Gesicht- und Handerkennung

Michael Schnell

25. Juni 2005

Seminar: Gruppenaktionen in dynamischen, unsicheren Umgebungen
Abteilung für Grundlagen der Künstlichen Intelligenz
Institut für Informatik
Universität Freiburg
Sommersemester 2005

1 Einleitung

Diese Arbeit stellt die schriftliche Ausarbeitung eines Vortrags dar, der im Rahmen des Seminars *Gruppenaktionen in dynamischen, unsicheren Umgebungen* im Sommersemester 2005 gehalten wurde. Sie basiert auf den Artikeln *Roboust Real-time Object Detection* von Paul Viola und Michael Jones [**?**] und *Robust Hand Detection* [**?**] von Mathias Kölsch und Matthew Turk.

Paul Viola und Michael Jones haben ein Framework zur visuellen Erkennung von Objekten entwickelt, das eine sehr hohe Verarbeitungsgeschwindikeit und eine hohe Erkennungsleistung bietet. Im wesentlichen besteht ihre Arbeit aus drei Beiträgen. Als erstes stellen sie eine neue Art, Bilder zu repräsentieren, vor, welche sie „Integral Image“ nennen. Das Integral Image kann man sehr schnell auf bestimmte Merkmalen untersuchen. Das zweite ist ein Lernalgorithmus, welcher auf AdaBoost basiert. Aus einer großen Menge von Merkmalen sucht er sich nur wenige herraus, die das Objekt am besten erkennen. Der dritte Beitrag ist die Idee mehere Klassifizierer stufenweise hintereinder zu schalten. Dadurch kann ein schneller Klassifizierer eine große Anzahl der untersuchten Teilbilder verwerfen. So bleibt mehr Rechenleisung für weniger Teilbilder übrig. Viola und Jones präsentieren ihr System anhand einem Klassifizierer, der Gesichter erkennt. Ihre Implementation erreichte auf einem gewöhnlichen PC 15 Bilder pro Sekunde.

Anschließend wird noch eine Handerkennung von Mathias Kölsch und Matthew Turk vorgestellt, die auf dem Ansatz von Viola und Jones basiert. Sie stellen vor allem eine Schätzfunktion vor, mit deren Hilfe man schon im Vorfeld bestimmen kann, wie gut sich ein Objekt klassifizieren lässt.

2 Arbeit von Viola und Jones

Im folgenden wird die Arbeit von Viola und Jones erläutert.

2.1 Merkmale der Objekte

Die Objekterkennung von Viola und Jones untersucht sehr einfache Merkmale der Bilder, da dies deutlich schneller ist, als eine Objekterkennung, die direkt auf Pixelebene arbeitet. Die Art der Merkmale, die sie benutzen ist folgenderweise aufgebaut: Sie betrachten von meheren Rechtecken die Pixelsumme, also die Summe aller Grauwerte, und berechnen daraus eine Gesammtsumme bzw. Differenz, in die einige Rechtecksummen positiv und andere negativ eingehen. Viola und Jones beschränkten sich dabei auf drei solche Merkmale. Die Rechtecke jedes Merkmals sind dabei immer gleich lang und gleich breit.

Abbildung 1: A und B: Merkmale Typ 1, C: Typ 2, D: Typ 3

- Das einfachste Merkmal ist die Differenz von 2 Rechtecksummen. Die Rechtecke sind dabei nebeneinander oder übereinander angeordnet.
- Das zweite Merkmal eine Summe von 3 Rechtecken. Die Rechtecke sind dabei ebenfalls nebeneinander oder übereinander angeordnet. Die Summe des mittleren Rechtecks wird von der Summe der beiden äußeren Rechtecke abgezogen.
- Das dritte Merkmal ist die Differenz von je zwei Diagonal angeordneten Rechtecken. Es werden also vier Rechtecke betrachtet, von denen zwei positiv und zwei negativ in die Summe eingehen.

Bei einem Bild von 24×24 Pixel gibt es 45.396 solche Merkmale. Damit der Klassifikator effektiv arbeiten kann, müssen diese Merkmale effektiv berechnet werden können. Jedoch ist die Berechnung so vieler Pixelsummen, basierend auf den einzenen Grauwerten, zunächst recht aufwendig. Daher definieren Viola und Jones zunächst eine neue Repräsentation eines Bildes, in der diese Summen in konstanter Zeit berechnet werden können.

2.2 Integral Image

Das Integral Image enthält am Punkt (x, y) die Summe aller Pixelwerte aus dem Rechteck, das sich mit den Eckpunkten $(0, 0)$ und (x, y) beschreiben lässt. Also die Summe aller Pixelwerte links und oberhalb von (x, y), einschießlich x und y:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Wobei $i(x, y)$ der Grauwert des Originalbildes an der Pixelposition (x, y) ist. Dieses Integral Image kann in einem Durchlauf über alle Pixel des Original-

Abbildung 2: Sei $a = ii(a_x, a_y)$, $b = ii(b_x, b_y)$ usw. Dann lässt sich die Pixelsumme von D berechnen durch $d - b - c + a$

i:

1	4	5
3	2	3
0	1	2

s:

1	4	5
4	6	8
4	7	10

ii:

1	5	10
4	10	18
4	11	21

Abbildung 3: Beipiel: Links ist das Original Bild (Grauwerte). In der Mitte sind die Spaltensummen. Rechts ist dann das Integral Image. Man beachte, daß s und ii den Wertebereich von i überschreiten (können).

bildes berechnet werden:

$$s(x, y) = s(x, y - 1) + i(x, y)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

Mit $s(x, -1) = 0$ und $ii(-1, y) = 0$.
Dabei ist $s(x, y)$ die Summe der Pixel in der Spalte x von 0 bis y. Also $s(x, y) = \sum_{y' \leq y} i(x, y)$

Nun können, basierend auf dem Integral Image, die oben beschreibenen Merkmale sehr effizient berechnet werden. Siehe Abbildung 2.

Die verwendeten Merkmale sind sehr primitiv. So sind sie z.B. immer horizontal oder vertikal ausgerichtet. Sie lassen sich aber bei beliebiger Position und beliebiger Größe sehr schnell berechnen. Da die Merkmale einfach skaliert werden können spart man sich das skalieren des Bildes, was sehr viel aufwendiger ist.

2.3 Lernen mit AdaBoost

Obwohl die Merkmale sehr schnell berechnet werden können würde das berechnen aller 45.396 Merkmale bei einem Bild von nur 24×24 Pixeln doch etwas lange dauern. Jedoch ist die Hypothese von Viola und Jones, daß eine sehr kleine Teilmenge der Merkmale ausreicht, um einen guten Klassifikator zu erstellen. Die Hauptaufgabe besteht also darin, diese wichtigen Merkmale zu finden.

2.3.1 AdaBoost

AdaBoost ist eine Mögleichkeit die Erkennungsleistung eines Klassifikators zu verbessern. Dazu werden nicht einzelne Merkmale betrachtet, sondern mehere. Jedes dieser Merkmale bekommt ein Gewicht. Am Anfang sind alle Merkmale gleichgewichtet. Der Lernalgorithmus lernt Hypothesen anhand von Beispielen unter berücksichtigung der Gewichte. Ist ein schwach gewichtetes Merkmal nicht vorhanden, dafür aber ein stark gewichtetes, so entscheidet sich der Lernalgorithmus für positiv. Auch jede Hypothese bekommt ein Gewicht, abhängig davon, wie viele Merkmale vorhanden sind. AdaBoost bekommt neben den Beispielen und dem Lernalgorithmus auch noch eine obere Grenze M, die vorgibt, wieviele Hyphothsen erstellt werden sollen. Aus den M Hypothesen erstellt AdaBoost dann die endgültige Hypothese, also den Klassifikator, als gewichtete Kombination.

AdaBoost hat eine schöne theoretische Eigenschaft: Ist der gegebene Lernalgorithmus ein schwacher Lerner, das heißt er ist zumindest besser, als blind zu tippen, dann liefert AdaBoost für ausreichend große M einen Klassifikator, der die Übungsmenge perfekt klassifiziert.

2.3.2 Variante von Viola und Jones

Viola und Jones änderten den Algorithmus etwas ab, so daß er sowohl die Merkmale auswählt, als auch den Klassifikator erstellt. Dazu beschränken sie die schwachen Klassifikatoren auf ein einziges Merkmal. Der schwache Lerner sucht sich dann das Merkmal aus, das am besten die positiven von den negativen Beispielen trennt. Für jedes Merkmal bestimmt er einen Schwellwert, so daß die Anzahl der falsch klassifizierten Beispiele minimal ist. Ein schwacher Klassifikator $h_j(x)$ besteht dann aus einem Merkmal f_j, einem Schwellwert θ_j und einer Parität, welche die Richtung des Ungleichheitszeichens angiebt:

$$h_j(x) = \begin{cases} 1 & \text{wenn } p_j \cdot f_j(x) < p_j \cdot \theta_j \\ 0 & \text{sonst} \end{cases}$$

Wobei x hier ein 24×24 Pixel Bild ist. Abbildung 4 zeigt den Ablauf des Algortihmus.

In einem ersten Test erzeugten Viola und Jones einen Klassifikator aus 200 Merkmalen. Er hatte eine Erkennungsrate von 95% und er stufte nur ein Teilbild von 14084 fälschlicherweise Positiv ein (ein *false positiv*). Er brauchte nur 0,7 Sekunden für ein Bild der Größe 384×288. Die naheliegendste Möglichkeit, die Erkennungsleistung weiter zu verbessern, ist die Anzahl der Merkmale zu erhöhen. Damit steigt jedoch auch der Rechenaufwand.

Die ersten zwei Merkmale, die AdaBoost gewählt hat, lassen sich leicht nachvollziehen. Das erste Merkmal ist eines vom ersten Typ. Dabei sind zwei breite Rechtecke übereinander. Das eine über den Augen, das andere über Wangen, denn der bereich der Augen ist meist dunkler als der Bereich der Wangen. Das zweite Merkmal besteht aus drei Rechtecken, die nebeneinander angeordnet sind, so daß die zwei äußeren über den Augen liegen, und der mittlere über dem Bereich der Nase, welcher meist heller ist als der Bereich der Augen.

2.4 Hintereinanderschalten von Klassifikatoren

Diese Idee von Viola und Jones ist leicht zu verstehen. Die Anzahl der Teilbilder die vom Klassifikator untersucht werden müssen ist sehr groß. An jeder Position im Bild müssen Teilbilder verschiedener Größen untersucht werden damit das Objekt auch an verschiedenen Stellen im Bild in unterschiedlicher Größe auch erkannt wird. In jedem Teilbild könnte sich das gesuchte Objekt befinden. Tatsächlich ist es jedoch nur in einer sehr kleinen Anzahl von Teilbilder wirklich vorhanden.

Die Idee ist also alle Teilbilder zunächst durch einen sehr schnellen Klassifikator zu klassifizieren. Dieser Klassifikator muss so gebaut sein, daß er möglichst alle Objekte erkennt. Er sollte lieber zu viel erkennen, als zu wenig. Und trozdem wird er eine sehr große Anzahl an Teilbildern verwerfen. Teilbilder, die nicht verworfen werden, werden dann durch den nächsten etwas aufwendigeren Klassifikator geschickt, welcher ebenfalls wieder einige der negativen Teilbilder verwirft, usw. Abbildung 5 zeigt den schematischen Aufbau.

Eine höhere Erkennungsrate kann erreicht werden, wenn der Schwellwert des Klassifikators ($\frac{1}{2}\sum_{t=1}^{T}\alpha_t$, siehe Abbildung 4) kleiner gemacht wird. Dadurch werden mehr Objekte erkannt, aber auch es entsteht dadurch auch eine höhere Rate der falsch als positiv eingestuften Bilder (*false positiv rate*).

Auf diese Weise kann ein Klassifikator erstellt werden, der nur aus den zwei oben beschriebenen Merkmalen besteht, 100% der Gesichter erkennt und 60% der Teilbilder ohne Gesichter verwirft. Dies allein ist zwar noch

- Gegeben: Beispiele $(x_1, y_1), \cdots, (x_n, y_n)$.

  Mit $y_i = \begin{cases} 1 & \text{wenn } x_i \text{ positives Beispiel (enthält das ges. Objekt)} \\ 0 & \text{sonst} \end{cases}$

- Initialisiere die Gewichte $\omega_{1,i} = \begin{cases} \frac{1}{2l} & \text{wenn } x_i \text{ positives Beispiel} \\ \frac{1}{2m} & \text{sonst} \end{cases}$

  Wobei l die Anzahl der positiven Beispiele und m die Anhal der negativen Beispiele ist. $(n = m + l)$

- Für t aus $1, \cdots, T$:
  - Normalisiere die Gewichte: $\omega_{t,i} \leftarrow \frac{\omega_{t,i}}{\sum_{j=1}^{n} \omega_{t,j}}$
  - Für jedes Merkmal j
    - Trainiere einen Klassifikator h_j, welcher sich auf ein einziges Merkmal beschränkt.
    - Berechne den Fehler des Klassifikators als $\epsilon_j = \sum_i \omega_i |h_j(x_i) - y_i|$
  - Wähle den Klassifikator h_t mit dem kleinsten Fehler ϵ_t
  - Aktualisiere die Gewichte: $\omega_{t+1,i} = \omega_{t,i} \left(\frac{\epsilon_t}{1-\epsilon_t}\right)^{1-e_i}$

    Wobei $e_i = \begin{cases} 0 & \text{wenn } h_t(x_i) = y_i \\ 1 & \text{sonst} \end{cases}$

- Der endgültige Klassifikator ist:

  $h(x) = \begin{cases} 1 & \text{wenn } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=0}^{T} \alpha_t \\ 0 & \text{sonst} \end{cases}$

  Wobei $\alpha_t = \log \frac{1-\epsilon_t}{\epsilon_t}$ die Gewichte der schachen Klassifikatoren sind.

Abbildung 4: AdaBoost. Variante von Viola und Jones. T schwache Klassifikatoren mit nur einem Merkmal werden erstellt. Der endgültige starke Klassifikator ist eine Linearkombination der schwachen. Die Gewichte sind Antiproportional zu den Lernfehlern.

Abbildung 5: Schematischer Aufbau der Kaskade. Verwirft ein Klassifikator ein Teilbild, so wird es nicht mehr weiter betrachtet. Andernfalls wird das Teilbild an den nächsten Klassifikator weitergereicht.

kein akzeptabler Klassifikator, aber er reduziert die Anzahl der weiter zu betrachtenden Teilbilder sehr stark. Viola und Jones rechnen vor, daß der Klassifikator nur etwa 60 Prozessoroperationen braucht. Negative Teilbilder können so schon recht früh und schnell verworfen werden. Wohingegen positive Teilbilder jede Ebene der Kaskade durchlaufen müssen.

Klassifikatoren in tieferen Ebenen bekommen während der Lernphase nur die positiven Bilder, die den Klassifikator der vorherigen Ebene passiert haben. Daher lassen sich die Bilder nicht mehr anhand der Merkmale aus höheren Ebenen unterscheiden. Tieferliegende Klassifikatoren werden also automatisch andere Merkmale wählen. Auch werden sie mehr Merkmale brauchen, um die Restlichen Teilbilder zu Klassifizieren. Da sie aber auch später, während der Klassifikation, weniger Teilbilder zu bearbeiten haben, können sie auch mehr Rechenzeit beanspruchen.

2.4.1 Trainieren einer Kaskade

Die Erkennungsrate D des resultierenden Klassifikators lässt sich wie folgt berechnen:

$$D = \prod_{i=0}^{K} d_i$$

Dabei ist K die Anzahl der Klassifikatoren bzw. Ebenen der Kaskade und d_i ist die Erkennungsrate der einzelnen Klassifikatoren. Auf die gleiche Weise

lässt sich auch die *false positiv rate* berechen:

$$F = \prod_{i=0}^{K} f_i$$

Möchte man also einen Klassifikator mit einer Erkennungsrate von 90% bei 10 Ebenen erreichen, so müssen die einzelnen Klassifikatoren etwa eine Erkennungsrate von 99% haben ($0.99^{10} \approx 0,904$). Allerdings dürfen die Klassifikatoren auf den einzelnen Ebenen eine *false positiv rate* von 30% haben, damit der gesamte Klassifikator eine *false positiv rate* von etwa $0,3^{10} \approx 5,9 \cdot 10^{-6}$ erreicht.

Die erwartete Anzahl N der Merkmale, auf die ein Bild während der Klassifikation untersucht wird, berechnet sich wie folgt:

$$N = n_0 + \sum_{i=1}^{K} \left(n_i \sum_{j<i} p_i \right)$$

Wobei p_i die *positiv rate* und n_i die Anzahl der Merkmale des i-ten Klassifikators sind.

Viola und Jones weisen darauf hin, daß man die Minimierung von N prinzipiell als Optimierungsproblem in Anhängigkeit der Ebenen, der Merkmalsanzahl jeder Ebene und des Schwellwertes jeder Ebene formulieren könnte. Jedoch wäre das finden des Minimums viel zu aufwendig.

Stattdessen wählen sie folgende Vorgehensweise: Man wählt die Erkennungsrate d_i und die *false positiv rate* f_i für jede Ebene. In jeder Ebene wird ein Klassifikator mit AdaBoost trainiert. Dabei wird die Anzahl der Merkmale erhöht, bis die gewählten Raten erreicht werden. Diese werden durch eine Menge von Testbilder ermittelt. Ist die gewünschte *false positiv rate* des gesamten Klassifikators noch nicht erreicht wird eine weitere Ebene hinzugefügt. Siehe auch Abbildung 6

In einem Versuch trainierten Viola und Jones 2 Klassifikatoren. Einen monolitischen mit 200 Merkmalen und einen Kaskadierten mit Zehn mal 20 Merkmalen. Dabei nutzten sie für den monolitischen Klassifikator die gleichen Bilder, die in der Kaskade benutzt wurden. Denn die Wahl der negativen Beispiele für den monolitischen Klassifikator würde sich ohne die Kaskade, welche einfache Beispiele verwirft und nur die schweheren weiterreicht, schwierig gestalten.

Das Ergibnis des Vergleichs ergibt, daß beide Klassifikatoren etwa gleich gut sind. Der kaskadierte Klassifikator ist jedoch etwa 10 mal schneller.

- Gegeben: d (minimale Erkennungsrate pro Ebene), f (höchste zugelassene *false positiv rate* pro Ebene), F_{target} höchste *false positiv rate* des gesamten Kassifikators.
- P = Menge von positiven Beispielen, N = Menge von negativen Beispielen.
- $F_0 = 1.0;\ D_0 = 1.0;\quad i = 0$
- while $F_i > F_{target}$
  - $i \leftarrow i + 1;\quad n_i = 0;\quad F_i = F_{i-1}$
  - while $F_i > f \times F_{i-1}$
    - $n_i \leftarrow n_i + 1$ (Nehme ein Merkmal mehr)
    - Klassifikator mit n_i Merkmalen durch AdaBoost mit N und P trainieren
    - F_i und D_i der momentanen Kaskade bestimmen
    - Schwellwert des aktuellen Klassifikators heruntersetzen, bis die aktuelle Kaskade eine Erkennungsrate von mindestens $d \times D_{i-1}$ erreicht. (Beeinflusst auch F_i.)
  - $N \leftarrow \emptyset$
  - Falls $F_i > F_{target}$ so mache alle Bilder in N, die keine Gesichter enthalten, aber von der aktuellen Kaskade als positiv erkannt werden.

Abbildung 6: Algorithmus, mit dem eine Kaskade von Klassifikatoren trainiert wird.

2.5 Gesichtserkennung

2.5.1 Lernphase

Um die Klassifikatoren zu trainieren sammelten Viola und Jones 4916 Bilder aus dem Internet. Sie markierten die Gesichter von Hand und skalierten sie auf eine Größe von 24×24 Pixeln.

Die Gesichtserkennung bestand letztlich aus 32 Ebenen mit insgesamt 4297 Merkmalen.

Der erste Klassifikator benutzte zwei Merkmale, verwarf etwa 60% der Bilder ohne Gesichter und erkannte beinahe 100% der Gesichter. Es folgten drei Klassifikatoren mit 20 Merkmalen, dann zwei mit 50 Merkmalen fünf mit 100 Merkmalen und schließlich 20 mit 200 Merkmalen. Ihre Angaben sind vermutlich gerundet, denn sie ergeben in der Summe 4667 Merkmale. Die Anzahl der Ebenen und Merkmale ermittelten sie durch reines Ausprobieren.

Die ersten drei Klassifikatoren wurden mit den 4916 Gesichtern und 10.000 Teilbilder ohne Gesichter mit AdaBoost (siehe Abblidung 4) trainiert. Es wurden für jeden Klassifikator andere Teilbilder ohne Gesichter genommen, damit sie sich nicht die gleichen Merkmale aussuchen.

Für Tiefere Ebenen wurden die *false positives* der vorherigen Ebenen und bis zu 6000 Teilbilder ohne Gesichter benutzt.

Die Trainingszeit des gesamten Kassifikators dauerte Wochen auf einer enizelnen 466 MHz AlphaStation XP900.

2.5.2 Der kaskadierte Klassifikator

Die Geschwindigkeit des Klassifikators hängt stark von der Anzahl der Merkmale ab, auf die ein Bild letztlich getestet wird. Diese wiederum hängen von den Bildern ab. Die Teilbilder aus der Bildermenge vom MIT+CMU wurden im Durchschnitt auf nur acht der 4297 Merkmalen getestet. Auf einem Pentium III mit 700 MHz konnte der Klassifikator ein 384×288 Pixelbild in ungefähr 0,067 Sekunden klassifizieren. Damit ist er etwa 15 mal schneller als die Gesichtserkennung von Rowley-Baluja-Kanade und etwa 600 mal schneller als die von Schneiderman-Kanade.

Zur Klassifikation wandert ein kleines Fenster, anfangs 24×24 Pixel, in kleinen Schritten über das Bild. Dabei entstehen die zu Untersuchenden Teilbilder. Das Fenster wird in jedem Durchlauf um einen Skalierungsfaktor s vergrößert. Gute Ergebnisse brachte der Faktor $s = 1,25$. Des Fenster wird dann mit einer Schrittgröße von $[s \cdot \Delta]$ über das Bild bewegt. Dabei ist s der Skalierungsfaktor Δ die Schrittgröße und [ ] der Rundungsoperator.

Bei der Bildermenge von MIT+CMU, welche aus 130 Bildern mit insgesamt 507 Gesichtern besteht, ergibt sich dabei für $\Delta = 1.0$, $s = 1,25$ und

einer Startskalierung von $1,0$ eine Anzahl von 75.081.800 Teilbildern.

Da der Klassifikator auf kleine Änderungen in der Skalierung und der Verschiebung unempfindlich ist, wird er das gleiche Objekt mehrmals erkennen. Bereiche in denen Gesichter erkannt wurden können sich also stark überschneiden. Viola und Jones verschmelzen daher alle Überschneidungen zu einem einzigen Treffer.

3 Handerkennung

Basierend auf der Arbeit von Viola und Jones erstellten Mathias Kölsch und Matthew Turk einen Handdetektor. Sie forschen auf dem Gebiet der Gestensteuerung, basierend auf Videobildern. Um dieses Ziel zu erreichen brauchen sie einen Klassifikator, welcher unabhängig vom Hintergrund, von der Beleuchtung, von der Person und von der Kamera ist. Um die geeignetsten Handstellungen herrauszubekommen könnte man jede Handstellung von einem Klassifikator trainieren lassen. Dies würde aber zu lange dauern. Daher versuchen Kölsch und Turk die Unterscheidbarkeit schon im Vorfeld zu analysieren.

3.1 Unterscheidbarkeitsabschätzung durch Analyse des Frequenzspektrums

Kölsch und Turk untersuchten dazu 8 verschiedene Handstellungen, die sie aussuchten, da sie unterschiedlich aussehen und leicht durchzuführen sind. Von bis zu Zehn ähnlichen Bildern einer Handstellung berechneten sie je ein Durchschnittsbild, angepasst auf 25×25 Pixel. Höhere Frequenzen einer Fouriertransformation geben den Betrag der Abweichung in den Grauwerten wieder.

$$F(u,v) = \frac{1}{25 \cdot 25} \sum_{m=0}^{24} \sum_{n=0}^{24} I(m,n) e^{-i2\pi(\frac{mu}{25}+\frac{nv}{25})}$$

Da die Bilder jedoch endlich sind, produziert die Fouriertransformation starke künstliche Frequenzen. Daher subtrahierten Kölsch und Turk die Fouriertransformation P eines Bildes in einem Einheitsgrau.

$$P(u,v) = \frac{1}{25 \cdot 25} \sum_{m=0}^{24} \sum_{n=0}^{24} \frac{1}{2} e^{-i2\pi(\frac{mu}{25}+\frac{nv}{25})}$$

Dadurch erhielten sie eine Differenzentransformation D frei von Artefakten.

$$D(u,v) = \log |F(u,v) - P(u,v)|$$

Abbildung 7: Die untersuchten Handstellungen. Erste Zeile der Name, zweite Zeile ein Beispielbild mit 24×24 Pixel. Dritte Zeile der s-Wert. Letzte Zeile die Fouriertranformation

Schlussendlich berechneten sie die gesuchte Schätzfunktion s als normierte Summe aller Amplituden.

$$s = e^{\frac{1}{k} \cdot \sum_{u,v} D(u,v)}$$

Als Ergebnis ihrer Untersuchung kam herraus, daß die Handstellung *closed* (offene Hand mit gestreckten Fingern, die alle ohne Lücke nebeneinander sind) die höchste Variation der Grauwerte. Die Handstellung *fist* (Handrücken einer Faust) dagegen die geringste Variation.

3.2 Nichtkaskadierter Klassifikator

Um die sechs Klassifikatoren zu trainieren, machten Kölsch und Turk 2300 Bilder von Händen unter den verschiedensten Gegebenheiten (Hintergrund, Person, Licht etc.). Sie skalierten und drehten die Bilder, so daß jede Hand 25×25 Pixel einnahm. Die eine Hälfte der Bilder wurden zum Trainieren benutzt, die andere zum Testen.

Auch Kölsch und Turk bauten zunächst einen nichtkaskadierten Klassifikator. Sie trainierten ihn mit den oben genannten Bildern und 11.500 Bildern ohne Hände. Weitere 11.500 Bilder ohne Hände waren wieder für den Test bestimmt. Im resultierenden Klassifikator fanden sie ihre Schätzfunktion nur teilweise bestätigt. Die Handstellung *closed* lies sich am besten erkennen, wie es die Schätzfunktion auch vorraussagte. Jedoch lies sich entgegen der Annahmen *Lpalm* schlechter erkennen als *Lpack*, obwohl es durch die sichtbaren Finger mehr Struktur hat. Sie zeigen jedoch, daß die Schätzfunktion beim Kaskadierten Klassifikator, der auch ein zusätzliches Merkmal benutzt, besser zutrifft.

3.3 Kaskadierter Klassifikator

Um ihre endgültige Handerkennung zu bauen erweiterten Kölsch und Turk die Merkmalstypen. Zusätzlich zu denen, die Viola und Jones verwendeten, benutzten sie noch einen vierten Typ, bei dem vier Rechtecke betrachtet werden. Diese können zur Trainingszeit beliebig relativ zueinander stehen. Sie können sich sogar überlappen. Siehe Abbildung **??**a.

Betrachtet man die ROC-Kurve (Abb. **??**b) des resultierenden Klassifikators im linken Bereich, also dem Bereich, in dem die false positv rate sehr klein ist, so stimmen die Vorhersagen der Schätzfunktion besser als bei dem nicht kaskadierten Klassifikator. Die Handstellung *closed* ist wieder die, welche am besten erkannt wird, *sidepoint* ist die Zweitbeste. Und wie von der Schätzfunktion vorrausgesagt, ist *Lpalm* besser als *Lback* (Siehe Abb. **??**).

a) b)

Abbildung 8: a) Beispiel zu dem neuen Merkmal, das Kölsch und Turk zusätzlich verwendeten. b) ROC-Kurve für die untersuchten Handstellungen.

Für die endgültige Handerkennung entschieden sich Kölsch und Turk für die Handstellung *closed*. Bei einer Erkennungsrate von 92,23% hat der Klassifikator eine *false positiv rate* von $1,01 \cdot 10^{-8}$ auf ihrer Testmenge. Das ist bei 279 VGA Bildern (mit jeweils 355.614 Teilbildern) ein einziger falscher positiver Treffer.

Mit anderen Parametern lässt sich aber auch bei einer Erkennungsrate von 65,80% erreichen, daß gar kein *false positiv* mehr auftritt. Bei der hohen Geschwindigkeit des Klassifikators wird die Hand trotz geringerer Erkennungsrate in meheren aufeinanderfolgenden Bildern zuverlässig erkannt.

4 Zusammenfassung

Viola und Jones entwickelten einen neuen Klassifikator mit hoher Erkennungsrate und geringer Rechenlast. Als Beispiel konstuierten sie eine Gesichtserkennung, die etwa 15 mal schneller ist, als bisherige Ansätze.

Ihre erste Idee ist eine neue Repräsentation der Bilder, die sie Integral Image nannten. Die von ihnen gewählten Merkmale lassen sich damit an jeder Position zu jeder Skalierung extrem schnell berechnen. Dadurch spart man sich das deutlich aufwendigere vorskalieren des Bildes. Im bezug auf die Gesichtserkennung war der Klassifikator schon fertig, noch bevor man das

Bild in alle nötigen Größen skaliert hätte.

Ihr zweiter Beitrag ist ein Algorithmus zur Auswahl der Merkmale. Er basiert auf AdaBoost. Da er sehr effektiv ist kann, man ihm eine sehr große Anzahl an komplexen Merkmalen als Eingabe geben. Der resultierende Klassifikator ist trotzdem sehr schnell, da er nur eine kleine Anzahl an Merkmalen zu untersuchen hat. Mit einer größeren Anzahl an komplexen Merkmalen als Eingabe an AdaBoost steigt auch die Wahrscheinlichkeit, daß die AdaBoost Variante ein paar wenige sehr gut passende Merkmale findet.

Als drittes stellten Viola und Jones eine Technik vor Klassifikatoren zu kaskadieren. Die ersten Ebenen verwerfen schon einen Großteil aller Negativen, so daß tiefere Ebenen mehr Rechenzeit auf interessantere Bildausschnitte verwenden können. Viola und Jones heben hervor, daß ihr Ansatz sehr einfach aufgebaut ist, da jede Ebene gleich ist. Dadurch lässt er sich leicht verstehen und implementieren und hat den Vorteil, daß man einfache Kompromisse zwischen Erkennungsleistung und Rechenaufwand machen kann.

Kölsch und Turk ergänzten die Merkmale um einen weiteren Typ. Dieser ist deutlich aussagekräftiger. Sie heben auch ihre Schätzfunktion hervor, die es erlaubt, im Vorfeld zu bestimmen, welche Objekte, in diesem Fall Handstelungen, sich zum erkennen besser eignen als andere.

Insgesamt lässt sich sagen, daß das Verfahren von Viola und Jones gut geeignet ist um Objekte zu erkennen. Kölsch und Turk zeigten zudem, daß es sich noch etwas verbessern und leicht auf andere Objekte übertragen lässt.

Literatur

[1] Paul Viola und Michael Jones (2001). Robust Real-time Object Detection. In *Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Lerning, Computing and Sampling*, Cambridge.

[2] Mathias Kölsch und Matthew Turk, Robust Hand Detection. University of California, Santa Barbara.