

Bewegungserkennung in Bildsequenzen mit dem Hidden Markov Modell

Axel Thümmler
Dortmund

September 1995

Zusammenfassung

Dieser Bericht wurde im Rahmen der PG277 mit dem Thema 'Ein ergonomisches Dialogsystem zur Steuerung von technischen Systemen in Wohnbereichen mittels Gestenerkennung' geschrieben. Im wesentlichen handelt er von der Benutzung des Hidden Markov Modells, kurz HMMs, zur Erkennung von Bewegungen in Bildsequenzen. Die Bildsequenzen werden dabei in eine Merkmalvektorsequenz umgewandelt. Jedem Bild der Sequenz wird also ein Vektor mit spezifischen Merkmalen des Bildes zugeordnet. Um bestimmte Kategorien von Bildsequenzen zu erkennen müssen die HMMs, für jede Kategorie einer, mit den Sequenzen trainiert werden. Es wird ihnen dabei die jeweilige Sequenz vorgespielt und das HMM lernt sie zu erkennen, indem er seine Parameter optimiert. Soll eine Sequenz einer bestimmten Kategorie dann wiedererkannt werden, wird die Kategorie des HMMs gewählt, der auf die Sequenz am besten zugeschnitten ist. In experimentellen Ergebnissen mit realen Bildsequenzen wurden Wiedererkennungsraten von über 90% erreicht. Weiter wurde die Wiedererkennungsraten verbessert, wenn die Anzahl der Personen, mit der die HMMs trainiert wurden, erhöht wurde. Das zeigt die Möglichkeit auf, mit dieser Methode ein personenunabhängiges Verfahren zur Erkennung von bestimmten Bewegungen einzurichten.

1 Einleitung

In den vergangenen Jahren haben Themen, die mit Bewegung zu tun haben, im Bereich der Computervisualisierung an Bedeutung gewonnen. Außer bewegten Objekten in wirklichen Szenen ist es sehr wichtig geworden, auch Personen zu erkennen. Algorithmen, die Personen oder Bewegungen von Personen erkennen, spielen eine wichtige Rolle bei der Realisierung von automatischen Überwachungs-systemen für verschiedene Anwendungen, wie z.B. das Auffinden von Ladendieben in Einkaufsmärkten oder das Erkennen gefährlichen Verhaltens in Kindergärten. Dieser Bericht stellt eine neue Methode der Bewegungserkennung in Bildsequenzen vor. Jede beobachtete menschliche Bewegung kann einer Kategorie zugeordnet werden. So ist jede für die Anwendung wichtige Bewegung, also die Bewegungen, die erkannt werden sollen, eine Kategorie.

Die derzeit existierenden Verfahren, die in der Bewegungserkennung gebraucht werden, sind top-down Methoden, die auf einer geometrischen Körperrekonstruktion basieren und bottom-up Methoden, die auf low-level Bildmerkmalen basieren. Die meisten Systeme, die mit top-down Methoden arbeiten, setzen ein geometrisches Modell des Körpers ein. So werden die menschlichen Körperteile mit leicht beschreibbaren Objekten wie Zylindern oder Quadriken usw. dargestellt. Das bedeutet, die Körperhaltung der Personen wird aus den Bildern extrahiert, die einzelnen Körperteile werden durch geometrische Formen dargestellt und anschließend wird der Körper mit den geometrischen Formen wieder rekonstruiert. Man erhält so den gesamten Körper durch geometrische Formen zusammengesetzt. Diese Methode ist allerdings für wirkliche Bilder weder robust noch zuverlässig, weil die Bilder für gewöhnlich zu unruhig sind, um in ein einfaches Schema gepreßt zu werden. Es ist somit sehr schwierig eine gute Repräsentation eines Bildes auf einer höheren Ebene mit geometrischen Formen zu erhalten. Die Rekonstruktion des Körpers ist ein wichtiger Punkt, den es bei dieser Methode zu verbessern gilt, weil Fehler in der Rekonstruktion erfolgreiche Wiedererkennung von Bewegungen verhindern.

Die Frage ist allerdings ob eine Rekonstruktion überhaupt notwendig ist für die Bewegungserkennung? Unser Ziel ist, Bewegungen wiederzuerkennen und nicht eine geometrische Repräsentation des Körpers zu erhalten. Somit konzentrieren wir uns auf die Erkennung von Bewegungen und nutzen dazu low-level Bildmerkmale in einer bottom-up Methode aus. Im allgemeinen beschreiben low-level Merkmale ein Bild nicht so gut wie eine Repräsentation durch geometrische Formen. Allerdings ist es einfacher, sie aus den Bildern zu extrahieren. Ein anderes Problem mit low-level Merkmalen ist, daß man mit ihnen eine Bewegungskategorie nicht so gut beschreiben kann wie mit einer high-level Repräsentation, die auf geometrischen Formen basiert. Das liegt daran, daß keine explizite Beziehung zwischen der low-level Repräsentation der Bilder und der auf einer höheren Ebene liegenden Beschreibung einer Kategorie besteht. Weiterhin war es mit low-level Merkmalen bisher nur möglich die Anzahl von Leuten in einem Bild zu zählen, aber nicht deren Bewegung zu erkennen.

Es gibt zwei grundlegende Probleme bei der Verwendung einer bottom-up Methode. Wie können wir es dem System möglich machen, kompliziertere Bewegungen zu erkennen? Wie können wir die Bewegungskategorien beschreiben ohne eine high-level Repräsentation der Bilder?

Diese Probleme können mit einer Lern-Prozedure, die low-level Merkmale gebraucht, gelöst werden. Unser Ziel ist Bildsequenzen, die in low-level Merkmalsequenzen umgewandelt wurden, in Bewegungskategorien einzuteilen. Um diese Lernfähigkeit zu realisieren setzen wir das Hidden Markov Modell, das mit zeitabhängigen Daten umgehen kann, ein.

In Abschnitt 2 dieses Berichts wird zuerst auf die Grundlagen der Wahrscheinlichkeitsrechnung in kurzer Form eingegangen, damit die nachfolgende Erklärung des Hidden Markov Modells und seiner Algorithmen besser verständlich wird. Danach wird in Abschnitt 3 beschrieben wie das Hidden Markov Modell speziell für die Bewegungserkennung genutzt werden kann. In Abschnitt 4 werden dann abschließend Ergebnisse, die aus Experimenten mit dem Modell stammen, dargelegt.

2 Das Hidden Markov Modell

Ein Hidden Markov Modell (HMM) ist eine Art stochastisches Zustandsübergangmodell. HMMs machen es möglich, mit zeitabhängigen Daten umzugehen, und sie sind bekannt für ihre Lernfähigkeit, die dadurch erreicht wird, daß man dem HMM zeitabhängige Daten vorführt und das Modell sich automatisch optimiert und an die Daten anpaßt. Hidden Markov Modelle wurden vor kurzem mit Erfolg in der Spracherkennung [1] eingesetzt.

2.1 Einführung in die Wahrscheinlichkeitsrechnung

In dieser kurzen Einführung in die Wahrscheinlichkeitsrechnung soll kein Anspruch auf Vollständigkeit erhoben werden, sondern lediglich eine kurze Erinnerung gegeben werden. Für Interessierte an der Stochastik seien die Bücher [4, 5, 6] empfohlen.

Man stelle sich ein Glücksrad mit dem Umfang 1 vor, daß in n verschieden große Bereiche (Kuchenstücke) eingeteilt ist. Die Bereiche seien mit $\omega_1, \omega_2, \dots, \omega_n$ bezeichnet. Zu jedem Bereich ω_i gehört eine Bogenlänge p_i .

$$p_i \geq 0, \quad \sum_{i=1}^n p_i = 1 \quad (1)$$

Einmaliges Drehen des Glücksrades ist ein *einstufiger Zufallsversuch*. Die Menge $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ aller möglichen Ausfälle nennt man *Ergebnisraum*. Ω enthält für unsere Zwecke endlich viele Elemente. Man nennt p_i die *Wahrscheinlichkeit* des Ausfalls ω_i . Der Vektor $p = (p_1, p_2, \dots, p_n)$, dessen Komponenten (1) erfüllen, heißt *Wahrscheinlichkeitsverteilung* auf Ω . Man kann p auch als Abbildung auffassen, die jedem ω die zugehörige Wahrscheinlichkeit zuordnet. Es wird dann geschrieben $P(\omega_i) = p_i$. Das Paar (Ω, p) heißt *diskreter Wahrscheinlichkeitsraum*.

Beispiel 1: Für den Wurf mit einem Würfel ist $\{1, 2, \dots, 6\}$ ein geeigneter Ergebnisraum. Die "Homogenität" des Würfels spiegelt sich in unserem Modell in $p_1 = p_2 = \dots = p_6$ wider, woraus wegen (1) folgt $p_i = \frac{1}{6}$ für $i = 1, 2, \dots, 6$.

Jede Teilmenge A von Ω nennt man ein *Ereignis*. Liefert eine Drehung den Ausfall $\omega \in A$, so sagt man, das Ereignis A sei eingetroffen. Wir wollen jedem Ereignis $A \subseteq \Omega$ wie folgt eine Wahrscheinlichkeit $P(A)$ zuordnen:

$$P(A) = \sum_{\omega_i \in A} p_i \quad (2)$$

Das Modell aus Beispiel 1 für den Wurf mit einem Würfel ist ein Beispiel für die sogenannte *Gleichverteilung*. Sie ist als diejenige Wahrscheinlichkeitsverteilung in Ω definiert, die jedem Elementarereignis, also jedem Ereignis mit einem Element, dieselbe Wahrscheinlichkeit zuordnet. Nach (1) ist also $p_i = \frac{1}{|\Omega|}$ für alle $i = 1, 2, \dots, n$. Hieraus und aus (2) folgt

$$P(A) = \frac{|A|}{|\Omega|} \text{ für alle } A \subseteq \Omega \quad (3)$$

Als nächstes wollen wir uns den bedingten Wahrscheinlichkeiten widmen, auf die es uns hier besonders ankommt. Betrachten wird zunächst wieder die Gleichverteilung, hier mit P bezeichnet, in einem endlichen Ergebnisraum Ω . Es seien A und B Ereignisse und $A \neq \emptyset$. Wir fragen nach einer sinnvollen Definition der Wahrscheinlichkeit von B unter der Bedingung, daß A eintritt oder, wie man auch kurz sagt, "bei gegebenem A ". Wir haben, daß das Ereignis A eingetreten ist. Das Ergebnis der bedingten Wahrscheinlichkeit muß also aus A sein. Wir können uns vorstellen, A jetzt als neuen Ergebnisraum zu sehen, unter dem B eintritt. Mit A als Ergebnisraum kann B aber nur noch eintreten wenn $A \cap B$ eintritt. Daher wird die "bedingte" Wahrscheinlichkeit von B unter der Bedingung A sinnvollerweise wie folgt erklärt:

$$\frac{|A \cap B|}{|A|} = \frac{|A \cap B| |\Omega|}{|\Omega| |A|} = \frac{P(A \cap B)}{P(A)} \quad (4)$$

Diese Überlegung legt die folgende für beliebige Wahrscheinlichkeitsräume gültige Definition nahe.

Definition: Ist (Ω, P) ein diskreter Wahrscheinlichkeitsraum und A ein Ereignis mit $P(A) > 0$, so heißt

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad (5)$$

die *bedingte Wahrscheinlichkeit* des Ereignisses B unter der *Bedingung* A .

Beispiel 2: Beim Würfeln mit einem Würfel ist die Wahrscheinlichkeit einer 6 gleich $\frac{1}{6}$. Unter der Annahme, daß eine gerade Zahl fällt, ist es wohl intuitiv klar, daß die Zahlen 1, 3 und 5 mit der Wahrscheinlichkeit 0 und die Zahlen 2, 4 und 6 mit gleicher Wahrscheinlichkeit, nämlich $\frac{1}{3}$, erscheinen. Dies steht im Einklang mit (4) und (5) für $A = \{2, 4, 6\}$ und $B = \{6\}$.

Beispiel 3: Beim Wurf mit einem roten und einem schwarzen Würfel ist die Wahrscheinlichkeit dafür, daß der schwarze Würfel eine 6 zeigt, gleich $\frac{1}{6}$. Unter der Bedingung, die Augensumme sei gleich 12, müssen beide Würfel eine 6 zeigen, und daher leuchtet es ein, daß die bedingte Wahrscheinlichkeit für das Würfeln einer 6 mit dem schwarzen Würfel unter dieser Bedingung gleich 1 ist. Dies ergibt sich wieder aus (4) und (5), wenn man $\Omega = \{(i, k) | 1 \leq i, k \leq 6\} = \{1, \dots, 6\}^2$ setzt, für P die Gleichverteilung auf Ω nimmt und $A = \{(6, 6)\}$, $B = \{1, \dots, 6\} \times \{6\}$ definiert. Dagegen wird die Bedingung, die Augensumme sei gleich 11, durch die Menge $A = \{(5, 6), (6, 5)\}$ dargestellt, und (4) liefert dann

$$P(B|A) = \frac{\frac{1}{36}}{\frac{2}{36}} = \frac{1}{2}$$

Bedingte Wahrscheinlichkeiten spielen eine wichtige Rolle in der Konstruktion und Berechnung von Wahrscheinlichkeiten. Häufig kennt man z.B. von der Struktur des Problems her sowohl $P(A)$ als auch $P(B|A)$. Daraus leitet man dann $P(A \cap B)$ mittels Gleichung (5) ab. $P(A \cap B)$ werde ich in den folgenden Kapiteln kurz mit $P(A, B)$ bezeichnen. Es ist die Wahrscheinlichkeit dafür, daß A und B zugleich eintreten.

2.2 Definition des Hidden Markov Modells

Um das Konzept eines Hidden Markov Modells (HMMs) zu verstehen betrachten wir ein Beispiel, das die Darstellung vereinfacht. Eine Person führt ein Experiment hinter einem Vorhang aus. Es befinden sich dort $N = 3$ Urnen und jede von ihnen beinhaltet eine große Anzahl farbiger Bälle. Es sind $M = 5$ verschiedene Farben. Eine erste Urne wird zufällig ausgewählt und aus ihr wird irgendein Ball zufällig herausgenommen. Dann wird der ausgewählte Ball vor den Vorhang gehalten und eine zweite Person notiert sich die Farbe. Anschließend wird der Ball wieder in die gleiche Urne zurückgelegt und eine neue Urne wird nach einem Zufallsprinzip bezogen auf die aktuelle Urne ausgewählt. Der Ballauswahlvorgang wird von der neuen Urne aus wiederholt. Dieses Experiment erzeugt eine endliche Sequenz von farbigen Bällen, die sich die Person vor dem Vorhang notiert hat. Zu bemerken ist, daß vor dem Vorhang nur die Folge der farbigen Bälle zu erkennen ist, nicht etwa die Folge der Urnen aus denen sie kommen. Dieses einfache Experiment besitzt schon die Eigenschaften, die mit einem HMM verbunden werden: Es ist ein erzeugender Mechanismus der eine Ausgabesequenz erstellt, und der Mechanismus ist ein stochastischer Prozess mit einer verborgenen (hidden) Komponente. Was uns interessiert ist, wie man ein stochastische Modell erstellen kann, das den Vorgang hinter dem Vorhang beschreibt. Ein HMM kann formal auf die folgende Weise beschrieben werden:

$T =$ Länge der Ausgabesequenz $O = O_1, O_2, \dots, O_T$ (Anzahl der farbigen Bälle die vor den Vorhang gehalten wurden)

$N =$ Anzahl der Zustände (Anzahl der Urnen)

$M =$ Anzahl verschiedener Ausgabesymbole (Anzahl der Farben)

$S = \{1, 2, \dots, N\}$ Menge mit N verschiedenen Zuständen, wobei $s_t = i$ bedeutet, daß zum Zeitpunkt t ($1 \leq t \leq T$) der Ausgabe der Zustand $i \in S$ angenommen wird.

$V = \{v_1, v_2, \dots, v_M\}$, Menge der möglichen Ausgabesymbole. $O_t \in V$ ($1 \leq t \leq T$) ist eines dieser Symbole.

$A = \{a_{ij} | a_{ij} = P(s_{t+1} = j | s_t = i)\}$, für alle Zustände i und j . A ist eine Zustandsübergangsmenge, in der steht, mit welcher Wahrscheinlichkeit von Zustand i nach Zustand j gegangen wird.

$B = \{b_j(k) | b_j(k) = P(v_k | s_t = j)\}$, für alle Zustände j und alle Ausgabesymbole v_k . $b_j(k)$ gibt die Wahrscheinlichkeit dafür an, daß im Zustand j Symbol v_k ausgegeben wird.

$\pi = \{\pi_i | \pi_i = P(s_1 = i)\}$, für alle Zustände i . π_i gibt die Wahrscheinlichkeit dafür an, daß Zustand i als Anfangszustand gewählt wird.

Ein HMM wird repräsentiert durch die kompakte Notation $\lambda = (A, B, \pi)$. Um ein HMM eindeutig zu bestimmen ist es notwendig, die Anzahl der Zustände N , die Anzahl der Ausgabesymbole M sowie die drei Wahrscheinlichkeitsmatrizen A , B und π festzulegen. In der Matrix A stehen die Übergangswahrscheinlichkeiten von jedem Zustand zu jedem anderen. A ist also eine $N \times N$ Matrix. Matrix B enthält für jeden Zustand die Wahrscheinlichkeit für jedes Ausgabesymbol. Sie ist also eine $N \times M$ Matrix. π ist eine

$N \times 1$ Matrix in der für jeden Zustand die Wahrscheinlichkeit, daß er Anfangszustand ist, enthalten ist. Zur Wohldefiniertheit des HMMs muß noch folgende, für die Wahrscheinlichkeitsverteilung notwendige, Beziehung gelten:

$$\sum_{j=1}^N a_{ij} = \sum_{k=1}^M b_i(k) = \sum_{k=1}^N \pi_k = 1, \quad 1 \leq i \leq N \quad (6)$$

Zusätzlich zu den obigen Definitionen kann man noch eine Menge S_I mit Anfangszuständen und eine Menge S_F mit Endzuständen definieren. Ist ein Zustand i nicht in S_I enthalten, so ist natürlich $\pi_i = 0$. N_I und N_F gibt die Anzahl der Anfangszustände bzw. der Endzustände an. Mit dem Modell wird eine Ausgabesequenz $O = O_1, O_2, \dots, O_T$ wie folgt generiert:

1. Wähle einen Startzustand s_1 bezogen auf die Wahrscheinlichkeitsverteilung π .
2. Setze $t := 1$
3. Wähle ein Ausgabesymbol $O_t := v_k$ bezogen auf die Wahrscheinlichkeitsverteilung $b_{s_t}(k)$, $k = 1, 2, \dots, M$.
4. Wähle einen neuen Zustand $s_{t+1} := i$ bezogen auf die Wahrscheinlichkeitsverteilung $a_{s_t i}$, $i = 1, 2, \dots, N$.
5. Setze $t := t + 1$ und gehe zu Schritt 3 falls $t < T$ sonst beende das Verfahren.

Es werden zwei grundlegende Voraussetzungen an das HMM gestellt. Zum einen ist das die Markovsche Voraussetzung, d.h., daß zu jedem Ausgabezeitpunkt t ein neuer Zustand nach einer Übergangswahrscheinlichkeit angewählt wird, die nur von dem aktuellen Zustand abhängt. Es ist zu beachten, daß ein Übergang von einem Zustand zu sich selbst möglich ist. Die zweite Voraussetzung ist die Ausgabe-Unabhängigkeitsvoraussetzung, d.h., daß die Ausgabewahrscheinlichkeit für ein Symbol nur von dem aktuellen Zustand abhängt, also unabhängig davon ist, wann und wie dieser erreicht wurde. Diese Voraussetzungen reduzieren die Anzahl der freien Parameter in einem HMM und machen somit Lern- und Wiedererkennalgorithmen effizienter.

2.3 Grundlegende Algorithmen für HMMs

Hat man ein HMM wie im vorigen Abschnitt gegeben, so sind drei Hauptprobleme, die es zu lösen gilt, von Interesse. Die Probleme sind die folgenden:

1. Hat man eine Ausgabesequenz $O = O_1, O_2, \dots, O_T$ und ein HMM $\lambda = (A, B, \pi)$ gegeben, ist die Frage wie man $P(O|\lambda)$, die Wahrscheinlichkeit dafür, daß O von λ erzeugt wird, berechnet. Das Problem kann man auch wie folgt sehen: Hat man mehrere konkurrierende HMMs und eine Sequenz O , wie findet man das HMM, das die Sequenz mit größter Wahrscheinlichkeit erzeugt. Man will damit diese Sequenz wiedererkennen. Eine Lösung des Problems ist als *forward-* bzw. *backward-Algorithmus* bekannt.

2. Man hat eine Ausgabesequenz O und ein HMM λ gegeben. Wie können wir die Modellparameter (A, B, π) berichtigen, sodaß die Sequenz O mit einer größtmöglichen Wahrscheinlichkeit $P(O|\lambda)$ erzeugt wird? Mit anderen Worten, wir wollen λ verändern, um $P(O|\lambda)$ zu maximieren. Wir wollen damit erreichen, daß λ möglichst gut auf O zugeschnitten wird. Diesen Prozeß kann man auch als Lernen bezeichnen und eine Lösung ist als *Baum-Welch-Neueinschätzung* bekannt.
3. Hat man eine Ausgabesequenz O und ein Modell λ gegeben, so möchte man wissen, was die wahrscheinlichste Zustandsfolge $S = s_1, s_2, \dots, s_T$ ist, die durchlaufen wird um O auszugeben. Es wird quasi der verborgene Teil des HMMs aufgedeckt. Ein solcher Algorithmus ist als *Viterbi-Algorithmus*[1, 2] bekannt. Auf ihn soll aber hier nicht näher eingegangen werden, da er für unsere Zwecke nicht von Interesse ist.

2.3.1 Forward-Backward-Algorithmus

Um die Wahrscheinlichkeit dafür, daß die Ausgabesequenz O vom HMM λ erzeugt wird zu bestimmen, ist es das Naheliegendste, die Wahrscheinlichkeiten aller möglichen Zustandsfolgen der Länge T , die zu O führen, aufzusummieren. Für jede feste Zustandsfolge $S = s_1, s_2, \dots, s_T$ ist die Wahrscheinlichkeit dafür, daß λ die Ausgabesequenz O erzeugt $P(O|S, \lambda)$ wobei

$$P(O|S, \lambda) = b_{s_1}(O_1)b_{s_2}(O_2) \cdots b_{s_T}(O_T) \quad (7)$$

Andererseits ist die Wahrscheinlichkeit dafür, daß λ die Zustandsfolge S durchläuft

$$P(S|\lambda) = \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} \cdots a_{s_{T-1} s_T} = a_{s_0 s_1} a_{s_1 s_2} a_{s_2 s_3} \cdots a_{s_{T-1} s_T} \quad (8)$$

wobei $a_{s_0 s_1}$ der Einfachheit halber auf π_{s_1} gesetzt wird.

Die vereinte Wahrscheinlichkeit von O und S , d.h. die Wahrscheinlichkeit, daß O und S zur gleichen Zeit erscheinen, ist nach Definition der bedingten Wahrscheinlichkeit (5) einfach das Produkt der beiden oberen Terme.

$$P(O, S|\lambda) = P(O|S, \lambda)P(S|\lambda) \quad (9)$$

Die Wahrscheinlichkeit $P(O|\lambda)$ ist dann die Summation der Gleichung (9) über alle möglichen Zustandsfolgen:

$$P(O|\lambda) = \sum_{\text{alle } S} P(O|S, \lambda)P(S|\lambda) = \sum_{\text{alle } S} \prod_{t=1}^T a_{s_{t-1} s_t} b_{s_t}(O_t) \quad (10)$$

In Gleichung (10) kann man sehen, daß der Anfangszustand s_1 mit der Wahrscheinlichkeit $a_{s_0 s_1} = \pi_{s_1}$ gewählt wird und dort das Symbol O_1 mit der Wahrscheinlichkeit $b_{s_1}(O_1)$ erzeugt wird. Dann wird mit der Wahrscheinlichkeit $a_{s_1 s_2}$ in den Zustand s_2 übergegangen und Symbol O_2 mit einer Wahrscheinlichkeit von $b_{s_2}(O_2)$ erzeugt. Der Vorgang wiederholt sich dann so lange, bis der letzte Übergang von Zustand s_{T-1} nach s_T erreicht ist, und Symbol O_T mit einer Wahrscheinlichkeit von $b_{s_T}(O_T)$ generiert wurde.

Will man die Rechenzeit zum Ermitteln von $P(O|\lambda)$ bestimmen, so kommt die Frage auf, wie viele verschiedene Zustandsfolgen S es gibt. Da aus N Zuständen T ausgewählt werden und die Zustände auch öfters vorkommen können (es kann ja von jedem Zustand jedes Ausgabesymbol erzeugt werden) bekommt man N^T verschiedene Zustandsfolgen. Zur Bestimmung von $P(O|\lambda)$ sind dann $(2T - 1)N^T$ Multiplikationen und $N^T - 1$ Additionen notwendig. Der Rechenaufwand liegt also in der Größenordnung $O(TN^T)$ und somit ist diese Methode nicht sehr effizient. Es wären z.B. für $N = 5$, $T = 100$ ungefähr 10^{72} Rechenschritte notwendig.

Zur Lösung des Problems benötigt man also eine effizientere Methode. Solch eine Methode ist als Forward-Backward-Algorithmus bekannt, der im folgenden erklärt wird. Als erstes definieren wir die forward-Variable:

$$\alpha_t(i) := P(O_1, O_2, \dots, O_t, s_t = i | \lambda) \quad (11)$$

$\alpha_t(i)$ ist die Wahrscheinlichkeit dafür, daß λ bis zum Zeitpunkt t die Sequenz O_1, O_2, \dots, O_t erzeugt und dann im Zustand i ist. Sie kann induktiv mit dem forward Algorithmus berechnet werden:

1. $\alpha_1(i) = \pi_i b_i(O_1)$ für alle Zustände i ($1 \leq i \leq N$) (Falls $i \notin S_I$ dann $\pi_i = 0$)

2. for $t := 2$ to T do
 for $j := 1$ to N do

$$\alpha_t(j) = \left(\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right) b_j(O_t) \quad (12)$$

 end;
 end;

3. Die endgültige Wahrscheinlichkeit ist dann:

$$P(O|\lambda) = \sum_{i \in S_F} \alpha_T(i) \quad (13)$$

Im ersten Schritt wird $\alpha_1(i) := P(O_1, s_1 = i | \lambda)$ mit der Wahrscheinlichkeit, daß i der Anfangszustand ist und Symbol O_1 ausgegeben wird, initialisiert, und zwar für jeden Zustand i . Ist $i \notin S_I$, also nicht Anfangszustand, dann ist natürlich $\pi_i = 0$ und somit auch $\alpha_1(i)$. Gleichung (12) veranschaulicht, daß Zustand j zum Zeitpunkt t von allen möglichen Zuständen i zum Zeitpunkt $t - 1$ erreicht werden kann. Es ist zu beachten, daß $\alpha_{t-1}(i)$ die Wahrscheinlichkeit des zusammengesetzten Ereignisses, daß O_1, O_2, \dots, O_{t-1} erzeugt wird und als letzter Zustand i angenommen wird, ist. Also ist das Produkt $\alpha_{t-1}(i) a_{ij}$ die Wahrscheinlichkeit dafür, daß O_1, O_2, \dots, O_{t-1} generiert wird und Zustand j zum Zeitpunkt t über Zustand i zum Zeitpunkt $t - 1$ erreicht wird. Summiert man dieses Produkt über alle möglichen Zustände i zum Zeitpunkt $t - 1$ so erhält man die Wahrscheinlichkeit dafür, daß Zustand j zum Zeitpunkt t über irgend einen Zustand erreicht wurde und die Sequenz O_1, O_2, \dots, O_{t-1} erzeugt wurde. Durch anschließende Multiplikation mit $b_j(O_t)$ wird dann noch die Ausgabesequenz um das entsprechende Symbol erweitert und man erhält $\alpha_t(j)$. Zum Schluß erhält man in Schritt 3 die gewünschte Berechnung von $P(O|\lambda)$. Die letzten forward-Variablen $\alpha_T(i)$ werden über alle möglichen Endzustände $i \in S_F$ aufsummiert,

da $\alpha_T(i) := P(O_1, O_2, \dots, O_T, s_T = i | \lambda)$ ist und es für $P(O | \lambda)$ keine Rolle spielt, welcher Zustand als letzter angenommen wird.

Es bleibt die Frage nach der Rechenzeit. Zum Initialisieren (Schritt 1) sind N Multiplikationen und keine Addition notwendig. Geht man davon aus, daß die Menge S_I mit den Anfangszuständen gegeben ist, werden genau genommen nur N_I Multiplikationen verwendet. In Schritt 2 werden für das einmalige Berechnen der forward-Variable $N + 1$ Multiplikationen und $N - 1$ Additionen gebraucht. $\alpha_t(j)$ wird für jedes $1 \leq j \leq N$ und für jedes $2 \leq t \leq T$ berechnet. In Schritt 3 wird dann nur noch $N - 1$ mal bzw. $N_F - 1$ mal addiert. Es sind also insgesamt $N(N + 1)(T - 1) + N_I$ Multiplikationen und $N(N - 1)(T - 1) + N_F - 1$ Additionen notwendig. Der Rechenaufwand des Algorithmus liegt also in der Größenordnung $O(N^2T)$, was um einiges besser ist als bei der ersten Methode. Es sind hier z.B. für $N = 5, T = 100$ nur ca. 4955 Rechenschritte notwendig im Gegensatz zur ersten Methode wo ungefähr 10^{72} Rechenschritte gebraucht wurden.

Auf eine ähnliche Art und Weise wie die forward-Variable wird die backward-Variable definiert:

$$\beta_t(i) := P(O_{t+1}, O_{t+2}, \dots, O_T | s_t = i, \lambda) \quad (14)$$

Es ist die Wahrscheinlichkeit dafür, daß λ ab Zeitpunkt $t + 1$ die Sequenz $O_{t+1}, O_{t+2}, \dots, O_T$ erzeugt, wenn es zum Zeitpunkt t in Zustand i war. Die backward-Variable kann auch induktiv ähnlich wie die forward-Variable bestimmt werden. Dazu der folgende Algorithmus:

1. $\beta_T(i) = 1$ für alle Zustände i ($1 \leq i \leq N$)
2. for $t := T - 1$ downto 1 do
for $j := 1$ to N do

$$\beta_t(j) = \sum_{i=1}^N a_{ji} b_i(O_{t+1}) \beta_{t+1}(i) \quad (15)$$

end;
end;

3. Die endgültige Wahrscheinlichkeit ist dann:

$$P(O | \lambda) = \sum_{i \in S_I} \pi_i b_i(O_1) \beta_1(i) \quad (16)$$

Der forward sowie der backward-Algorithmus können zur Berechnung der Wahrscheinlichkeit $P(O | \lambda)$ benutzt werden. Der Rechenaufwand beträgt bei beiden $O(N^2T)$. Interessant ist noch der folgende Zusammenhang, der bei der Baum-Welch-Neueinschätzung gebraucht wird:

$$\sum_{i \in S_F} \alpha_T(i) = \sum_{i \in S_I} \pi_i b_i(O_1) \beta_1(i) = P(O | \lambda) = \sum_{i=1}^N P(O, s_t = i | \lambda) = \sum_{i=1}^N \alpha_t(i) \beta_t(i) \quad (17)$$

2.3.2 Baum-Welch-Neueinschätzung

Das komplizierteste Problem im Umgang mit HMMs ist, die Modellparameter (A, B, π) so zu korrigieren, daß die Wahrscheinlichkeit $P(O|\lambda)$ für die Ausgabesequenz maximiert wird. Es ist keine analytische Methode für dieses Problem bekannt. Deshalb wird ein iterativer Algorithmus, der die Wahrscheinlichkeit dem Maximum nähert, gebraucht. Dieser Algorithmus ist der Baum-Welch-Algorithmus. Er wird hier von einem intuitiven Gesichtspunkt aus eingeführt. Für einen genauen Beweis schaue man in [1] Seite 158.

Das Ziel ist es, Parameter für das HMM von der vorhandenen Ausgabesequenz zu erhalten. Sind die Parameter bekannt, kann die Wahrscheinlichkeit für die Sequenz mit dem forward-backward-Algorithmus berechnet werden. Dann können wir eine Neueinschätzung der Modellparameter aufgrund der aktuellen Wahrscheinlichkeiten vornehmen.

Betrachten wir ein Modell, dessen Parameter λ keine Nullwerte enthalten. Die Wahrscheinlichkeitsberechnungen werden jetzt auf dem Modell ausgeführt und es wird so getan, als ob es das endgültige Modell wäre. Unter Gebrauch des forward-backward Algorithmus auf solch einem Modell kann die *a posterior* Wahrscheinlichkeit $\gamma_t(i, j)$ berechnet werden. Sie gibt die Wahrscheinlichkeit dafür an, daß bei gegebenem Modell und gegebener Ausgabesequenz zum Zeitpunkt t Zustand i und zum Zeitpunkt $t + 1$ Zustand j erreicht wird. Aus den Definitionen und Überlegungen aus dem vorherigen Abschnitt läßt sich leicht folgern, daß gilt

$$P(O, s_t = i, s_{t+1} = j | \lambda) = \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \quad (18)$$

und daraus folgt nach Definition der bedingten Wahrscheinlichkeit (Gleichung 5)

$$\gamma_t(i, j) := P(s_t = i, s_{t+1} = j | O, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{k \in S_F} \alpha_T(k)} \quad (19)$$

$\gamma_t(i, j)$ ist die Wahrscheinlichkeit dafür, daß das HMM zum Zeitpunkt t im Zustand i ist und dann zum Zeitpunkt $t + 1$ in den Zustand j übergeht. Offensichtlich berechnet sich diese vereinte Wahrscheinlichkeit aus $\alpha_t(i)$, was für den Pfad, der zum Zeitpunkt t in Zustand i endet, steht, multipliziert mit $a_{ij} b_j(O_{t+1})$, der Wahrscheinlichkeit des Übergangs von Zustand i nach j mit Ausgabe O_{t+1} , multipliziert mit $\beta_{t+1}(j)$, was für den Pfad von Zustand j zum Zeitpunkt $t + 1$ bis zum Schluß steht.

Auf eine ähnliche Weise wird die Wahrscheinlichkeit $\gamma_t(i)$ berechnet. Sie ist die Wahrscheinlichkeit dafür, daß sich das HMM zum Zeitpunkt t im Zustand i befindet, bei vorgegebener Ausgabesequenz O und vorgegebenem HMM λ .

$$\gamma_t(i) := P(s_t = i | O, \lambda) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{k \in S_F} \alpha_T(k)} \quad (20)$$

Aus Gleichung (20) wird ersichtlich, daß sich $\gamma_t(i)$ auch durch $\sum_{j=1}^N \gamma_t(i, j)$ für $t < T$ berechnen läßt. Da diese Art der Berechnung nur Additionen benötigt, ist es besser $\gamma_t(i)$

so zu berechnen, als direkt aus den forward- und backward-Variablen, zumal $\gamma_t(i, j)$ auf jeden Fall benötigt wird.

Summieren wir nun $\gamma_t(i)$ über den Zeitindex t , so bekommen wir einen Wert, den man als die erwartete Anzahl von Zeitpunkten, in denen Zustand i erreicht wird, bezeichnen kann, oder anders ausgedrückt, die erwartete Anzahl von Übergängen, die von Zustand i aus gemacht werden, wenn wir den letzten Zeitpunkt T in der Summe herausnehmen. Genauso kann die Summation von $\gamma_t(i, j)$ über den Zeitindex t ($1 \leq t \leq T - 1$) als die erwartete Anzahl von Übergängen von Zustand i nach j bezeichnet werden. Summieren wir hingegen $\gamma_t(i)$ über alle Zeitpunkte t , in denen Symbol v_k erzeugt wird, also $O_t = v_k$ ist, erhalten wir die erwartete Anzahl von Ausgaben v_k im Zustand i .

Im folgenden soll ein neues Modell $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ erstellt werden, das das alte Modell λ verbessert, d.h., das die Ausgabesequenz mit einer höheren Wahrscheinlichkeit erzeugt. Mit den obigen Überlegungen haben die Neueinschätzungsformel dann folgendes Aussehen:

1. $\bar{\pi}_i = \gamma_1(i), \quad 1 \leq i \leq N$
2. $\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \leq i, j \leq N$
3. $\bar{b}_j(k) = \frac{\sum_{t=1}^T \gamma_t(j)}{\sum_{t=1}^{O_t=v_k} \gamma_t(j)}, \quad 1 \leq j \leq N, 1 \leq k \leq M$

Nochmal zu bemerken wäre, daß die Summation von $\gamma_t(i)$ über den Zeitindex t ($1 \leq t \leq T$) die erwartete Anzahl von Zeitpunkten ist, an denen Zustand i erreicht wird, und die Summation von $\gamma_t(i)$ über den Zeitindex t ($1 \leq t \leq T - 1$) die erwartete Anzahl von Übergängen aus Zustand i ist. Zeitpunkt T entfällt dabei natürlich, weil ja von dort in keinen neuen Zustand mehr übergegangen wird.

Die neuen Matrizen \bar{A} , \bar{B} und $\bar{\pi}$ erfüllen wieder Gleichung (6), womit die Wahrscheinlichkeitsverteilungen wohldefiniert sind. Es kann gezeigt werden, daß entweder das Anfangsmodell λ einen kritischen Zustand darstellt, von dem aus die Wahrscheinlichkeit der Ausgabesequenz nicht mehr verbessert werden kann, also $\bar{\lambda} = \lambda$ ist oder, daß $\bar{\lambda}$ mit höherer Wahrscheinlichkeit die Ausgabesequenz erzeugt, also $P(O|\bar{\lambda}) > P(O|\lambda)$ ist. Gebrauchen wir also iterativ $\bar{\lambda}$ für λ , so kann die Wahrscheinlichkeit für die Sequenz Schritt für Schritt gesteigert werden, bis ein Endpunkt erreicht wird.

3 Bewegungserkennung mit dem HMM

Wir wollen jetzt das Hidden Markov Modell dazu verwenden, um Bewegungen in Bildsequenzen zu erkennen. Diese Bewegungen können z.B. Bewegungen einer Person oder auch Bewegungen von Gegenständen sein. Um ein HMM für eine Sequenz von Bildern $I = I_1, I_2, \dots, I_T$, in denen eine Bewegung dargestellt wird, zu verwenden, muß die Bildsequenz in eine Symbolsequenz O transformiert werden. Es soll dabei jedem Bild I_t der Sequenz ein Symbol O_t zugeordnet werden. Dies geschieht dadurch, daß von jedem Bild I_t der Bildsequenz ein Merkmalvektor $f_t \in \mathbf{R}^n$ extrahiert wird. n ist die Dimension des Merkmalvektors bzw. anders ausgedrückt die Anzahl der Merkmale. Jedem Merkmalvektor wird dann ein Symbol v_k aus der Symbolmenge V zugeordnet.

Es gibt in der Regel mehr verschiedene Merkmalvektoren f_t als Symbole v_k . Das ist einerseits so, weil aus zwei sich ähnlichen Bildern verschiedene Merkmalvektoren extrahiert werden, andererseits aber die Zahl der Symbole v_k gering bleiben soll, damit die Anzahl der Modellparameter nicht zu groß wird und das Modell dadurch nicht an Effizienz verliert. Es werden daher Merkmalvektoren, die sich ähneln, zu einem Repräsentantenvektor zusammengefaßt. Dies geschieht dadurch, daß der Merkmalvektorraum \mathbf{R}^n in M Bereiche aufgeteilt wird, und jeder Bereich einen Repräsentantenvektor g_k bekommt, der möglichst mittig in seinem Bereich liegt. Jedem Repräsentantenvektor g_k wird dann ein Symbol v_k zugeordnet. Die Anzahl der Repräsentantenvektoren ist also gleich der Anzahl verschiedener Ausgabesymbole des HMMs. Die gesamte Zuordnung läuft dann so, daß jedem Merkmalvektor das Ausgabesymbol zugeordnet wird, welches für den Repräsentantenbereich des Merkmalvektors steht. Das bedeutet, f_t wird in Symbol v_k transformiert, wenn der Abstand zwischen f_t und g_k der kleinste aller Abstände zwischen f_t und g_j ($1 \leq j \leq M$) ist. Auf diese Weise entsteht aus dem Bild I_t das Ausgabesymbol $v_k = O_t$.

Es bleibt noch zu beschreiben, wie aus einem Bild I_t der Merkmalvektor f_t erbracht wird. Um den Merkmalvektor f_t zu erzeugen, wird das Bild I_t gerastert, d.h. es wird in gleich große Maschen eingeteilt. Die Anzahl der Maschen ist dann die Dimension bzw. die Anzahl der Merkmale des Vektors. Jedes Merkmal bestimmt sich dann ganz einfach dadurch, indem die Anzahl der schwarzen Bildpunkte (die Bilder müssen schwarz/weiß Bilder sein) gezählt wird. Diese Anzahlen werden dann für jede Masche in den Merkmalvektor eingetragen.

Die Symbolsequenzen, die man mit diesem Verfahren erhält, werden sowohl für die Erkennphasen als auch für die Lernphasen der HMMs gebraucht. Soll das Hidden Markov Modell dazu benutzt werden, eine Bewegungssequenz zu erkennen, wird bei allen HMMs, die jeweils auf eine Bewegungskategorie trainiert worden sind, die Wahrscheinlichkeit für die Sequenz berechnet. Es wird dann die Kategorie als erkannte Kategorie ausgewählt, dessen HMM die Sequenz mit größter Wahrscheinlichkeit erkannt hat. In der Lernphase werden die Parameter eines HMMs mit der angegebenen Sequenz optimiert, d.h. die Wahrscheinlichkeit für die Sequenz erhöht.

4 Experimentelle Ergebnisse

In diesem Abschnitt sollen experimentelle Ergebnisse mit dem HMM zur Bewegungserkennung aufgeführt werden. Sie beruhen nicht auf eigenen Versuchen sondern sind aus [3] übernommen.

4.1 Experimentelle Bedingungen

Der Algorithmus wurde an einer Real-Zeit Bewegungssequenz von Tennisschlägen getestet. Die Kategorien, die erkannt werden sollten, waren sechs verschiedene Tennisschläge: Vorhandschlag, Rückhandschlag, Vorhandvolley, Rückhandvolley, Schmetterball und Aufschlag. Drei Personen führten jede der sechs Tennisschläge 10 mal aus. Sie wurden dabei von einer Videokamera (NTSC, 30 Bilder/Sekunde) aufgenommen, und die Bilder wurden in 200×200 pixel, 256 Graustufen digitalisiert. Die Bewegungssequenzen der Tennisschläge enthielten einen komplizierten Hintergrund (üblicher Hintergrund einer Tennishalle) und die Position des Spielers bewegte sich während der Sequenz in dem Bildfenster. Deswegen war es notwendig, die Person in der Sequenz zu extrahieren, und in die Mitte des Bildes zu rücken und sie entsprechend zu skalieren. Es wurde dabei als erstes ein Bild vom Hintergrund ohne Spieler gemacht und dann das jeweilige Bild mit Spieler darübergelegt und die ungleichen Stellen herausgefiltert. Es wurde mit folgender Bedingung gefiltert

$$\text{if } |I_a(x, y) - I_b(x, y)| < s \text{ then } I_c(x, y) = 0 \text{ else } I_c(x, y) = I_a(x, y), \quad 1 \leq x, y \leq 200$$

wobei I_a das Bild mit Spieler, I_b das Hintergrundbild und I_c das Bild mit dem extrahierten Spieler ist. s ist ein kleiner Schwellenwert, der notwendig ist, da nicht davon ausgegangen werden kann, daß die digitalisierten Hintergründe bis auf jeden Wert genau übereinstimmen. Als nächstes wurden die extrahierten Bilder binarisiert, was heißen soll, daß sie in 0/1 Bilder umgewandelt wurden. Der Hintergrund war ohnehin schon schwarz, also brauchten nur noch die verschiedenen Graustufen des Spielers auf 1, d.h weiß, gesetzt werden. Es wurde mit jedem Bild und jeder Bildsequenz so verfahren.

Aus den binarisierten Bildern wurden dann die Maschenmerkmale (mesh feature), die in Abschnitt 3 beschrieben wurden, extrahiert. Die Maschen wurden 8×8 Pixel groß gewählt. Somit hatte der Merkmalvektor f_i 625 Einträge. Dies ergibt sich daraus, daß die Bilder ja 200×200 Pixel groß sind, also $25 \times 25 = 625$ Maschen Platz haben.

Als Repräsentanten für jede Kategorie wurden 12 Bilder ausgewählt, somit war die Anzahl der HMM Symbole gleich 72 (6 Kategorien a 12 Bilder). Die Symbole 0 bis 11 repräsentierten den Rückhandschlag, 12 bis 23 Rückhandvolley, 24 bis 35 den Aufschlag, 36 bis 47 den Schmetterball, 48 bis 59 den Vorhandschlag und 60 bis 71 Vorhandvolley.

Die Anzahl der Zustände der 6 HMMs (für jede Kategorie einen), die in diesem Experiment gebraucht wurden, wurde auf 36 gesetzt. Da die Dauer der verschiedenen Tennisschläge verschieden lang war, variierte die Länge der Ausgabesequenzen, die den HMMs präsentiert wurden, zwischen 23 und 70 Symbolen, die sich natürlich jeweils durch wiederholen der 12 Repräsentanten einer Sequenz zusammensetzten. Für das Training der HMMs wurden 150 Iterationen für jedes HMM durchgeführt, auch wenn die Wahrscheinlichkeit des HMMs schon nach ca. 100 Iterationen dem Grenzwert sehr nahe war.

4.2 Experiment 1

Im ersten Experiment sollten die Tennisschläge wiedererkannt werden, die von drei Personen ausgeführt wurden. Dazu hat jede Person (A,B und C) jeden der 6 Tennisschläge 10 mal wiederholt. 5 Sequenzen wurden jeweils dazu gebraucht, um die HMMs zu trainieren, und die anderen 5 sollten von den HMMs dann mit hoher Wahrscheinlichkeit erkannt werden. Es sollten dabei immer die Sequenzen der Person, mit der die HMMs auch trainiert wurden, erkannt werden.

Es wurde die Wahrscheinlichkeit für Rückhandvolley mit den sechs HMMs berechnet, die jeweils mit den Daten von Person C trainiert wurden. Das Ergebnis ist in der folgenden Tabelle zu sehen. Es ist klar zu erkennen, daß die richtige Kategorie mit der höchsten Wahrscheinlichkeit erkannt wird.

Tennisschlag	Wahrscheinlichkeit
Rückhandvolley	74,17%
Rückhandschlag	1,33%
Vorhandvolley	9,63%
Vorhandschlag	1,20%
Schmetterball	10,88%
Aufschlag	0,94%

Als nächstes wurden von den 10 Versuchen, die jeder Spieler bei jeder der 6 Kategorien machte, 10 mal 5 verschiedene ausgewählt, um die HMMs zu trainieren, und die anderen 5 wurden für die Wiedererkennung verwendet. Es gab also für jede Person $10 \times 5 \times 6 = 300$ Sequenzen, die erkannt werden sollten. Ziel war es hierbei, die Wiedererkennungsrate zu bestimmen. So wurden z.B. bei Spieler A 272 von 300 Sequenzen richtig erkannt. Die durchschnittliche Wiedererkennungsrate liegt bei 96,0%. Das genaue Ergebnis ist in folgender Tabelle dargestellt.

Spieler	Wiedererkennungsrate
Spieler A	90,66% (272/300)
Spieler B	97,33% (292/300)
Spieler C	100% (300/300)
Durchschnitt	96,00%

Zu bemerken ist noch, daß die Wiedererkennungsrate schlechter wird, wenn die Anzahl der Trainingssequenzen reduziert wird. So fiel die Rate auf bis zu 78,5%, wenn nur mit drei Sequenzen trainiert wurde. Die Leistung des Wiedererkennungssystems hängt allerdings nicht nur von der Anzahl der Trainingssequenzen ab, sondern auch davon, wie gut sie die jeweilige Kategorie repräsentieren. Somit ist es notwendig für ein robustes Wiedererkennungssystem, daß die passenden Trainingssequenzen gewählt werden, also solche, die möglichst gut gestreut sind, was heißen soll, daß die Trainingssequenzen für eine Kategorie herangezogen werden sollten, die am unähnlichsten sind. Weisen die Sequenzen zu starke Differenzen auf, ist es dann allerdings besser, zwei Kategorien, also zwei HMMs, daraus zu machen.

4.3 Experiment 2

Im zweiten Experiment waren die Trainingssequenzen von anderen Spielern wie die Testsequenzen. Die HMMs wurden mit den Sequenzen von einem oder zwei Spielern trainiert und anschließend mit den Sequenzen der anderen Spieler bzw. dem anderen Spieler getestet. Dieses Experiment ist sinnvoll, um den Gebrauch dieser Wiedererkennungsmethoden in wirklichen Anwendungen zu testen, da dort wohl meistens eine Personenunabhängigkeit gefordert wird. Um dies zu erreichen, müssen für das Training Daten von vielen verschiedenen Personen herangezogen werden. In unserem Experiment wurden deshalb die Daten von zwei Personen gemischt, um den Tennisschlag der dritten Person zu erkennen.

Wie man der nachfolgenden Tabelle entnehmen kann, sind die Wiedererkennungsraten nicht so gut wie im ersten Experiment. Das liegt daran, daß die Trainings- und Testsequenzen von verschiedenen Spielern sind. Jede Person hat seine individuellen Eigenarten in der Bewegung, aber die Abweichungen sind begrenzt (es wird ja immer noch der gleiche Schlag ausgeführt), sodaß die Schläge trotzdem noch erkannt werden können. Wie zu sehen ist, wird die Wiedererkennungsrate verbessert, wenn mit gemischten Daten von zwei Spielern trainiert wird. Sie kann so von 61,2% auf 70,8% gesteigert werden.

Testspieler	Trainingspieler			
	A	B	A+B	C
C	61,2%	66,8%	70,8%	100%

Zusammenfassend kann das Hauptergebnis des Experiments wie folgt beschrieben werden: Sind die Trainingsdaten und Testdaten von der gleichen Person wird eine Wiedererkennungsraten von über 90% erreicht. Sind dagegen die Daten von verschiedenen Personen, wie es bei tatsächlichen Anwendungen am häufigsten ist, sinkt die Leistung des Systems. Die Wiedererkennungsraten kann dann allerdings durch Mischen der Daten von verschiedenen Personen wieder etwas gesteigert werden.

Literatur

- [1] X. D. Huang, Y. Ariki, M. A. Jack. "Hidden Markov models for speech recognition". Edinburgh University Press, 1990.
- [2] L. R. Rabiner, B. H. Juang. "An Introduction to Hidden Markov Models". IEEE ASSP MAGAZINE, Seite 4-16, Jan. 1986.
- [3] J. Yamato, J. Ohya, K. Ishii. "Recognizing Human Action in Time-Sequential Images using Hidden Markov Model". Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Seite 379-385, Jun. 1992.
- [4] K. Krickeberg, H. Ziezold. "Stochastische Methoden". Springer-Verlag, 1995.
- [5] R. Ineichen. "Stochastik - Einführung in die elementare Statistik und Wahrscheinlichkeitsrechnung". Rabeer Verlag Luzern, 6. Auflage 1984.
- [6] A. Engel. "Stochastik". Ernst Klett Verlag, 1. Auflage 1987.