

Gesichts-Erkennung I

Biometrische Systeme - Der Körper als Passwort (WS 2001/02)

Neuroinformatik

von: Carsten Wächter, Stefan Römer

email: {cw1,sr1}@informatik.uni-ulm.de

Einleitung

An Gesichtserkennung besteht ein großes kommerzielles Interesse. Zum Beispiel lassen sich in Kaufhäusern Gewohnheiten von Altersgruppen bestimmen oder sogar von einzelnen Kunden. Gesichtserkennungssysteme werden auch schon häufig eingesetzt (z.B. auf den US-Flughäfen: Oakland, Boston, Rhode Island oder in amerikanischen Casinos wo Gesichtserkennung schon länger benutzt wird, um bekannte Falschspieler zu identifizieren).

Der große Vorteil von Gesichtserkennung gegenüber anderen biometrischen Verfahren liegt darin, das Personen identifiziert werden können, ohne es zu wissen. Daß heißt, sie müssen keine besondern Aktionen ausüben, um erkannt zu werden. Außerdem reichen recht geringe Bildauflösungen aus, wie sie zum Beispiel Weitwinkelobjektive liefern. So kann schon ab einer Auflösung von 18x24 Pixel Gesichtserkennung erfolgen.

Probleme bereiten dagegen: verschiedene Lichtquellen; Ausrichtung des Kopfes; Gesicht ist oft halbverdeckt; Veränderungen durch Haare, Bart, Brille, Makeup, ...; Personen altern; Personen haben Emotionen und schneiden „Grimassen“ wie lachen, reden, gähnen,...; usw.

Viele dieser Einschränkungen lassen sich jedoch durch geschickte Ansätze beheben. So geht oft eine Normalisierung des Bildes voraus, die mit Hilfe markanter Punkte im Gesicht (z.B. Nase, Augen, Mund, Kinn,...) die Rotation, Translation, Skalierung des Kopfes berücksichtigt.

Verschiedene Verfahren der Gesichtererkennung

Bei der Gesichtererkennung kann generell zwischen zwei verschiedenen Ansätzen unterschieden werden:

1. merkmalsbasierte Gesichtererkennung:

Einzelne Merkmale (features) werden aus dem Gesicht extrahiert und die Gesichter anhand dieser Merkmale klassifiziert. (Elastic Bunch Graph Matching, Gesichtserkennung anhand geometrischer Merkmale)

2. holistischer Ansatz:

Das komplette Gesicht wird betrachtet. (Template Matching, Fourier-Transformation, Eigenface-Methode)
In der Praxis kommen oft Kombinationen aus 1. und 2. vor.

1. Gesichtserkennung mit Hilfe des Template Matching



Template Matching ist ein sehr häufig verwendetes Verfahren der Gesichtserkennung. Das Verfahren versucht dabei die Ähnlichkeit zwischen einem Bild und einem Template (= vorgegebene Maske, die einem Bild oder einem Teil eines Bilds ähnlich ist) zu berechnen. Für die Identifikation wird das gegebene Gesicht mit allen in der DB gespeicherten Gesichtern (genauer gesagt nur mit den Templates) verglichen. Wenn n Templates zur Gesichtserkennung verwendet werden, erhält man einen n dimensional Vektor, der die Ähnlichkeit jedes Templates zum getesteten Bild der DB angibt. Die Ähnlichkeit kann zum Beispiel durch die Summe der Fehlerquadrate bestimmt werden.

Probleme:

- Maske muß bei möglichst vielen unterschiedlichen Personen passen
- Maske sollte möglichst unabhängig von Helligkeits- oder Kontraständerungen sein
- Template Matching ist sehr rechenaufwendig

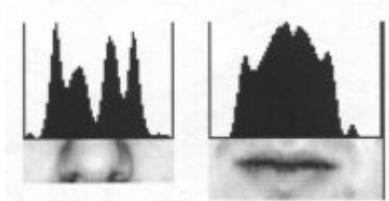
*vier Templates für Augen, Nase,
Mund und das ganze Gesicht*

2. Gesichtserkennung mit Hilfe geometrischer Merkmale

Bei diesem Verfahren werden geometrische Merkmale des Gesichts (zum Beispiel Position der Nase, der Augen, des Mundes und ihre relative Position zueinander) aus dem Bild extrahiert und als Zahlenwerte in einem Vektor gespeichert. Dabei wird folgendes gefordert:

- die Extraktion der Merkmale muß möglichst einfach sein
- die Abhängigkeit von der Umgebungsbeleuchtung muß möglichst gering sein
- kleine Änderungen im Bild, dürfen nur kleine Änderungen im Merkmalvektor verursachen
- der Vektor muß einen möglichst hohen Informationsgehalt besitzen

Gesetzmäßigkeiten (Constraints) eines Gesichtes werden ausgenutzt um die Merkmalsextraktion zu erleichtern (zum Beispiel: Augen liegen nebeneinander, Nase liegt unter den Augen,...). Um Merkmale aus dem Gesicht zu extrahieren, wird hauptsächlich ein „Aufsummieren“ der Grauwerte einer Spalte (bzw. einer Zeile) verwendet, oder ein Gradientenoperator, der vertikale und horizontale Kanten bestimmt. Aber auch andere Methoden sind denkbar.



vertikales Aufsummieren



vertikales und horizontales Kantenbild

Beispiele für Merkmale:

- Lage der Augen
- Umriß des Gesichtes
- Vertikale Position der Nase und Nasenbreite
- Vertikale Position des Mundes, Mundbreite und Mundhöhe
- Vertikale Position und Dicke der Augenbrauen über der Augenmitte
- Breite des Gesichtes an der Nasenunterkante
- Breite des Gesichtes auf der Höhe des Mittelpunkts von Augen und Nase

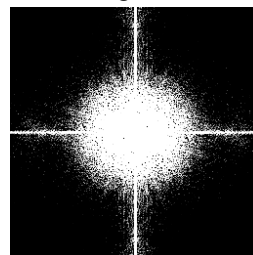
Aus diesen n Merkmalen wird ein n dimensionaler Vektor gebildet, der das gegebene Gesicht beschreibt, und zur Identifikation schnell mit einer DB verglichen werden kann. Probleme entstehen bei diesem Verfahren, wenn Gesichter unter verschiedenen Winkeln aufgenommen werden. In diesem Fall muß zuerst eine Normierung des Gesichtes erfolgen. Zum Beispiel wird eine seitliche Neigung des Kopfes über Winkel normiert.

3. Gesichtserkennung mit der 2-dimensionalen diskreten Fourier-Transformation

Die Grundidee dieses Verfahrens besteht darin, das Originalbild und das Vergleichsbild in den Frequenzbereich zu transformieren, um dort die Spektren der beiden Bilder einfacher vergleichen zu können.



Originalbild



Spektrum des Bilds (Amplitudenspektrum)

Durch die Varianz unterschiedlicher Spektren zueinander, lassen sich die Frequenzen bestimmen, die die meisten Informationen eines Gesichtes besitzen. Interessant ist hierbei, daß ein Großteil der Bildinformation zur Gesichtserkennung in den niedrigen Frequenzen enthalten ist. Daraus folgt, daß die Fourierkoeffizienten niedriger Ordnung wichtiger sind als die höherer Ordnung. Die Fourierkoeffizienten werden also entsprechend ihres Informationsgehalts angeordnet und nur die ersten n Koeffizienten für die Gesichtserkennung verwendet, wobei n hauptsächlich vom Einsatzgebiet abhängt. Zusätzlich kann noch ein Schwellwert eingeführt werden, damit Bilder, die keine Gesichter enthalten, nicht fälschlicherweise klassifiziert werden.

Vorteile:

- geringe Anzahl von Koeffizienten, die benötigt werden, um ein Gesicht zu klassifizieren
- es genügen 22 reale Frequenzen und 8 imaginäre Frequenzen, um eine korrekte Klassifikation zu erreichen

4. Gesichtserkennung mit Hilfe des „Elastic Bunch Graph Matching“ (← eigener Vortrag)

Beim „Elastic Bunch Graph Matching“ wird über das Gesicht ein Netz, ein sogenannter „Labeled Graph“, gelegt. Die Knoten dieses Graphen werden dabei auf markante Stellen des Gesichtes (den Landmarks) platziert. Zu jeder Kante des Graphen wird nun Länge und Winkel gespeichert. Zusätzlich werden Merkmale an den Knoten mit Hilfe von „Jets“ extrahiert. Jets bestehen aus mehreren Gabor-Wavelets, die unterschiedlich orientiert sind und unterschiedliche Größen aufweisen.

Die Gesichtserkennung mittels „Elastic Bunch Graph Matching“ erfolgt in drei Schritten: Lokalisierung des Gesichtes, Lokalisierung der Landmarks und Vergleich der Graphen. Beim Vergleich der Graphen werden zuerst Längen und Winkel der Kanten zueinander verglichen, und dann erst die Jets jedes Knoten.

Die sehr guten Erkennungsraten des „Elastic Bunch Graph Matching“ sind in erster Linie auf die Gabor-Wavelets zurückzuführen. Sie sind sehr robust gegenüber Helligkeits- und Kontraständerungen.

5. Gesichtserkennung mit Hilfe der Eigengesichter („Eigenfaces“) (← eigener Vortrag)

Die Grundlage der Eigenface-Methode ist die „Principal Component Analysis“ (PCA). Sie wird zur Dimensionsreduzierung bei Vektoren eingesetzt. Eine Dimensionsreduzierung ist beim „Eigenface“ Verfahren auch nötig, da hier jedes Bild quasi als ein Vektor der Dimension Breite*Höhe angesehen wird. Bei einem 512*512 Pixel großen Bild erhält man entsprechend einen 262144 dimensional Vektor. Jedoch haben Gesichter viele constraints, so daß alle Gesichtsvektoren lediglich in einem kleinen Bereich des gesamten Vektorraums aller Bilder angeordnet sind. Ziel ist es nun einen geeigneten Untervektorraum für Gesichtsbilder zu suchen. Die Basisvektoren dieses Untervektorraums sind die sogenannten principal components. Originalgesichter können durch Linearkombination dieser Basisvektoren (=Eigenvektoren) wieder angenähert werden.

6. Gesichtserkennung mit Hilfe eines neuronalen Netzes

Für Gesichtserkennung kommen verschiedene neuronale Netzwerk-Architekturen zum Einsatz, zum Beispiel: Multi-Layer Perzeptron Netze (MLP), Radial-Basis-Funktion Netze (RBF) und Self Organizing Maps (SOM), ... Zuerst werden die Gesichter der Trainingsmenge auf passende Größe und Orientierung normiert und Beleuchtungsunterschiede werden weitmöglichst herausgerechnet. Dann werden die Bilder in einzelne Trainingsmengen unterteilt (z.B. in Brillenträger und Nicht-Brillenträger) bzw. (Person A, Person B, ...). Der Lernvorgang des Netzes erfolgt durch Fehlerrückmeldungen zur jeweils vorhergehenden Netzwerkschicht. Dazu werden (beim überwachten Lernen) Trainingsmuster an die Eingabeschicht angelegt und dann an der Ausgabeschicht die Sollwerte mit den Istwerten verglichen, um die Gewichte der Neuronen anzupassen, um die Fehler zu verkleinern. Wichtig ist hierbei, daß sich das Netz nicht auf die Beispielmuster der Trainingsmenge spezialisieren darf. Es wäre sonst „übertrainiert“. Abhilfe schafft die Aufteilung in eine Trainingsmenge und eine Validierungsmenge. Nach jedem Trainingszyklus wird der Fehler für die Validierungsmenge ermittelt und wenn der Fehler für die Validierungsmenge minimal ist, dann wird das Training abgebrochen.

Neuronale Netze werden auch häufig als Klassifikator eingesetzt um bestehende Verfahren noch zu verbessern. Sie liefern beim Vergleich von Vektoren auf Ähnlichkeit meist bessere Ergebnisse als die Nearest-Neighbour-Methode. Sie sind gut für schwierige Probleme, jedoch ist eine Aussage über die Zuverlässigkeit außerhalb des „Labors“ schwer, da sie oft unerwartet reagieren/klassifizieren, falls sie flexibel dazulernen sollen (also nachtrainiert werden). Die Trainingszeit kann unter Umständen auch ziemlich lange sein.

Klassifizierung von Personen mittels eines RBF(Radial Basis Function)-Netzwerks

1. Konzept eines generellen RBF-Netzwerks

Radiale Basisfunktionen besitzen bestimmte Eigenschaften (Haupttheorem: RBF genau dann, wenn die Funktion positiv definit ist), die es erlauben das Interpolationsproblem bei einer Klassifizierung von Daten lösen zu können.

$$\mathcal{M} = \{(x^\mu, t^\mu) : \mu = 1, \dots, M\}, x^\mu \in \mathbb{R}^n, t^\mu \in \mathbb{R}^m$$
$$G(x^\mu) = t^\mu \quad \forall \mu = 1, \dots, M$$

Das heisst: Gesucht ist eine Funktion G , die bei Eingabe eines bestimmten Datensatz-Vektors x dessen zugehörige Klasse t berechnen kann. Durch die Art der Daten ist dies natürlich nicht immer zu 100% gewährleistet (Überschneidungen von Datenklassen) und erfordert deshalb Basisfunktionen, die eine möglichst genaue oder zumindest „weiche“ Trennung zwischen diesen Daten vollziehen können. Weiterhin soll diese Funktion auch ihr unbekannte Datensätze „richtig“ klassifizieren können. Über eine Linearkombination der Basisfunktionen erreicht man eine Näherung dieser gesuchten Funktion.

Einige Basisfunktionen, die diese Eigenschaft besitzen, sind z.B.

$$h(\mathbf{r}) = e^{-\frac{r^2}{2\sigma^2}}$$

Gauss

$$h(\mathbf{r}) = (c^2 + r^2)^\beta$$

Multiquadratische Funktion (und deren Inverse)

Durch die „Verschaltung“ dieser Basisfunktionen (Neuronen) zu einem gewichteten (neuronalen) Netzwerk wird die Linearkombination erreicht. Eine einfache visuelle Vorstellung der Funktionsweise eines solchen Netzwerks ist die zweidimensionale Interpretation: Datenpunkte verschiedener Klassen befinden sich auf einer Ebene. Das Netzwerk wird initialisiert (z.B. eine bestimmte Anzahl von 2D-Gaussglocken gleichmäßig auf der Ebene „verteilen“ und jeder Glocke eine Klasse zuordnen). Durch den Lernprozess des Netzwerks „wandern“ nun die Basisfunktionen auf dieser Ebene umher und umschließen die jeweiligen Datenpunkte abhängig von ihrer Klassifikation.

2. Konzept des hier vorgestellten Klassifikator-Netzwerks

gewünschte Eigenschaften:

- eine einfache und schnelle Skalierung des Netzwerks muß möglich sein, d.h. Personen hinzufügen/löschen
- wenige bis keine Fehlklassifizierungen, vor allem bei Personen mit „ähnlichen“ Gesichtsmerkmalen

Erreicht werden soll dies durch einen speziellen Aufbau des Gesamtnetzwerks, das jeder zu klassifizierenden Person genau ein ganzes RBF-Netzwerk (=Unit) „zur Verfügung“ stellt/zuordnet. Das heisst, das jeweils eine Unit einzeln entscheiden kann, ob ein eingegebener Datenvektor zu der zugeordneten/gelernten Person dieser Unit passt (=Datenvektor ist dieselbe Person) (bzw. eben nicht (=andere Person)).

Dies verkleinert zudem die Anzahl der Fehlklassifizierungen, da die restlichen Units des Gesamtnetzwerks (die ja anderen Personen zugeordnet sind) idealerweise eine Negativantwort liefern sollten.

Eine weitere Raffinesse dieses Klassifikators besteht in einer Art Vorverarbeitung der Datensätze. Da das Training des Netzwerks normalerweise bei jeder neuen bzw. gelöschten Person komplett von neuem geschehen müßte (und zwar für alle Units mit allen Datensätzen!), wäre ein schnelles Bearbeiten der Datenbasis (z.B. während der Benutzung des Systems) ab einer bestimmten Anzahl nicht mehr möglich und würde eine regelmäßige Wartungsphase erfordern.

Ein weiteres Problem das mit diesem zusammenhängt ist der Trainingserfolg einer RBF-Unit. Wird mit der kompletten Datenbasis trainiert ist es für die Unit „schwieriger“ eine sinnvolle Klassifizierung/Unterteilung zu finden und kann unter Umständen eine größere Zahl von Fehlklassifizierungen zur Folge haben, da die Unit sozusagen „überfordert“ ist.

Deswegen „sortiert“ der Klassifikator die Datensätze nach ihrer Ähnlichkeit zueinander. Sehr verschieden wirkende Personen sind demnach weniger interessant für den Lernprozess einer Unit als Personen mit sehr gleichartigen Gesichtsmerkmalen. Denn es ist davon auszugehen, das das Netzwerk bei einer richtigen Klassifizierung sehr ähnlicher Personen (z.B. Brüder) auch die grundverschiedenen Datensätze auseinanderhalten kann (z.B. bärtiger, kurzhaariger Brillenträger vs. Langhaariges, junges Mädchen).

Eine neu einzufügende Person wird deshalb nur mit einer Auswahl der anderen, bereits im System vorhandenen Personen trainiert. Diese Auswahl trifft der Klassifikator anhand der simpel zu berechnenden euklidischen Vektordistanz (=nächste Nachbarn) und zweier Kriterien:

- Single Best Negative: Durchschnitt aller Vektorabstände der einzelnen Posen/Trainingsdaten der zwei zu vergleichenden Klassen zueinander. Dies ergibt einen Wert pro Klasse und damit ein einfacher/schneller Vergleich und „Sortierung“ der Ähnlichkeit
- Multiple Best Negative: Für jede Pose/jedes Bild eine Auswahl entsprechend aller Klassen (und evtl. auch der Posen). Ergibt mehr Rechenaufwand aber eine vielschichtigere Auswahl zum Training mit besseren Klassifikationsergebnissen.

Vorteile ergeben sich durch die sehr stark verkürzten Trainingszeiten (jede Unit bekommt einen wesentlich kleineren Ausschnitt der Datenbasis zum Training präsentiert und bereits bestehende und trainierte Units bleiben bei Änderungen der Datenbasis unbeeinflusst), als auch die genauere Klassifikation sehr ähnlicher Datensätze.

Literaturzitate

- [1] A. Jonathan Howell - „Introduction to Face Recognition“
- [2] A. Jonathan Howell - „Face Unit Radial Basis Function Networks“
- [3] Alex Pentland, Tanzeem Choudhury - „Face Recognition for Smart Environments“
- [4] Markus Hofmann - „Grundsätzliche Untersuchung von Bildverarbeitungsalgo. zur Gesichtererkennung“ (<http://www.markus-hofmann.de/>)