Sie wird zur Zutrittskontrolle benutzt, zur Autorisierung und auch – als Gesichts-Detektion – in der Analyse von komplexen Szenen, wie zum Beispiel Videobildern von Überwachungskameras. Im Bereich der Zutrittskontrolle sei exemplarisch auf das Produkt ZN-Face[®] von ZN Vision Technologies AG (<http://www.zn-ag.de>) hingewiesen, das Elastic Graph Matching nutzt. Benutzt wird es zum Beispiel zur Zutrittskontrolle in einem Atomkraftwerk und im Zoo Hannover [7].

In [1, 2] wird ein System vorgestellt, das die Gesichtserkennung mit Elastic Bunch Graph Matching realisiert. Ziel des Projekts war es, ein System zu entwickeln, das Personen anhand eines einzelnen Bildes erkennt, wobei auch nur ein einzelnes Bild pro Person in der Referenzdatenbank enthalten ist. Das Problem dabei ist, dass die Gesichter in verschiedenen Posen, mit unterschiedlichen Gesichtsausdrücken, Positionen und Größen auftreten. Ziel ist es, diese Variationen durch diskriminative Merkmale auszugleichen.

Für die eigentliche Gesichtserkennung wird kein Bunch Graph benötigt, denn für jede Person existiert nur ein Bild, aus dem sich kein Bunch Graph erstellen lässt; die Gesichtserkennung ist eine *in-class recognition*. Aber für mehrere andere Teilaspekte der Erkennung, die im nächsten Abschnitt beschrieben werden, wird Elastic Bunch Graph Matching verwendet.

Das beschriebene Verfahren ist nicht speziell für die Gesichtserkennung entwickelt, so dass es auch für die Erkennung von anderen Objekten eingesetzt werden kann.

4.2.2 Verfahren

Eine Besonderheit im Matching beziehungsweise in der Definition der phasensensitiven Vergleichsfunktion für Jets ist die Benutzung eines *Displacement Vectors*. Wenn zwei Jets J und J' zu Positionen auf dem Objekt mit einem

kleinen relativen Abstand \mathbf{d} gehören, dann kann die daraus resultierende Phasenverschiebung näherungsweise durch $\mathbf{d}\mathbf{k}_j$ ausgeglichen werden. Dies führt zu folgender Definition von S_ϕ

$$S_\phi(\mathcal{J}, \mathcal{J}') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \mathbf{d}\mathbf{k}_j)}{\sqrt{\sum_j a_j^2 \sum_j a'_j^2}} \quad (7)$$

Um \mathbf{d} zu berechnen, kann S_ϕ in seiner Taylorentwicklung um $\mathbf{d} = 0$ maximiert werden [1]. Das berechnete Displacement kann dazu benutzt werden, um einen Jet exakter zu positionieren. Soll ein Jet \mathcal{J} auf dem Bild in der Nähe des Punktes \mathbf{x}_0 plaziert werden, dann wird \mathcal{J} mit dem Jet $\mathcal{J}_0 = \mathcal{J}(\mathbf{x}_0)$ verglichen und das displacement $\mathbf{d}_0 = \mathbf{d}(\mathcal{J}, \mathcal{J}_0)$ berechnet. Dann wird ein Jet \mathcal{J}_1 an der Position $\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{d}_0$ betrachtet und wieder das displacement berechnet. Dieses Verfahren wird iteriert, bis der Jet genau genug positioniert ist. Die schrittweise Positionierung eines Jets bzw. eines Knotens wird in mehreren Schritten des Matchings verwendet. Durch die Verwendung des Displacement Vectors können die Knoten exakter positioniert werden.

Im Training werden für alle auftretenden Gesichtsposen Bunch Graphs – hier als *Face Bunch Graphs (FBG)* bezeichnet – erstellt. Dabei ist zu beachten, dass die Bilder für einen FBG möglichst viele Varianten von Gesichtern beinhalten. Es werden unterschiedliche Hautfarben, Gesichtsausdrücke, Gesichtsformen, Bärte, Brillen usw. von Personen unterschiedlichen Alters und Geschlechts verarbeitet. Diese FBGs dienen der Klassifizierung der Pose (Frontal, Profil, Halb-Profil, ...). Hier wird die in Abschnitt 3.2 beschriebene Methode des Matchings zur Positionierung von Image Graphs auf neuen Trainingsbildern genutzt. Außerdem wird für jede Person, die erkannt werden soll, ein Image Graph – in diesem Fall *Model Graph* genannt – berechnet und in der sogenannten Model Gallery gespeichert.

Ein Image Graph wird in zwei Schritten aus einem Bild berechnet: Im ersten Schritt, der Normalisierung, wird die Position und die Größe des Gesichts bestimmt. Dazu werden Image Graphs genutzt, die mehr Knoten auf den Gesichtskanten als im Gesichtsinnen besitzen. Das Bild wird dann auf eine Standardgröße geschnitten und skaliert. Der zweite Schritt berechnet dann auf dem standardisierten Bild einen Image Graph, der für die Erkennung benutzt werden kann. Der erste Schritt wird in [8] näher beschrieben. In dieser Arbeit liegt der Fokus auf dem zweiten Schritt. Dort wird folgendes Matching verwendet:

1. Der FBG wird zu einem *Average Graph* verdichtet, indem der Mittelwert der Absolutwerte der Jets berechnet wird. Der Average Graph wird dann auf einem 4×4 Raster über das Bild bewegt. An der besten Position kann der Graph noch um 1 Pixel um seine aktuelle Position verschoben werden.
2. Es wird wieder der FBG verwendet; dieser wird auf vier verschiedenen Positionen (± 3 Pixel in x- und y-Richtung) jeweils um bis zu 18% vergrößert und verkleinert. Für alle acht Variationen wird für jeden Jet – bzw. für den local Expert des Jets im FBG – das Displacement berechnet und die Struktur des Graphen entsprechend geändert.
3. Schritt 2 wird wiederholt, wobei diesmal x- und y-Koordinaten unabhängig voneinander verändert werden.
4. Alle Knoten werden in pseudo-zufälliger Reihenfolge an ihrer aktuellen Position verschoben. Die geometrische Verformung wird dabei mit $\lambda = 2$ gewichtet.

Wenn FBGs für die Posen, Model Graphs für alle Bilder in der Galerie und ein Image Graph für ein zu erkennendes Bild vorliegen, kann die Erkennung gestartet werden. Da das Bild in der Galerie und das Testbild unter Umständen das Gesicht in unterschiedlichen Posen zeigen, müssen für Graphen unterschiedlicher Struktur Abbildungen zwischen den Knoten definiert sein. Die Beziehung zwischen den Knoten wurde manuell festgelegt. Für einen Model Graph \mathcal{G}^M und einen Image Graph \mathcal{G}^I mit unterschiedlicher Pose wird die Vergleichsfunktion so definiert:

$$S_{\mathcal{G}} = \frac{1}{N'} \sum_{n'} S_a(\mathcal{J}_{n'}^I, \mathcal{J}_{n'}^M) \quad (8)$$

Wobei $n_{n'}$ im Model Graph der zu n' im Image Graph korrespondierende Knoten ist und N' die Anzahl der Knoten im Image Graph mit einem korrespondierenden Knoten im Model Graph.

Für die Erkennung wird der Image Graph des Testbilds mit Hilfe von $S_{\mathcal{G}}$ mit allen Model Graphs in der Galerie verglichen. Eine Person wird korrekt erkannt, wenn der entsprechende Model Graph den höchsten Ähnlichkeitswert ergibt.

4.2.3 Ergebnisse

Das System wurde mit zwei unterschiedlichen Bilddatenbanken getestet: der FERET-Database und der Bochum-Database. Beide enthalten Bilder von Gesichtern in drei unterschiedlichen Posen (Frontal, Profil, Halbprofil) und zwei unterschiedlichen Gesichtsausdrücken in der frontalen Ansicht pro Person. In der FERET-Database liegen die Profile und Halbprofile sowohl von links als auch von rechts vor. Die Bilder der linken Gesichtshälfte wurden gedreht, so dass einheitlich nur Profile und Halbprofile von rechts vorlagen. Allerdings ist in der FERET-Database die Drehung des Kopfes im Halbprofil nicht bei allen Bildern exakt gleich, so dass es dort zu schlechten Ergebnissen kommt (siehe Tabelle 1). In der Bochum-Database liegen die Gesichter in relativ einheitlicher Größe vor, so dass keine Normalisierung nötig war. Die Unterschiede im Gesichtsausdruck sind aber deutlicher als in der FERET-Database, so dass die Erkennungsraten für die Frontalaufnahmen schlechter ausfallen (siehe Tabelle 2).

Tabelle 1: Ergebnisse für Tests auf der FERET-Database (f: Frontale Aufnahme; a, b: Gesichtsausdruck a und b; h: Halbprofil; p: Profil; l, r: links und rechts) (aus [2]).

Model Gallery	Testbilder	Erkennungsrate (%)
250 fa	250 fb	98
250 hr	181 hl	57
250 pr	250 pl	84
249 fa + 1 fb	171 hl + 79 hr	18
171 hl + 79 hr	249 fa + 1 fb	17
170 hl + 80 hr	217 pl + 33 pr	9
217 pl + 33 pr	170 hl + 80 hr	12

Tabelle 2: Ergebnisse für Tests auf der Bochum-Database (f: Frontale Aufnahme; a: Neutraler Gesichtsausdruck, b: anderer Gesichtsausdruck; 11°, 22° gedrehter Kopf) (aus [2]).

Model Gallery	Testbilder	Erkennungsrate (%)
108 fa	108 fb	91
108 fa	108 11°	94
108 fa	108 22°	88

5 Diskussion

Das Elastic Bunch Graph Matching zeigt bei schwierigen Objekterkennungsaufgaben, wie zum Beispiel bei komplexen Bildhintergründen, gute Ergebnisse. Allerdings zeigen sich auch Einschränkungen. So sinkt die Erkennungsleistung bei Rotationen in der Tiefe stark ab. Ein Grund dafür ist, dass man diese Rotation nur schwer modellieren kann. Bei der Gesichtserkennung sieht man, dass Gesichter mit einem Drehungswinkel, der etwas von der Vorgabe abweicht, nur sehr schlecht erkannt werden. Ähnliches beobachtet man bei der Handgestenerkennung: auch hier sind Tiefenrotationen Auslöser für Erkennungsfehler.

Die schlechte Modellierung der Tiefenrotation führt unter anderem zu folgender Beobachtung: Wenn eine Frontalaufnahme mit einer Model Gallery von Halbprofilen verglichen wird, liefert die Erkennung eher ein Bild von einer anderen Person, das in einem „korrekten“ Aufnahmewinkel aufgenommen wurde, als das korrekte Bild, wenn auf diesem der Kopf in einem größeren Winkel rotiert ist.

Im Vergleich der Ergebnisse auf der FERET-Database (Tabelle 1) mit denen auf der Bochum-Database (Tabelle 2) wird deutlich, wie groß die Abhängigkeit vom exakten Aufnahmewinkel ist. Während bei Bildern aus der Bochum-Database unterschiedliche Posen in Model Gallery und Testbildern Erkennungsraten von ca. 90 % liefern, gehen die Erkennungsraten in ähnlichen Tests auf der FERET-Database auf 10 bis 20 % zurück. Begründet wird dies, wie oben bereits erwähnt, mit den variierenden Aufnahmewinkeln für die Halbprofil-Bilder. Verglichen mit anderen Methoden sind die Erkennungsraten bei gleichen Posen in Model Gallery und Testbildern gut (siehe Tabelle 3). Insbesondere bei Halbprofil- und Profilaufnahmen erzielt Elastic Bunch Graph Matching gegenüber PCA

bessere Ergebnisse. Tabelle 3 ist mit Vorsicht zu bewerten, da die Model Galleries unterschiedlich groß sind, was den direkten Vergleich erschwert.

Bessere Ergebnisse bei tiefenrotierten Bildern liefert ein Verfahren, das die Effekte der Tiefenrotation mit linearer Jet Transformation ausgleicht. Ein solches Verfahren wird in [9] vorgestellt. Dort wird auf einer Model Gallery mit 90 Bildern, die von 0° auf 45° transformiert wurden, und 90 Testbildern (45° Rotation) eine Erkennungsrate von 53 % gemessen (FERET-Database).

Bessere Erkennungsraten könnten mit einer sorgfältigeren Auswahl von Trainingsbildern erreicht werden, wenn diese viele Variationen und wenig Redundanz aufweisen.

Man könnte vermuten, dass die Erkennungsfehler in der Handgestenerkennung würden abnehmen, wenn die Knoten im letzten Schritt des Matchings mehr verschoben werden und der Graph dadurch mehr verformt werden könnte. Dass dies nicht der Fall ist, wird in [4] damit begründet, dass die geometrischen Verformungen keiner statistischen Verteilung folgen, was durch einen Diffusionsprozess modelliert werden könnte. Stattdessen erlaubt die Kinematik der Hand nur bestimmte zusammenhängende Bewegungen von Knoten. Diese Bewegungen zu modellieren und in das Matching zu integrieren, könnte die Erkennungsrate verbessern.

Es scheint allgemein eine Schwäche der hier vorgestellten Verfahren zu sein, dass der Graph nur wenige Freiheitsgrade besitzt. Die Position des Graphen wird zwar ziemlich genau und in vielen abgestuften Schritten bestimmt, aber Verformungen sind nur sehr eingeschränkt möglich. Auch werden Drehungen – wenn überhaupt – nur grob approximiert. Die Bilder müssen dadurch bestimmte Voraussetzungen erfüllen, wobei eigentlich das Ziel war, die Auswahl der Bilder möglichst wenig einzuschränken.

Während des Matchings wird für jeden Schritt das Maximum des vorgehenden Schrittes verwendet. Hier besteht die Gefahr, dass es sich um ein lokales Maximum handelt, was zu schlechten Ergebnissen im nächsten Schritt und im gesamten Matching führen kann. Dieses Problem könnte beseitigt werden, indem man für jeden Schritt nicht nur das beste Ergebnis des vorhergehenden Schrittes, sondern die n besten Ergebnisse weiterverfolgt. Die so aufgestellten Hypothesen für einen optimalen Graphen können in den weiteren Schritten verfolgt oder – bei zu schlechten Ergebnissen – wieder verworfen werden, um den Suchraum nicht zu stark zu vergrößern.

Ein weiterer Nachteil ist die aufwändige Erzeugung von neuen Bunch Graphs. Wenn die Knoten, wie in [4], auf jedem Trainingsbild manuell positioniert werden müssen, bedeutet das automatisch eine stark eingeschränkte Trainingsdatenbank und somit auch weniger korrekt erkannte Variationen. Die Automatisierung in [1] bringt bereits Verbesserungen. Es ist aber immer noch mit Aufwand verbunden, die Graphen zu kontrollieren und zu korrigieren.

Ein Vorteil des Elastic Bunch Graph Matchings gegenüber anderen Verfahren zur Objekterkennung ist, dass für neue Objekte kein intensives Training notwendig ist. Bei der Gesichtserkennung reicht ein Bild pro Person, um dieses in die Erkennung mit einzubeziehen (vorausgesetzt, dass bereits geeignete FBGs vorhanden sind). Auch für andere Objekte reicht eine kleine Menge von typischen Bildern des Objekts, um ein Modell zu erstellen, das ziemlich robust gegenüber bestimmten Variationen ist.

Tabelle 3: Vergleich von unterschiedlichen Methoden und deren Ergebnissen. Es wurden gleiche Posen für Model Gallery und Testbilder verwendet. Nomenklatur wie in Tabelle 1 (aus [1]).

Methode	Model Gallery	Testbilder	Erkennungsrate (%)
Normalisierte Kreuz-Korrelation [10]	194 fa	194 fb	62
Radial basis function network [11]	100 fa	100 fb	83
Trained matching pursuit filters [12]	311 fa	311 fb	95
Principal Component Analysis (PCA) [13]	150 fa	150 fb	99
	150 hr	150 hl	38
	150 pr	150 pl	32
Elastic Bunch Graph Matching [1]	250 fa	250 fb	97
	250 hr	181 hl	57
	250 pr	250 pl	84

6 Fazit

Elastic Bunch Graph Matching liefert in der Handgesten- und der Gesichtserkennung weitgehend gute Ergebnisse, solange die Objekte nicht zu stark verzerrt oder gedreht sind. Somit wird sie dem Anspruch, keine Voraussetzungen an die Eingabebilder zu stellen, nicht ganz, aber doch in weiten Teilen gerecht. Zukünftige Verbesserungen im

Trainingsmaterial und in der automatischen Positionierung von Knoten können das Training vereinfachen und bessere Bunch Graphs, die mehr Variationen abdecken, erzeugen.

Die Konstruktion der Bunch Graphs könnte mit den folgenden Methoden noch verbessert werden: Statt die Variationen der geometrischen Merkmale in den Trainingsbildern nur durch Mittelung der Distanzvektoren zu behandeln, könnte der Bunch Graph mit *Active Shape Models* [14] generiert werden. Das halb-automatische Training, wie es in Abschnitt 3.2 beschrieben ist, könnte durch die Verwendung einer *Expectation-Maximization-Iteration* [15] zu besseren Ergebnissen führen. Diese Methoden werden in den Anwendungen, die in Abschnitt 4 vorgestellt wurden, weder angewendet noch in den entsprechenden Veröffentlichungen erwähnt.

Da Elastic Bunch Graph Matching keine spezialisierte Methode ist, läßt sie sich auch in medizinischen Anwendungen einsetzen. In allen Bereichen, in denen bildgebende Verfahren eingesetzt werden und die abgebildeten Objekte sich durch besondere Regionen auszeichnen und klassenweise ähnliche geometrische Formen haben, wären ähnliche wie die in dieser Arbeit vorgestellte Verfahren anwendbar. In der Röntgendiagnostik könnten zum Beispiel unterschiedliche Knochen auf Röntgenbildern lokalisiert, segmentiert und erkannt werden.

Literatur

- [1] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," in *Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns, CAIP'97, Kiel* (G. Sommer, K. Daniilidis, and J. Pauli, eds.), no. 1296, (Heidelberg), pp. 456–463, Springer-Verlag, 1997.
- [2] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.
- [3] J. Triesch and C. von der Malsburg, "Robust classification of hand postures against complex backgrounds," in *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition 1996, Killington, Vermont, USA, October 14–16*, pp. 170–175, IEEE Computer Society Press, 1996.
- [4] J. Triesch and C. von der Malsburg, "A system for person-independent hand posture recognition against complex backgrounds," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 23, no. 11, pp. 1449–1453, 2001.
- [5] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.
- [6] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, pp. 300–311, 1993.
- [7] P.-M. Ziegler, "Adlerauge - Europas größte Gesichtserkennungsanlage im Zoo Hannover," *c't magazin für computer technik*, vol. 9, pp. 26–28, 2003.
- [8] N. Krüger and C. von der Malsburg, "Determination of face position and pose with a learned representation based on labelled graphs," *Image and Vision Computing*, vol. 15, pp. 665–673, 1997.
- [9] T. Maurer and C. von der Malsburg, "Linear feature transformation to recognize faces rotated in depth," in *Proceedings of the International Conference on Artificial Neural Networks, ICANN'95, Paris*, pp. 353–358, 1995.
- [10] G. G. Gordon, "Face recognition from frontal and profile views," in *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zürich* (M. Bichsel, ed.), pp. 47–52, MultiMedia Laboratory, University of Zürich, 1995.
- [11] S. Gutta, J. Huang, D. Singh, I. Sham, B. Takacs, and H. Wechsler, "Benchmark studies on face recognition," in *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zürich* (M. Bichsel, ed.), pp. 227–231, MultiMedia Laboratory, University of Zürich, 1995.
- [12] P. J. Phillips and Y. Vardi, "Data driven methods in face recognition," in *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zürich* (M. Bichsel, ed.), pp. 65–70, MultiMedia Laboratory, University of Zürich, 1995.
- [13] B. Moghaddam and A. P. Pentland, "Face recognition using view-based and modular eigenspaces," in *Proceedings SPIE Conference on Automatic Systems for the Identification and Inspectations of Humans*, pp. 12–21, volume SPIE 2277, 1994.
- [14] T. Cootes, C. Taylor, and A. Lanitis, "Active shape models: Evaluation of a multiresolution method for improving image search," in *5th British Machine Vision Conference* (E. Hancock, ed.), (York, England), pp. 327–336, BMVA Press, 1994.
- [15] J. Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," Tech. Rep. ICSI-TR-97-021, University of Berkeley, 1997.