Universität Karlsruhe
Fakultät für Informatik

Institut für Theoretische Informatik
Prof. Dr. Alexander Waibel

Audio-visuelle Aktivitätenerkennung und Personenverfolgung in einer Büroumgebung

Diplomarbeit

von

Christian Wojek

Juni 2006

Betreuer:

Prof. Dr. Alexander Waibel
Dr.-Ing. Rainer Stiefelhagen
Dipl.-Inform. Kai Nickel

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Karlsruhe, 8. Juni 2006 .....

Inhaltsverzeichnis

1 Einleitung

Die automatische Erkennung von Aktivitäten hat sich in den vergangenen Jahren zu einem sehr aktiven Forschungsgebiet entwickelt. Dabei wird die Problematik nicht nur mit Methoden der Bildverarbeitung intensiv erforscht, sondern auch in den Bereichen *Ubiquitous* oder *Pervasive Computing*. Dabei bezeichnet Ubiquitous Computing die Allgegenwart von Informationstechnologie, während unter dem Schlagwort Pervasive Computing das Eindringen von Computern in den Alltag in der Form von intelligenten, vernetzten Gegenständen zusammengefasst wird. Die hierbei durchgeführten Arbeiten verwenden durchaus ähnliche Modelle und Vorgehensweisen, unterscheiden sich aber in der Art der Umweltsensoren, bei denen es sich zum Beispiel um RFID-Marker oder einfache Kleinstcomputer handeln kann.
Die Spannweite der Anwendungen reicht von reinen Überwachungsaufgaben hin zur Anwendung in intelligenten Räumen. Diese erlauben es, zwischenmenschliche Interaktionen zu analysieren und darauf aufbauend weitere Dienste zur Verfügung zu stellen. Eine Beispielanwendung könnte etwa das folgende Szenario sein. Der Beginn einer spontanen Besprechung wird von einem intelligenten Raum erkannt, woraufhin dieser den Mitschnitt der Diskussion initiiert und so im Nachhinein die Rekapitulation aller Diskussionspunkte ermöglicht, ohne dass das explizite Eingreifen eines Menschen von Nöten war.
Außerdem kann zum Beispiel aus Informationen über die Aktivitäten von Personen auf deren Verfügbarkeit geschlossen werden und ob gerade eine Unterbrechung der Tätigkeit möglich ist. Danninger *et al.* [DFB+05, DKR+06] zeigen zum Beispiel eine Anwendungsdomäne, bei der Telefonate und Nachrichten auf der Grundlage der Verfügbarkeit eines Benutzers weitergeleitet werden.
Weiterhin kann die Erkennung von Aktivitäten dazu dienen, die Absichten eines Benutzers besser zu verstehen und die Rollen von Personen innerhalb einer Gruppe zu deuten.
Diese Arbeit ist Teil des EU-Projekts „Computers in the Human Interaction Loop" (CHIL) [WSStCPC04], das zum Ziel hat, bessere Mensch-Maschine Schnittstellen zu entwerfen. Insbesondere sollen dabei Computersysteme entwickelt werden, die menschliche Interaktionen beobachten, analysieren und davon ausgehend verschiedene Dienste anbieten können, um so ein leichteres computerunterstütztes Leben und Arbeiten zu ermöglichen.
Dem nachfolgenden Überblick über das für diese Arbeit implementierte System in Abschnitt 1.1 schließt sich in Abschnitt 1.2 eine Übersicht über den Stand der Forschung an, die sich im Wesentlichen auf Arbeiten aus der Bildverarbeitung

konzentrieren soll, da auch die vorliegende Arbeit in diesem Bereich einzuordnen ist. Den Abschluss der Einführung bildet eine kurze Übersicht zum Beitrag der vorliegenden Arbeit in Abschnitt 1.3

1.1 Systemüberblick

Bei den wesentlichen mathematischen Grundlagen dieser Arbeit handelt es sich um Hidden Markov Models (HMMs) und Mixturen aus Gaußverteilungen, die in Kapitel 2 eingeführt werden. Ziel dieser Arbeit ist es, ein System zu entwickeln, das in der Lage ist, Situationen, die in einer typischen Mehrpersonen-Büroumgebung entstehen, zu erfassen. Dies soll auf der Basis audio-visueller Merkmale geschehen, die in Kapitel 3 näher vorgestellt werden. Diese werden mit einem sehr einfachen Aufbau erfasst, der pro Büroraum jeweils eine Kamera und ein omnidirektionales Mikrofon umfasst. Zur besseren Veranschaulichung ist die Versuchsumgebung in Abbildung 1.1 schematisch dargestellt.
Eine mehrschichtige Hierarchie von Hidden Markov Models dient dann dazu, Aktivitäten zu erkennen, wobei Details hierzu in Kapitel 4 näher erläutert werden. Die Videomodalität ermöglicht dabei auf unterster Ebene die Erkennung von **Ereignissen** wie JEMAND SITZT AM SCHREIBTISCH A oder JEMAND VERLÄSST BÜRO B. Weiterhin kann durch Ausnutzung des Audiosignals Sprache von Hintergrundgeräuschen getrennt werden. Als Ereignis werden hierbei zeitlich kurze Aktivitäten bezeichnet, die vornehmlich in lokal definierten Bereichen stattfinden. Auf einer höheren Ebene sollen dem Eingabestrom dann semantische Beschreibungen dessen, was geschieht, zugeordnet werden. Dabei handelt es sich für dieses System um die Klassen NIEMAND IM BÜRO, DISKUSSION, BESPRECHUNG, TELEFONAT und SCHREIBTISCHARBEIT. Diese werden als **Situationen** bezeichnet und erstrecken sich über einen längeren Zeitraum und bieten eine globale Beschreibung dessen, was in einem Raum geschieht.
Schließlich sollen die beteiligten Benutzer anhand ihres Bewegungsprofiles mit Hilfe eines Bayes'schen Filters auf Raumniveau verfolgt und identifiziert werden. Auch hierauf wird im zweiten Teil des Kapitels 4 näher eingegangen.
Für die Evaluation der Arbeit in Kapitel 5 ist zu bemerken, dass realistische, unbeeinflusste und dadurch schwierig zu klassifizierende Daten verwendet wurden, um den Ansatz zur Aktivitätenerkennung experimentell zu erproben. Die Schwierigkeiten resultieren in erster Linie aus den stark wechselhaften Beleuchtungsverhältnissen, die von künstlichem durch Deckenlampen abgestrahltem Licht bis hin zu Tageslicht mit Sonnenschein und Schlagschatten reichen.
Für die Experimente zur Evaluation der Personenverfolgung wurde ein zweiter Datensatz verwendet. Die Hauptschwierigkeit des Personenverfolgungsproblems besteht darin, dass auf dem Flur aus Gründen der Privatsphäre keine Kamera vorhanden ist und somit Benutzer auf dem Weg in ein anderes Büro oder auf dem Weg zum Drucker im Seminarraum (Lab) nicht sichtbar sind.

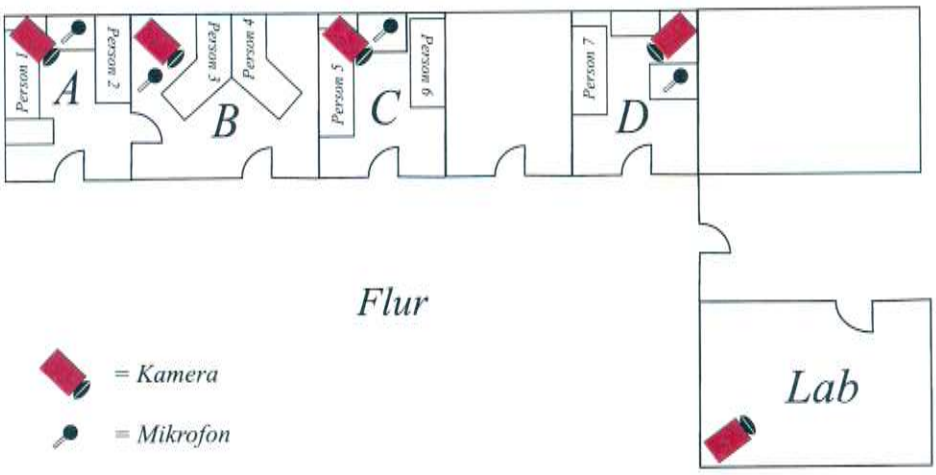

Abbildung 1.1: *Schematische Darstellung der Versuchsumgebung; A-D bezeichnen die vier überwachten Büros, während der Seminarraum mit Lab benannt ist.*

Abschließend zu dieser kurzen Einführung sei auf Abbildung 1.2 verwiesen, die nochmals einen kurzen Überblick zum Aufbau des implementierten Systems geben soll.

1.2 Stand der Forschung

Im Bereich der Bildverarbeitung ist die Erkennung von Aktivitäten oftmals verbunden mit Überwachungsaufgaben. So gibt es etwa in Großbritannien nach Informationen von heute.de mehr als zwei Millionen Überwachungskameras. Allein die Menge an Daten macht deshalb deutlich, dass deren Auswertung unmöglich von Hand geschehen kann.
So schlagen zum Beispiel Oliver *et al.* [ORP00] vor, gekoppelte Hidden Markov Models zu verwenden, um Interaktionen von Fußgängern auf öffentlichen Plätzen zu erkennen. Bei der Evaluation des Ansatzes auf künstlich generierten Daten werden Aktivitäten wie zum Beispiel sich treffende und einander folgende Personen erkannt. Dabei stellt sich außerdem heraus, dass gekoppelte Hidden Markov Models für die Erkennung des Verhaltens von Fußgängern besser geeignet sind als herkömmliche HMMs.
Ganz ähnlich schlagen Stauffer und Grimson [SG00] ein Echtzeitpersonenverfolgungssystem vor, das eine Hierarchie von Klassifikatoren unüberwacht erlernt und mit dieser in der Lage ist, ungewöhnliche Ereignisse zu erkennen. Das System basiert dabei auf einem adaptiven Hintergrundmodell, bei dem die auftretenden Grauwerte pro Bildpunkt mit einer Mixtur von Gaußverteilungen beschrieben werden. Dieses Überwachungssystem stellt für das Verfolgen von Personen in Außenbereichen unter wechselnden Lichtverhältnissen eine gute Lösung dar. Auch Johnson und Hogg [JH96] lernen die gewöhnlichen Trajektorien von Fußgängern

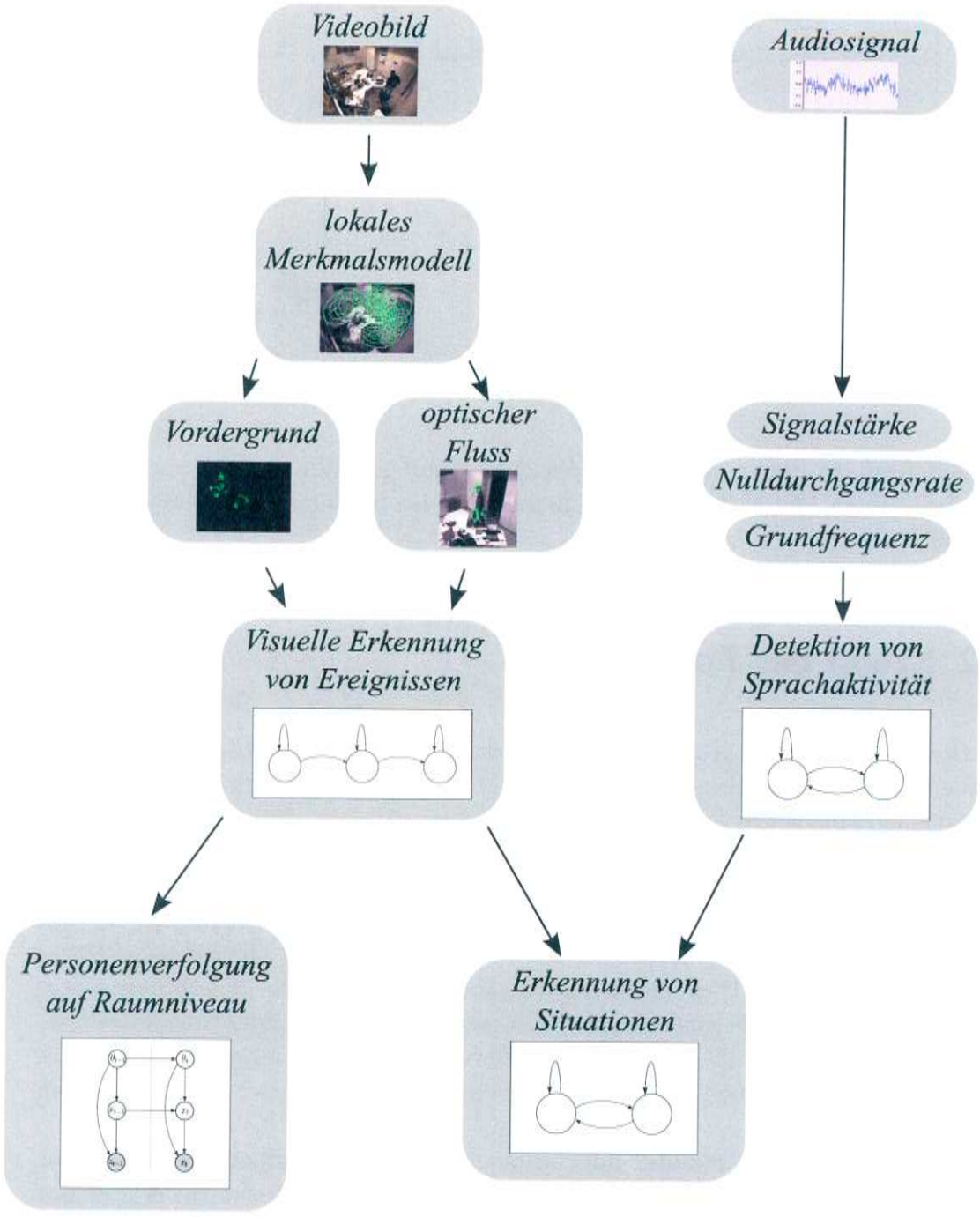

Abbildung 1.2: *Übersicht zum Aufbau des implementierten Systems zur Aktivitätenerkennung und dem Verfolgen von Personen auf Raumniveau*

in einer Außenumgebung mit einem Verfahren, das auf neuronalen Netzen basiert, und sind damit in der Lage, Zwischenfälle zu detektieren.
Bei Hongeng *et al.* [HBN00] wird eine mehrschichtige Hierarchie naiver Bayes-Klassifikatoren unter anderem dazu verwendet, Aktivitäten auf einem Parkplatz zu überwachen. Als Merkmale dienen dabei die Bahnen bewegter Vordergrundbereiche.
Des Weiteren präsentieren Brand und Kettnaker [BK00] einen auf Entropieminimierung basierenden Algorithmus zur Bestimmung der Parameter eines HMMs. Das entwickelte Verfahren wird mit einer Überwachungsaufgabe, bei der es darum geht, eine Kreuzung zu beobachten und ungewöhnliche Ereignisse zu detektieren, evaluiert. Darüber hinaus werden in einer Ein-Personen-Büroumgebung Aktivitäten unter konstanten Verhältnissen, basierend auf Videomerkmalen, erkannt. Hierbei zeigt sich, dass die vom Trainingsalgorithmus erlernten Zustände, verglichen mit dem Baum-Welch Algorithmus, besser vom Menschen zu interpretieren sind.
In [IB00] versehen Ivanov und Bobick kontextfreie Grammatiken mit Wahrscheinlichkeitsübergängen und erhalten dadurch einen stochastischen Parser, um Aktivitäten auf der Grundlage visueller Informationen zu erkennen, wobei Elementarereignisse durch Hidden Markov Models erkannt werden. Das Echtzeitsystem kann seine Leistungsfähigkeit dabei bei einer Parkplatzüberwachungsaufgabe, wie auch bei der Erkennung von Gesten unter Beweis stellen.
Abseits von Überwachungsaufgaben zeigen Bobick und Davis [BD01], dass sogenannte *Temporal Templates* wie *Bewegungsverlaufsbilder* und *Bewegungsenergiebilder* ausreichend sind, um menschliche Bewegungen bei der Durchführung von Sportübungen zu klassifizieren. Der echtzeitfähige Algorithmus vergleicht dabei bekannte Trainingsbilder mit unbekannten Beobachtungen und ist invariant gegenüber linearen Geschwindigkeitsänderungen.
Ben-Arie *et al.* [BAWPR02] zeigen ein System, das menschliche Bewegung, basierend auf der Pose verschiedener Körperteile wie Hände, Beine und Torso mit Hilfe eines mehrdimensionalen Hashverfahrens klassifiziert. Dabei funktioniert das vorgeschlagene Verfahren auch bei einer niedrigen Verarbeitungsrate sehr zuverlässig und kann trotz eines im Vergleich zu den Trainingsdaten um bis zu 30 Grad veränderten Blickwinkels eingesetzt werden. Die zu klassifizierenden Bewegungsabläufe sind dabei gestellt und die einzelnen Körperteile werden mit einem sogenannten *EXpansion Matching (EXM) Filter* verfolgt.
Demirdjian *et al.* [DTK+02] gehen davon aus, dass in einer Büroumgebung die ausgeführte Aktivität stark vom Aufenthaltsort der beteiligten Personen abhängig ist. Deshalb schlagen sie vor, 3D-Trajektorienpunkte, die durch einen Verfolgungsalgorithmus gewonnen werden, zusammen mit der Bewegungsgeschwindigkeit zu sogenannten *Activity Maps* mit dem *k-Means*-Verfahren zu clustern.
Der vorliegenden Arbeit am ähnlichsten ist jedoch das SEER System, das von Oliver *et al.* [OHG02] entwickelt wurde. Dabei wird ein mehrschichtiges Hidden Markov Model ausgenutzt, um ein Ein-Personen-Aktivitätenerkennungssystem in einer Büroumgebung zu verwirklichen. Es werden allerdings nicht nur Informationen aus

dem Kamerabild und dem Audiosignal extrahiert, sondern diese werden zusätzlich noch mit Benutzereingaben am Arbeitsplatzcomputer fusioniert, um die Situation in einem Blickfeld vor dem Computer des Benutzers zu analysieren. Diese Arbeit unterscheidet sich von der vorliegenden dahingehend, dass die Lichtverhältnisse stabil sind, so dass farbabhängige Merkmale verwendbar sind. Außerdem wird die Erkennungsaufgabe durch Verwendung von Tastatur- und Mauseingaben sowie die Einschränkung des Sichtfeldes, in denen Personen sehr gut erkennbar sind, vereinfacht. Dies ermöglicht die Verwendung von Detektoren für Frontalansichten von Gesichtern, um die Anzahl von anwesenden Personen auf einfache Art und Weise zu bestimmen. Das hier präsentierte System ist im Gegensatz zu SEER durch die einfachere Ausstattung mit Sensoren wesentlich flexibler, allerdings auch eingeschränkter bei der Wahl der Merkmale.
Ein weiterer wichtiger Beitrag kommt von Zhang und McCowan *et al.* [ZGPBM06, MGPB+05], die ebenfalls mehrschichtige Hidden Markov Models einsetzen, um Besprechungen automatisch zu verstehen. Im Gegensatz zu der Umgebung der hier vorgestellten Arbeit ist jeder Benutzer mit einem Mikrophon, das am Hemdkragen befestigt wird, ausgestattet. Zusätzlich kommen ein Mikrophone Array sowie drei Kameras zum Einsatz. Das beschriebene System ist allerdings auf eine feste Anzahl von genau vier Leuten und auf den Einsatz in einem gut ausgestatteten, intelligenten Raum beschränkt. Die Arbeiten von Zhang und McCowan untersuchen dabei verschiedene Wege, Informationen aus visueller und Audiomodalität mit unterschiedlichen Modellansätzen zu verbinden, wobei die verwendeten Daten gestellt wurden. Auf dem gleichen, öffentlich zugänglichen Datensatz arbeiten auch Dielmann und Renals [DR04], um verschiedene Arten von dynamischen Bayes-Netzen zur Erkennung von Besprechungsaktivitäten zu erforschen.
Daneben existieren noch zahlreiche weitere Arbeiten verschiedener Autoren [GX03, DBPV05, HHE03, ZGPBM05], die unterschiedlichste Arten von HMMs beziehungsweise Bayes-Netze dazu verwenden, Aktivitäten in diversen Domänen zu erkennen.
Was den Aspekt der Personenverfolgung angeht, stammen die meisten ähnlichen Arbeiten aus dem Bereich Pervasive Computing, wo Beobachtungen hauptsächlich mittels RFID Sensoren oder sonstigen Kleinstcomputern gesammelt werden. So statten Wilson und Atkeson [WA05] ein Altenheim mit binären Sensoren wie zum Beispiel druckempfindlichen Matten aus und versehen die Bewohner mit Bewegungssensoren, um einerseits deren Aufenthaltsort zu verfolgen und gleichzeitig Informationen über die Bewegungsaktivität zu erhalten.
Außerdem versehen Schulz *et al.* [SFH03, FHL+03] Personen mit Infrarotmarkern und verwenden in einer Innenumgebung Laserscanner, um die Benutzer gleichzeitig zu verfolgen und zu identifizieren.
Schließlich existieren einige Arbeiten von Black *et al.* [MEB04, BME05] zum Verfolgen von Objekten durch mehrere Kameras mit nicht überlappendem Sichtfeld. Die Verfahren konzentrieren sich dabei unter anderem auf das automatische, unüberwachte Lernen der Positionen der Kameras zueinander und werden bei der Überwachung von Verkehrsszenen evaluiert.

1.3 Forschungsbeitrag

Die bereits im Überblick zum Stand der Forschung kurz erwähnten Beiträge dieser Arbeit seien an dieser Stelle nochmals übersichtlich zusammengefasst.

- Der verwendete Versuchsaufbau zeichnet sich durch seine Einfachheit aus. Pro Raum kommen lediglich eine unkalibrierte Kamera, sowie ein handelsübliches omni-direktionales Mikrofon zum Einsatz. Sämtliche überwachten Personen sind nicht mit Sensoren ausgestattet, so dass lediglich ein Videobild aus einer nicht weiter kontrollierten Umgebung, sowie ein einfaches mit Störgeräuschen überlagertes Audiosignal zur Verfügung steht.
- Aktivitäten können für eine beliebige Anzahl von Personen erkannt werden, so dass es unerheblich ist, ob diese sich ständig in der überwachten Umgebung aufhalten oder nicht.
- Für die Evaluation dieser Arbeit werden realistische Daten verwendet, die in mehreren Büros ohne Einflussnahme gesammelt wurden. Das Geschehen ist also nicht gestellt und die Umgebungsbedingungen unkontrolliert, was bedeutet, dass mitunter starke Beleuchtungsunterschiede auftreten.
- Des Weiteren kommen neuartige Merkmale zum Einsatz, die auf einer lokalen Beschreibung von Bewegung basieren. Das lokale Merkmalsmodell kann dabei datengetrieben auf einer geringen Anzahl von Trainingsdaten erlernt werden.
- Zusätzlich zur Aktivitätenerkennung können Personen visuell anhand erkannter Ereignisse mit einem verteilten Kameranetzwerk verfolgt werden. Dabei ist es wichtig zu bemerken, dass die Ansichten der Kameras nicht überlappen.

2 Grundlagen

Das folgende Kapitel soll dazu dienen, die wesentlichen mathematischen Grundlagen zu erläutern, die für das Verständnis dieser Arbeit notwendig sind. Dies sind zum einen Hidden Markov Models, die in Abschnitt 2.1 vorgestellt werden sowie Mixturen von Gaußverteilungen, auf die in Abschnitt 2.2 mit dem zugehörigen EM-Trainingsalgorithmus eingegangen wird.

2.1 Hidden Markov Models

Hidden Markov Models stellen ein mächtiges Werkzeug dar, um stochastische Prozesse über die Zeit zu beschreiben. Sie haben sich unter anderem in der Spracherkennung bewährt, wo sie zum Beispiel dazu dienen, Phoneme zu erkennen. Eine sehr gute Einleitung zum Thema findet sich bei Rabiner [Rab89]. Der nachfolgende Abschnitt 2.1.1 stellt zunächst Markovketten vor und erweitert diese dann im Abschnitt 2.1.2 zu Hidden Markov Models, auf deren Eigenschaften im Weiteren eingegangen wird.

2.1.1 Markovketten

Gegeben sei ein System, das durch eine Menge von N Zuständen $S = \{S_1, \ldots, S_N\}$ charakterisiert werden kann. Der beobachtete stochastische Prozess befinde sich zu jedem Zeitschritt t in einem Zustand q_t. Des Weiteren sei die Veränderung des Systemzustandes für jeden Zustand i durch eine Übergangsfunktion beschrieben. Eine vollständige stochastische Beschreibung des Prozesses wäre allerdings nur unter der Kenntnis des kompletten Prozessverlaufs möglich, weshalb man für eine Markovkette erster Ordnung aus praktischen Gründen die Annahme trifft, dass die Übergangsfunktion nur vom jeweils vorhergehenden Zustand abhängig ist:

$$\begin{aligned} a_{ij} &= P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots) \\ &= P(q_t = S_j | q_{t-1} = S_i) \end{aligned}$$

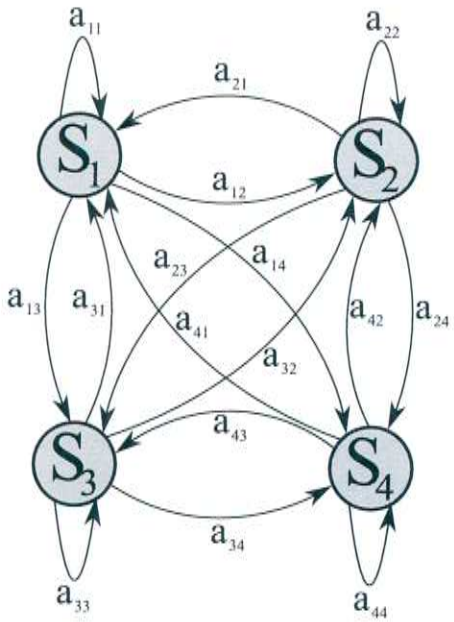

Abbildung 2.1: *Markovkette mit vier diskreten Zuständen und allen Zustandsübergängen*

Um für die Übergangsfunktion eine Wahrscheinlichkeitsdichte zu erhalten, muss weiterhin gelten:

$$a_{ij} > 0$$
$$\sum_{j=1}^{N} a_{ij} = 1$$

Darüber hinaus wird durch

$$\pi_i = P(q_1 = S_i), \quad 1 \leq i \leq N$$

eine Anfangsverteilung über die möglichen Startzustände festgelegt.

Abbildung 2.1 zeigt zur Veranschaulichung eine Markovkette mit vier diskreten Zuständen. Da Ereignisse oder Beobachtungen in diesem Modell mit den Zuständen verbunden sind, spricht man auch von einem beobachtbarem Markovmodell. Anhand dieses Modells können nun beispielsweise Berechnungen über die Wahrscheinlichkeit einer Beobachtung $O = \{S_1, S_2, S_4, S_3, S_1\}$ aufgestellt werden:

$$\begin{aligned} P(O|Markovkette) &= P(S_1, S_2, S_4, S_3, S_1|Markovkette) \\ &= P(S_1) \cdot P(S_2|S_1) \cdot P(S_4|S_2) \cdot P(S_3|S_4) \cdot P(S_1|S_3) \\ &= \pi_1 \cdot a_{12} \cdot a_{24} \cdot a_{43} \cdot a_{31} \end{aligned}$$

2.1.2 Hidden Markov Models

Um praktische Probleme zu modellieren, eignen sich Markovketten allerdings auf Grund der zu restriktiven Annahmen nicht. Deshalb soll das eingeführte Markovkettenmodell nun zu einem Hidden Markov Model (HMM) weiterentwickelt werden. Hierfür wird der zu Grunde liegende stochastische Prozess in zwei Teilprozesse unterteilt. Der erste Teilprozess modelliert dabei die Zustandsübergänge wie von der Markovkette bereits bekannt. Allerdings wird die Annahme getroffen, dass der Zustand, in dem sich das System befindet, verborgen und nicht direkt beobachtbar ist. Stattdessen ist an jeden Zustand eine Wahrscheinlichkeitsdichte gebunden, die angibt wie hoch die Wahrscheinlichkeit ist, eine bestimmte Ausgabe zu beobachten. Bei dieser Wahrscheinlichkeitsdichte kann es sich um eine beliebige diskrete oder kontinuierliche Verteilung handeln. Für praktische Probleme werden oftmals Mixturen von Gaußverteilungen verwendet, da mit diesen jede beliebige Verteilung angenähert werden kann.
Die Sequenz der Beobachtungen bis zu einem Zeitpunkt T wird meist mit $O = O_1 O_2 \cdots O_T$ abgekürzt. Formal kann ein Hidden Markov Model mit diskreten Beobachtungswahrscheinlichkeiten also wie folgt definiert werden:

- N sei die Anzahl der verborgenen Zustände, die mit $S = \{S_1, \ldots, S_N\}$ bezeichnet werden sollen. Der gegenwärtige Zustand zu einem Zeitpunkt t wird im Folgenden q_t genannt.
- M sei die Anzahl der möglichen Beobachtungen $V = \{V_1, \ldots, V_M\}$
- $A = \{a_{ij} | 1 \leq i, j \leq N\}$ gibt die Zustandsübergangsverteilung an. Abhängig davon, welche Zustandsübergänge erlaubt sind, ergeben sich verschiedene Topologien, die in Abschnitt 2.1.3 näher erläutert werden.
- $B = \{b_i(k) | 1 \leq i \leq N, 1 \leq k \leq M\}$ bezeichnet die Ausgabewahrscheinlichkeitsdichte für jeden Zustand, wobei

$$b_i(k) = P(v_k \text{ wird zum Zeitpunkt t beobachtet } | q_t = S_i)$$

- Die Anfangsverteilung wird durch $\pi_i = P(q_1 = S_i)$ festgelegt.

Folglich ergibt sich der zu erlernende Parametersatz für bekanntes N und V als

$$\lambda = \{A, B, \pi\}$$

Für die praktische Anwendung ergeben sich daraus folgende Teilprobleme, die in den nächsten Abschnitten genauer betrachtet werden sollen:

Evaluierungsproblem: Wie lässt sich für eine gegebene Beobachtung O und ein gegebenes Modell λ möglichst effizient die Beobachtungswahrscheinlichkeit $P(O|\lambda)$ berechnen?

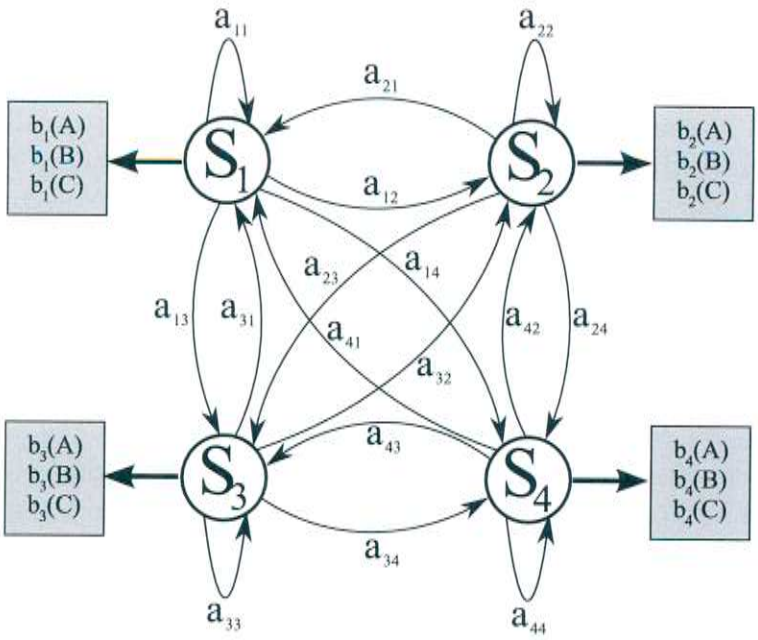

Abbildung 2.2: *Ergodisches Hidden Markov Model mit* $V = \{A, B, C\}$

Dekodierungsproblem: Welche Zustandsfolge $Q = q_1 q_2 \dots q_T$ kann die Beobachtung O am besten erklären?

Lernen der Parameter: Wie stellt man die Parameter λ ein, um für gegebene Ausgaben O die Ausgabewahrscheinlichkeit $P(O|\lambda)$ zu maximieren? Dafür existieren mehrere geeignete Lernverfahren, von denen das Baum-Welch-Verfahren näher erläutert werden soll.

2.1.3 Topologien

Folgende Topologien sind in praktischen Anwendungen verbreitet und sollen deshalb hier näher vorgestellt werden. Es sollte außerdem angemerkt werden, dass für ein und denselben stochastischen Prozess oftmals verschiedene Modelle existieren können, die diesen hinreichend gut beschreiben.

Ergodisches Modell: Als ergodisch wird ein HMM dann bezeichnet, wenn alle Zustände voll verbunden sind. Das heißt für alle Übergangswahrscheinlichkeiten gilt $a_{ij} > 0$.

Links-Rechts-Modell: Eine Links-Rechts-Topologie ist dadurch charakterisiert, dass Zustandsübergänge nur in eine Richtung möglich sind. Formal bedeutet dies:

$$a_{ij} = \begin{cases} a_{ij} > 0 & j \geq i, \quad 1 \leq i, j \leq N \\ 0 & \text{sonst} \end{cases}$$

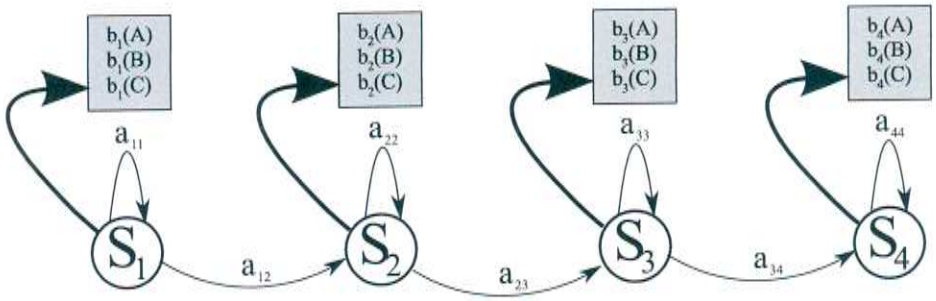

Abbildung 2.3: *Links-Rechts Hidden Markov Model mit* $V = \{A, B, C\}$

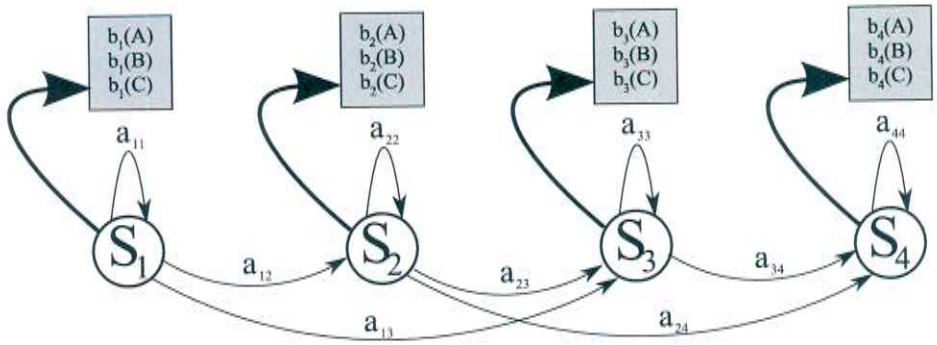

Abbildung 2.4: *Bakis Hidden Markov Model mit* $V = \{A, B, C\}$

Bakis-Modell: Die Bakis-Topologie ist ein Spezialfall der Links-Rechts-Topologie, denn neben Selbstübergängen und Übergängen zum nächsten Zustand ist nur noch das Überspringen eines Zustandes erlaubt. Also gilt:

$$a_{ij} = \begin{cases} a_{ij} > 0 & j = i+1 \vee j = i+2 \vee j = i, \quad 1 \leq i, j \leq N \\ 0 & \text{sonst} \end{cases}$$

In den Abbildungen 2.2 bis 2.4 sind die drei Topologien zum leichteren Verständnis als Schaubilder dargestellt.

2.1.4 Lösung des Evaluierungsproblems

In diesem Abschnitt soll der Frage nach der effizienten Berechnung der Wahrscheinlichkeit $P(O|\lambda)$ einer Beobachtung O bei einem gegebenen HMM nachgegangen werden. Dabei habe $O = O_1 O_2 \cdots O_T$ die Länge T. Unter der Annahme, dass die Beobachtungen nur vom gegenwärtigen Zustand abhängen und untereinander unabhängig sind, lässt sich die Beobachtungswahrscheinlichkeit ausdrücken

als:

$$\begin{aligned} P(O|\lambda) &= \sum_{Q \in Q^*} P(O|Q,\lambda) P(Q|\lambda) \\ &= \sum_{q_1,q_2,\cdots,q_T \in Q^*} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T) \end{aligned}$$

Dabei stellt Q^* die Menge aller möglichen Zustandsfolgen $Q = q_1 q_2 \cdots q_T$ dar. Bei genauer Betrachtung stellt man fest, dass eine sequentielle Berechnung aller auftretenden Wahrscheinlichkeiten eine Komplexität in der Größenordnung von $2T \cdot N^T$ besitzt und deshalb für praktische Probleme kaum durchführbar ist.
Die Lösung dieses Problems erfolgt mit Hilfe des Forward-Algorithmus, der sich das Prinzip des dynamischen Programmierens zu Nutze macht. Hierfür wird zunächst die Forward-Variable $\alpha_t(i)$ definiert als:

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i|\lambda)$$

$\alpha_t(i)$ gibt also die Wahrscheinlichkeit an, bei gegebenem Modell λ die Ausgabesequenz $O_1, \ldots, O_t$ zu beobachten und sich zum Zeitpunkt t dann im Zustand S_i zu befinden. Der Vorteil dieser Definition liegt darin, dass die Berechnung induktiv, wie im Folgenden gezeigt, durchgeführt werden kann:

- Initialisierung:

$$a_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N$$

- Induktionsschritt:

$$a_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) b_j(O_{t+1}), \quad 1 \leq t \leq T-1, 1 \leq i, j \leq N$$

Schließlich erhält man $P(O|\lambda)$, indem man die Forward-Variable über alle möglichen Endzustände aufsummiert:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Analysiert man die Laufzeit dieses induktiven Algorithmus, so wird man feststellen, dass dieser nur in der Größenordnung von $N^2 T$ Schritten liegt, was eine wesentliche Verbesserung darstellt.
Für die beiden verbleibenden Probleme ist es zudem sinnvoll, analog zur Forward-Variablen eine Backward-Variable $\beta_t(i)$ einzuführen:

$$\beta_t(i) = P(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, \lambda)$$

Diese enthält die Wahrscheinlichkeit, von einem Zustand q_t aus die restliche Beobachtung $O_{t+1}O_{t+2}\cdots O_T$ zu machen.
Auch hier ist eine induktive Berechnung möglich:

- Initialisierung:
$$\beta_T(i) = 1, \quad 1 \leq i \leq N$$

- Induktionsschritt:
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \cdots, 1; 1 \leq i \leq N$$

2.1.5 Lösung des Dekodierungsproblems

Im Gegensatz zum Evaluierungsproblem gibt es für das Dekodierungsproblem keine eindeutige Lösung, da die angenommene optimale Zustandsabfolge von der Wahl des Gütekriteriums abhängt. Wählt man als Kriterium einen einzelnen besten Pfad, lässt sich das Problem mit dem so genannten Viterbi-Algorithmus lösen, der dem Forward-Algorithmus sehr ähnlich ist. Dazu wird zunächst die Hilfsvariable $\delta_t(i)$ definiert:

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda)$$

Diese bezeichnet die maximale Wahrscheinlichkeit entlang eines einzelnen Pfades während der ersten t Zeitschritte bis zum Zustand q_t. Offensichtlich ist eine induktive Definition für den nächsten Zeitschritt möglich:

$$\delta_{t+1}(j) = \left(\max_i \delta_t(i) a_{ij} \right) \cdot b_j(O_{t+1})$$

Um schlussendlich den optimalen Pfad auslesen zu können, muss entlang des Pfades für jeden Zeitschritt gespeichert werden, welcher Zustand obige Gleichung maximiert hat. Dies geschieht formal mit Hilfe der Variablen $\Psi_t(j)$.

- Initialisierung:
$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1), \quad 1 \leq i \leq N \\ \psi_1(i) &= 0 \end{aligned}$$

- Rekursionsschritt ($2 \leq t \leq T, 1 \leq j \leq N$):
$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} \left(\delta_{t-1}(i) a_{ij} \right) b_j(O_t) \\ \psi_t(j) &= \arg\max_{1 \leq i \leq N} \left(\delta_{t-1}(i) a_{ij} \right) \end{aligned}$$

- Terminierung:

$$\begin{aligned} p^* &= \max_{1 \le i \le N} (\delta_T(i)) \\ q_T^* &= \arg\max_{1 \le i \le N} (\delta_T(i)) \end{aligned}$$

Die optimale Zustandssequenz kann dann leicht aus den gespeicherten Zwischenschritten rückwärts abgelesen werden:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \cdots, 1$$

2.1.6 Erlernen der Parameter

Schließlich bleibt das Problem, die Parameter λ eines Hidden Markov Models aus Trainingsdaten zu bestimmen. Es ist gleichzeitig das schwierigste der drei Probleme, wobei auch hierfür mehrere Lösungsansätze bestehen. Allerdings ist keiner davon in der Lage, das globale Optimum der Parameterkonfiguration garantiert zu finden. An dieser Stelle soll auf den Baum-Welch-Algorithmus, der bei genauer Betrachtung ein *Expectation-Maximization* Algorithmus ist, eingegangen werden, der iterativ auf ein lokales Maximum konvergiert. Dafür wird zunächst die Hilfsvariable $\xi_t(i,j)$ definiert:

$$\begin{aligned} \xi_t(i,j) &= \frac{P(q_t = S_i, q_{t+1} = S_j | O, \lambda)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \end{aligned}$$

Außerdem soll $\gamma_t(i)$ bei gegebenem Modell λ und Beobachtung O als die Wahrscheinlichkeit definiert werden, sich zum Zeitschritt t in S_i zu befinden:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \sum_{j=1}^{N} \xi_t(i,j)$$

Folglich ist die zu erwartende Anzahl von Übergängen aus Zustand S_i:

$$\sum_{t=1}^{T-1} \gamma_t(i)$$

und die zu erwartende Anzahl von Übergängen von Zustand S_i nach Zustand S_j:

$$\sum_{t=1}^{T-1} \xi_t(i,j)$$

Damit lässt sich ein iteratives Verfahren definieren, um aus gegebenem $\lambda = \{A, B, \pi\}$ ein $\overline{\lambda} = \{\overline{A}, \overline{B}, \overline{\pi}\}$ neu zu schätzen, so dass gilt $P(O|\overline{\lambda}) \geq P(O|\lambda)$:

$$\begin{aligned}
\overline{\pi}_i &= \gamma_1(i) \\
\overline{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\
\overline{b}_j(k) &= \frac{\sum_{t=1,\text{ wobei } O_t=V_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}
\end{aligned}$$

Werden mehrere Trainingssequenzen $O^1, O^2, \ldots$ verwendet, so werden diese als stochastisch unabhängig betrachtet. Bei der Verwendung einer Mixtur von M Gaußverteilungen für die Modellierung der Ausgabewahrscheinlichkeiten wird $b_j(O)$ definiert als:

$$b_j(O) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(O, \mu_{jm}, U_{jm})$$

Hierbei bezeichnet c_{jm} das Gewicht der m-ten Mixturkomponente im Zustand S_j und μ_{jm} sowie U_{jm} entsprechend Mittel und Kovarianzmatrix. Damit lässt sich $\gamma_t(j,k)$ umdefinieren als Wahrscheinlichkeit, dass im Zeitschritt t im Zustand S_j die k-te Mixturkomponete für O_t verantwortlich ist:

$$\gamma_t(j,k) = \left(\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right) \left(\frac{c_{jk}\mathcal{N}(O_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^{M} c_{jm}\mathcal{N}(O_t, \mu_{jm}, U_{jm})} \right)$$

Anhand dessen lassen sich die Mixturparameter wie folgt anpassen:

$$\begin{aligned}
\overline{c}_{jk} &= \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \\
\overline{\mu}_{jk} &= \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot O_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \\
\overline{U}_{jk} &= \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot (O_t - \mu_{jk})(O_t - \mu_{jk})^T}{\sum_{t=1}^{T} \gamma_t(j,k)}
\end{aligned}$$

2.1.7 Implementierung

Für diese Arbeit wurde auf die Implementierung von HMMs in der Torch Bibliothek [CBM02] zurückgegriffen. Diese verwendet aus numerischen Gründen logarithmierte Wahrscheinlichkeiten, was bedeutet, dass beim Forward-Algorithmus

statt $P(O|\lambda)$ in Wirklichkeit $\log P(O|\lambda)$ berechnet wird. Diese Notwendigkeit besteht, da nach Definition alle Wahrscheinlichkeiten größer als null und kleiner als eins sind und folglich bei mehrfacher Multiplikation sehr kleine Wahrscheinlichkeiten entstehen. Wie anhand der Herleitung ersichtlich ist, tritt dieses Problem insbesondere bei langen Eingabesequenzen und langen Merkmalsvektoren auf, was zu Unterläufen bei Fließkommazahlen führen kann. Die Skalierung mit Logarithmen behebt dieses Problem. Außerdem werden für die Modellierung der Ausgabewahrscheinlichkeiten $b_j(O)$ in den Zuständen nicht volle Kovarianzmatrizen verwendet, sondern nur Diagonalmatrizen, die die Korrelation zwischen den einzelnen Merkmalen vernachlässigen. Dies führt zu einer niedrigeren Anzahl zu schätzender Parameter, so dass die benötigte Menge an Trainingsdaten ebenfalls kleiner wird.

2.2 Mixturen von Gaußverteilungen

Mixturen von Gaußverteilungen stellen eine wichtige Möglichkeit dar, um beliebige Wahrscheinlichkeitsverteilungen anzunähern. Das anschließende Kapitel stellt nach einer kurzen, allgemeinen Problembeschreibung einen *Expectation-Maximization* Algorithmus zum Lernen der Parameter vor [Bil97, DLR77].

2.2.1 Allgemeine Problemstellung

Anhand einer gegebenen Menge von Datenpunkten $x = \{x^{(1)}, \ldots, x^{(N)}\}$ soll eine Verteilung definiert werden, die diese erzeugt haben kann. Dabei wird von einem zweistufigen Zufallsexperiment ausgegangen, bei dem im ersten Schritt eine von k möglichen Mixturkomponenten $\mathcal{N}_j$ anhand der diskreten Wahrscheinlichkeitsdichte W ausgewählt wird, woraufhin im zweiten Schritt auf Grund deren Parameter μ_j, Σ_j ein Datenpunkt gezogen wird. Die gesuchte Darstellung hat also die Form:

$$\begin{aligned} P(x|W,\mu,\Sigma) &= \sum_{i=1}^{k} p(W = w_i) \cdot \mathcal{N}(x, \mu_i, \Sigma_i) \\ &= \sum_{i=1}^{k} p(W = w_i) \cdot \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)} \end{aligned}$$

2.2.2 EM-Algorithmus zum Erlernen der Parameter

Das globale Optimum der Parameterkombination $\Theta = \{W, \mu, \Sigma\}$ lässt sich leider im allgemeinen Fall nicht analytisch bestimmen, weshalb man auf einen Expectation-Maximization Algorithmus zurückgreifen muss, der allerdings auf einem

lokalen Maximum konvergieren kann. Dabei geht man von der stochastischen Unabhängigkeit der einzelnen Datenpunkte $x^{(1)}, \ldots, x^{(N)}$ aus und ist bestrebt, $P(x|\Theta) = P(x^{(1)}|\Theta) \cdots P(x^{(N)}|\Theta)$ zu maximieren. Bei der Herleitung behilft man sich bei der Schätzung der verborgenen Parameter w_i damit, dass man ausgehend von einer Initialisierung die Erwartungswerte verwendet, dann in mehreren Iterationen die restlichen Parameter durch Maximierung der Beobachtungswahrscheinlichkeit berechnet und daran anschließend die Erwartungswerte $E[w_{i,j}]$ wieder anpasst. Dadurch erhält man für den *Expectation*-Schritt:

$$E[w_{i,j}] = \frac{w_i \cdot \mathcal{N}(x_j, \mu_i, \Sigma_i)}{\sum_{i=1}^{k} w_i \cdot \mathcal{N}(x_j, \mu_i, \Sigma_i)}$$

Im *Maximization*-Schritt werden die Parameter wie folgt angepasst:

$$\begin{aligned} \mu_i &= \frac{\sum_{j=1}^{N} E[w_{i,j}] x_i}{\sum_{j=1}^{N} E[w_{i,j}]} \\ \Sigma_i &= \frac{\sum_{j=1}^{N} E[w_{i,j}](x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{N} E[w_{i,j}]} \\ w_i &= \frac{\sum_{j=1}^{N} E[w_{i,j}]}{N} \end{aligned}$$

Algorithmus 1 EM-Algorithmus zur Bestimmung der Parameter einer Mixtur von Gaußverteilungen

Initialisiere w_i, μ_i, Σ_i mit K-Means Clusteralgorithmus
Wiederhole
  Für $j = 1, \ldots, N$ **Berechne** ▷ Expectation-Schritt
    Für $i = 1, \ldots, k$ **Berechne**
      $E[w_{i,j}] \leftarrow \frac{w_i \cdot \mathcal{N}(x_j, \mu_i, \Sigma_i)}{\sum_{i=1}^{k} w_i \cdot \mathcal{N}(x_j, \mu_i, \Sigma_i)}$
    Ende Für
  Ende Für
  Für $i = 1, \ldots, k$ **Berechne** ▷ Maximization-Schritt
    $\mu_i \leftarrow \frac{\sum_{j=1}^{N} E[w_{i,j}] x_i}{\sum_{j=1}^{N} E[w_{i,j}]}$
    $\Sigma_i \leftarrow \frac{\sum_{j=1}^{N} E[w_{i,j}](x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{N} E[w_{i,j}]}$
    $w_i \leftarrow \frac{\sum_{j=1}^{N} E[w_{i,j}]}{N}$
  Ende Für
Bis Parameter konvergiert sind
Rückgabe $\forall i : 1 \leq i \leq k : w_i, \mu_i, \Sigma_i$

Der komplette Ablauf ist in Algorithmus 1 genauer dargestellt. Initialisiert wird dieser Algorithmus für gewöhnlich, indem man die Datenpunkte x mit dem wohl bekannten *k-Means* Clusteralgorithmus bezüglich eines passenden Abstandsmaßes gruppiert.

3 Merkmale

Nach Einführung der wichtigsten Grundlagen soll nun näher auf deren konkrete Anwendung eingegangen werden. Der erste Schritt für den Aufbau eines Lernalgorithmus ist die Auswahl geeigneter Merkmale, die die zu erkennenden Klassen gut charakterisieren. Dies wird in der gegebenen Umgebung durch mehrere Faktoren erschwert.
Das Audiosignal etwa wird durch Störquellen wie zum Beispiel die Lüfter von Computern oder die Bewegung von Stühlen überlagert. Außerdem sind zwei der Büros durch eine Zwischentüre, die allerdings die meiste Zeit geöffnet ist, direkt verbunden, so dass Geräuschquellen aus dem jeweiligen Nebenraum wahrgenommen werden. Da es für die gewünschte Anwendung ausreichend ist, Sprache von Hintergrundgeräuschen zu trennen, genügt es allerdings, wenige, eher einfache Merkmale zu verwenden. Dieses sind Signalenergie, Nulldurchgangsrate und Grundfrequenz, die in Abschnitt 3.1 genauer erläutert werden.
Die Auswahl von Merkmalen, die den Videoeingangsstrom beschreiben, erschwert sich durch die stark wechselnden Beleuchtungsbedingungen. Diese sind einerseits Resultat der Tatsache, dass das System über mehrere Stunden hinweg zuverlässig arbeiten soll und somit Tageslicht als auch künstliche Beleuchtung am Abend auftreten. Außerdem haben die Büros sehr große Fensterfronten, so dass bei leicht bedecktem Himmel mit vorbeiziehenden Wolken von diffuser Beleuchtung bis hin zu direkter Sonneneinstrahlung mit Schlagschatten alles auftreten kann. Dies bedeutet, dass jegliches Merkmal, das von Farbe abhängig ist, für eine gute Beschreibung kaum geeignet ist. Weiterhin wird die Situation dadurch erschwert, dass Personen oftmals in beträchtlicher Distanz zur Kamera stehen und deren Köpfe aus allen möglichen Richtungen wahrgenommen werden. In Folge dessen sind auch Gesichtsdetektoren kaum zu verwenden, um die Anzahl der Personen zu ermitteln, die sich in einem Büro befinden.
Es bleibt also die Verwendung von Merkmalen, die von der Bewegung der Benutzer abhängig sind. Zur Verwendung kamen hierfür ein einfacher adaptiver Bewegungsdetektor sowie der optische Fluss, der die Bewegungsrichtung näher charakterisiert. Details zu diesen visuellen Erkennungsmerkmalen finden sich in Abschnitt 3.2.

3.1 Audiomerkmale

Wie bereits erwähnt soll das Audiosignal nur in Abschnitte eingeteilt werden, in denen gesprochen wird und in solche, in denen nicht gesprochen wird. Die hierfür extrahierten Merkmale Signalenergie, Nulldurchgangsrate und Grundfrequenz werden in den folgenden Abschnitten näher vorgestellt.
Dabei sei das eingehende Audiosignal bereits digitalisiert und werde mit $s_a(n)$ bezeichnet, wobei n den Zeitschritt bezeichnet. Außerdem wird das Signal in Fenster der Länge N unterteilt, die jeweils halb überlappend sind, wobei m den Index des Fensters bezeichnet. Für die Experimente wurde eine Fensterlänge von $N = 320$ Abtastungen verwendet, was bei einer Aufnahmefrequenz von 16 kHz 20 Millisekunden entspricht. Des Weiteren sei die Signumfunktion definiert als:

$$\text{sign}(s_a(n)) = \begin{cases} 1 & \text{fürs}_a(n) \geq 0 \\ -1 & \text{fürs}_a(n) < 0 \end{cases}$$

3.1.1 Signalenergie

Die Energie eines Signals ist definiert als das durchschnittliche Quadrat der Amplitude über einem Zeitfenster:

$$SP(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} s_a(n)^2$$

3.1.2 Nulldurchgangsrate

Die Nulldurchgangsrate misst die Rate der Nulldurchgänge der Signalamplitude bezogen auf die gesamte Länge eines Fensters. Sie ist gegeben durch:

$$ZCR(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} \frac{|\text{sign}(s_a(n)) - \text{sign}(s_a(n-1))|}{2}$$

3.1.3 Grundfrequenz

Die Grundfrequenz ist von den drei verwendeten Merkmalen das am schwierigsten zu bestimmende. Es gibt verschiedenste Algorithmen hierfür, wovon in dieser Arbeit der so genannte YIN Algorithmus nach de Cheveigné und Kawahara [dK02] verwendet wird. Dieser ist autokorrelationsbasiert und verwendet einige Nachverarbeitungsschritte, um Fehler zu vermeiden. Der genaue Ablauf soll im Folgenden näher erläutert werden.
Allgemein ist die Grundfrequenz F_0 definiert als die Inverse der Periode T, die

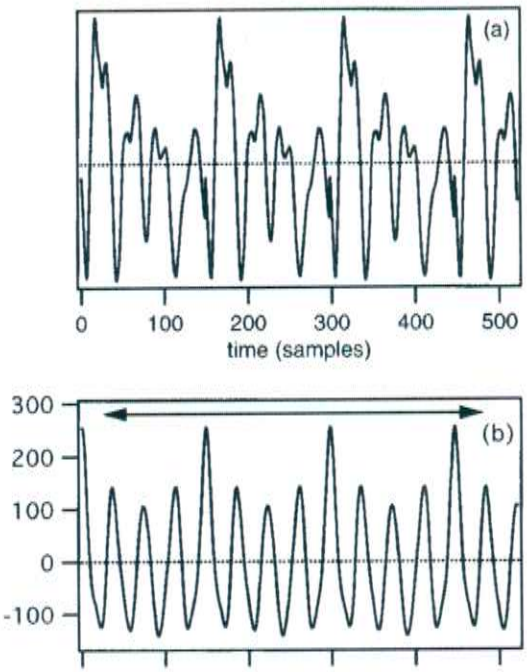

Abbildung 3.1: *(a) Signalbeispiel einer Sprachquelle. (b) Autokorrelationsfunktion $r_n(\tau)$ zur Signalquelle aus (a) (aus [dK02])*

als kleinste Verschiebung im Zeitbereich definiert ist, gegenüber der das Signal $s_a(n)$ invariant ist. Diese Definition eignet sich allerdings nur für unmodulierte, vollkommen periodische Signale, die in der realen Welt natürlich kaum vorkommen. Sämtliche Modulationen wie Musik oder Sprache machen jedoch die perfekte Periodizität zunichte, was ein Problem für alle Detektionsalgorithmen darstellt. Klassischerweise kann die gesuchte Verschiebung mit Hilfe der Autokorrelationsfunktion $r_t(\tau)$ errechnet werden. Diese ist definiert als:

$$r_n(\tau) = \sum_{j=n+1}^{n+N} s_a(j)s_a(j+\tau)$$

Abbildung 3.1 zeigt ein beispielhaftes Signal und die zugehörige Autokorrelationsfunktion. Für ein periodisches Signal treten hier bei Vielfachen der Periode Maxima auf, bei der die Autokorrelationsmethode in einem Suchfenster die Position des globalen Maximalwerts zur Periodendauer bestimmt. Damit geht das Problem einher, dass bei falscher Größe des Suchfensters ein Maximum höherer Ordnung gewählt werden kann. Um Fehler dieser Art für praktische Anwendungen zu vermeiden, wird das Differenzkriterium als besseres Maß für Verschiebungsinvarianz vorgeschlagen:

$$\forall n : s_a(n) - s_a(n+T) = 0$$

An dieser Bedingung ändert sich auch nichts, wenn man über ein gesamtes Zeitfenster aufsummiert, so dass die gesuchte Periode durch das Minimum der folgen-

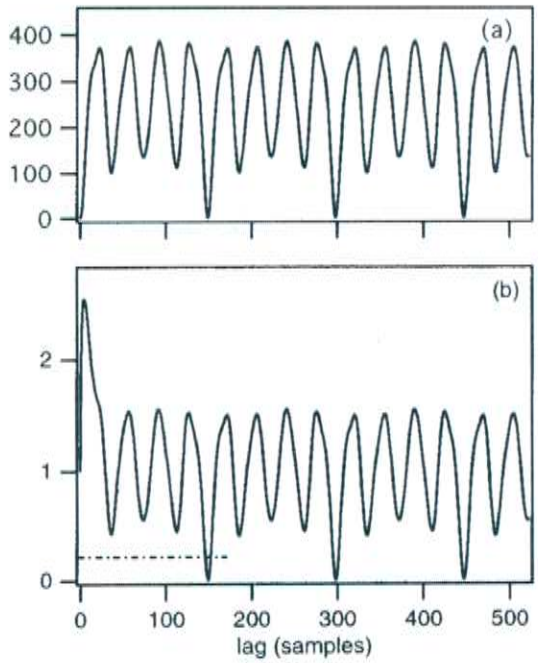

Abbildung 3.2: *(a) Differenzfunktion $d_n(\tau)$ für das Signal aus Abbildung 3.1(a) (b) zugehörige kumulative mittelnormalisierte Differenzfunktion mit sinnvollem Schwellwert als gestrichelte Linie (aus [dK02])*

den Funktion definiert ist:

$$d_n(\tau) = \sum_{j=1}^{N} \left(s_a(n) - s_a(n+\tau)\right)^2$$

Abbildung 3.2 zeigt $d_n(\tau)$ für die bereits bekannte Beispielfunktion. Problematisch bei dieser Herangehensweise ist, dass die Funktion bei $\tau = 0$ immer Null wird, aber oftmals im Bereich der wirklichen Periodendauer verschieden von Null ist, da das Eingangssignal nicht streng periodisch ist. Abhilfe hierfür schafft die so genannte *kumulative mittelnormalisierte Differenzfunktion*, die sich wie folgt ergibt:

$$d'_n(\tau) = \begin{cases} 1 & \tau = 0, \\ d_n(\tau) / \frac{1}{\tau} \sum_{j=1}^{\tau} d_n(j) & \text{sonst} \end{cases}$$

$d'_n(\tau)$ tendiert im Gegensatz zu $d_n(\tau)$ dazu, für niedrige τ groß zu bleiben und fällt erst unter eins, sobald $d_n(\tau)$ kleiner als der Durchschnitt von $d_n(\tau)$ für kleinere τ wird. Es bleibt somit noch das Problem, dass Minima, die aus einem Vielfachen der Periode resultieren, das globale Minimum darstellen können. De Cheveigné und Kawahara schlagen zur Behebung dieses Problems vor, einen Schwellwert u festzulegen und die Verschiebung, die dem ersten Minimum entspricht, das diesen unterschreitet als Periodendauer $T_{min}(m)$ zu bestimmen. Wird u nicht unterschrit-

ten, so wird das globale Minimum zurückgegeben. Für die verwendete Implementierung betrug $u = 0,5$. Abbildung 3.2 veranschaulicht schließlich $d'_n(\tau)$ für die Beispielfunktion.
Darüber hinausgehend werden noch zwei weitere Schritte vorgeschlagen, die allerdings wenig zur Verbesserung des Fehlers beitragen und deshalb nicht implementiert wurden. Zum Einen wird vorgeschlagen, $d'_n(\tau)$ mit einer Parabel zu interpolieren, um dem Fehler, der durch eine zu niedrige Abtastrate entsteht, entgegen zu wirken. Zum Anderen wird ein abschließender Glättungsschritt durchgeführt, um Schätzungsdiskontinuitäten vorzubeugen.
Für die vorliegende Implementierung ergibt sich die Grundfrequenz somit als:

$$F_0(m) = \frac{1}{T_{min}(m)}$$

3.1.4 Synchronisation

Da der Videostrom mit einer Abtastrate von 1 Hz verarbeitet werden soll, die Audiomerkmale allerdings mit einer Frequenz von 50 Hz berechnet werden, muss ein geeigneter Weg gefunden werden, die Merkmale zu synchronisieren. Dies wurde dadurch erreicht, dass für jedes Audiomerkmal das Mittel und die Varianz über den Zeitraum einer Sekunde berechnet wurde. Der sich neu ergebende Index soll hier mit k bezeichnet werden, die Mittel mit $\overline{S}(k), \overline{ZCR}(k), \overline{F}_0(k)$ und die Varianzen mit $\mathrm{Var}(S(k)), \mathrm{Var}(ZCR(k)), \mathrm{Var}(F_0(k))$. Damit besteht der Beobachtungsvektor o_t^a, der für die HMMs zur Audioklassifikation verwendet wird, aus den folgenden sechs Komponenten:

$$o_t^a = (\overline{S}(k), \overline{ZCR}(k), \overline{F}_0(k), \mathrm{Var}(S(k)), \mathrm{Var}(ZCR(k)), \mathrm{Var}(F_0(k)))^T$$

3.2 Videomerkmale

Wie bereits einleitend erwähnt, konnten für die Verarbeitung des Videosignals nur bewegungsabhängige Merkmale extrahiert werden. Dies sind zum einen die Vordergrundbereiche, die durch einen einfachen, adaptiven Segmentierungsalgorithmus gewonnen werden sowie der optische Fluss. Vor der Anwendung der in den nächsten beiden Abschnitten beschriebenen Verfahren, werden die Farbbilder noch in Bilder mit 256 Graustufen konvertiert.

3.2.1 Vordergrundbereiche

Auf Grund der wechselnden Beleuchtung kann zur Modellierung des Hintergrundes kein statisches Bild verwendet werden. Die alleinige Verwendung von Differenzbildern ist allerdings auch nicht wünschenswert, da dieses Verfahren bei wenig

Bewegung sehr schnell keine Vordergrundbereiche mehr liefert. Deshalb soll ein einfaches, adaptives Verfahren, das zwischen diesen beiden Extremen einen Kompromiss findet, zur Anwendung kommen. Dieses soll für jeden Pixel $i \in \mathbb{N}^2$ ein Hintergrundmodell $bg(i)$ nach der folgenden Lernregel anpassen:

$$bg_t(i) = \alpha \cdot bg_{t-1}(i) + (1 - \alpha) \cdot p_t(i), \quad 0 \leq \alpha \leq 1$$

Dabei bezeichnet bg_t das Hintergrundmodell zum Zeitpunkt t, p_t das aktuell beobachtete Eingabebild und α die Lernrate. Um nun zu entscheiden, ob ein Pixel dem Vordergrund oder dem Hintergrund zugehörig ist, wird ein Differenzbild zwischen dem Hintergrundmodell und dem aktuell beobachteten Eingabebild errechnet und auf Grund dessen mit Hilfe eines Schwellwerts m entschieden. Das Vordergrundmodell $fg(i)$ entsteht also wie folgt:

$$fg_t(i) = \begin{cases} 0 & |p_t(i) - bg_{t-1}(i)| \leq m \\ |p_t(i) - bg_{t-1}(i)| & |p_t(i) - bg_{t-1}(i)| > m \end{cases}$$

Differenzen, die kleiner sind als der Schwellwert, werden also zu Null gesetzt und folglich dem Hintergrund zugerechnet, Pixel mit einem positiven Abstand $fg_t(i) > 0$ gehören zu einer Vordergrundregion. Der Abstand $fg_t(i)$ gibt zudem ein Indiz dafür, wie stark sich der beobachtete Vordergrund vom Hintergrund unterscheidet. So haben typischerweise Vordergrundbereiche, die aus Schattenwürfen resultieren, relativ niedrige Werte von $fg_t(i)$. Bildbereiche, die Personen zeigen, die nicht zum Hintergrund adaptiert wurden, besitzen hingegen sehr hohe Werte $fg_t(i)$ für die entsprechenden Pixel. Abschließend sollen noch kurz die beiden grenzwertigen Einstellungen der Lernrate diskutiert werden. Für $\alpha = 0$ wird das aktuelle Eingabebild p_t im nächsten Zeitschritt als Hintergrundmodell verwendet, so dass man also ein reines Differenzbildverfahren erhält. Im Gegensatz dazu erhält man für $\alpha = 1$ ein statisches Hintergrundbild, das über die Zeit nicht adaptiert wird.
Abbildung 3.3 zeigt beispielhaft einige Segmentierungsergebnisse für verschiedene Büros. In der linken Spalte ist jeweils das Eingangsbild p_t zu sehen. Die rechte zeigt das Segmentierungsergebnis, wobei das Grün umso intensiver dargestellt wird, je größer der Abstand $fg_t(i)$ zum Hintergrundmodell ist.

3.2.2 Optischer Fluss

Um das Blendenproblem möglichst gut zu vermeiden, muss die Bestimmung des optischen Flusses in zwei Schritten erfolgen. Der erste dient dazu, geeignete Bildpunkte zu finden, an denen die Verschiebung zuverlässig zu extrahieren ist. Hierfür sind zum Beispiel untexturierte Flächen sehr schlecht geeignet, wo hingegen eckenartige Strukturen sehr gut in zwei aufeinanderfolgenden Bildern beim zweiten Verarbeitungsschritt einander zugeordnet werden können. Abbildung 3.4 zeigt den

Abbildung 3.3: *Segmentierungsergebnisse für beispielhafte Bildausschnitte aus allen Büros. Grüne (helle) Bereiche entsprechen dem Vordergrund*

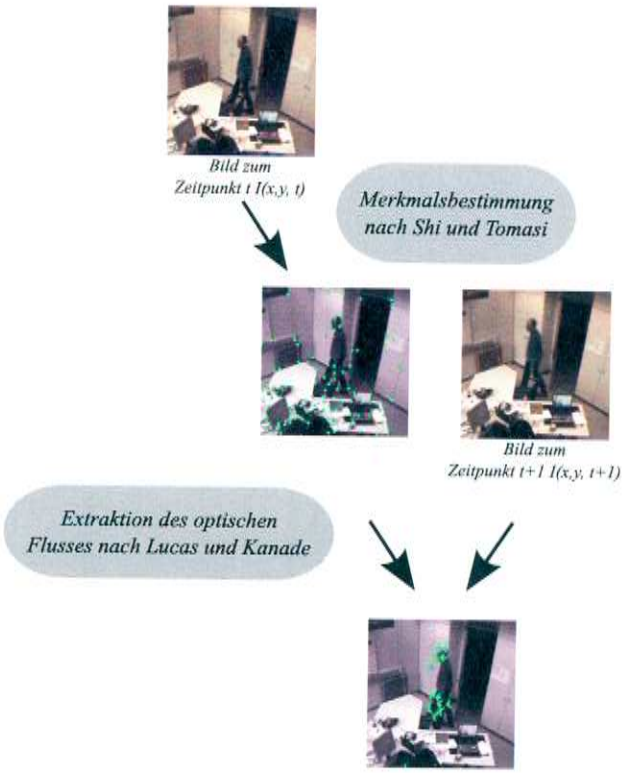

Abbildung 3.4: *Gesamtablauf zur Bestimmung des optischen Flusses*

Gesamtablauf zur Bestimmung des optischen Flusses. Der nächste Abschnitt befasst sich somit also zunächst mit der Auswahl geeigneter Merkmalspunkte des zweidimensionalen Eingabebildes und der übernächste Abschnitt mit der eigentlichen Bestimmung des optischen Flusses.

Extraktion geeigneter Merkmalspunkte

Die Bestimmung geeigneter Merkmalspunkte geschieht durch die Minimierung eines Fehlerkriteriums, wie von Shi und Tomasi vorgeschlagen [ST94, ST93]. Die Autoren fordern dabei zunächst, dass die Wahl der zu verfolgenden Merkmalspunkte von der verwendeten Verfolgungsmethode abhängig sein muss. Zunächst wird hierfür allgemein die Verschiebung (ξ, η) eines Bildpunktes $I(x, y, t)$ dargestellt als:

$$I(x, y, t + \tau) = I(x - \xi(x, y, \tau), y - \eta(x, y, \tau), t)$$

t bezeichnet hierbei die Zeiteinheit und τ den zeitlichen Abstand der aufgenommen Eingabebilder. $\delta = (\xi, \eta)$ wird im Folgenden als Ausdruck für die *Verschiebung* verwendet. Diese wird allerdings besser als die Kombination einer affinen Transformation mit einer Verschiebung ausgedrückt:

$$\delta = Dx + d$$

Damit lässt sich das zeitlich später aufgenommene Bild J als Ergebnis einer affinen Transformation mit Verschiebung des ersten Bildes I ausdrücken:

$$I(x) = J(Ax + d) = J([1 + D]x + d) = J\left(\left[\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{pmatrix}\right] x + \begin{pmatrix} d_x \\ d_y \end{pmatrix}\right)$$

Da diese Beziehung auf Grund von Bildrauschen und anderen Einflüssen nicht immer exakt besteht, wird vorgeschlagen, den Fehler zu minimieren. Dies geschieht durch die Minimierung des Fehlermaßes *Verschiedenheit* ϵ in Abhängigkeit der Parameter der Transformationsmatrix D und des Verschiebungsvektors d.

$$\epsilon = \int \int_W [J(Ax + d) - I(x)]^2 w(x)\, \mathrm{d}x$$

In diesem Ausdruck bezeichnet W den Bildausschnitt um den in Frage kommenden Pixel. Mit Hilfe einer Gauß-Funktion als Gewichtungsfaktor $w(x)$ können dabei Pixel im Zentrum des Fensters stärker gewichtet werden. Leitet man ϵ nach den Parametern aus D und d ab und nähert $J(Ax + d)$ mit seiner Taylorentwicklung, so erhält man ein sechsdimensionales Gleichungssystem der Form:

$$T \begin{pmatrix} d_{xx} \\ d_{yx} \\ d_{xy} \\ d_{yy} \\ d_x \\ d_y \end{pmatrix} = \int \int_W [I(x) - J(x)] \begin{pmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{pmatrix} w\, \mathrm{d}x$$

mit

$$g = \nabla J = \begin{pmatrix} \frac{\partial J}{\partial x} \\ \frac{\partial J}{\partial y} \end{pmatrix} = \begin{pmatrix} g_x \\ g_y \end{pmatrix}$$

$$T = \int \int_W \begin{pmatrix} U & V \\ V^T & Z \end{pmatrix} w\, \mathrm{d}x$$

$$U = \begin{pmatrix} x^2 g_x^2 & x^2 g_x g_y & xyg_x^2 & xyg_x g_y \\ x^2 g_x g_y & x^2 g_y^2 & xyg_x g_y & xyg_y^2 \\ xyg_x^2 & xyg_x g_y & y^2 g_x^2 & y^2 g_x g_y \\ xyg_x g_y & xyg_y^2 & y^2 g_x g_y & y^2 g_y^2 \end{pmatrix}$$

$$V^T = \begin{pmatrix} xg_x^2 & xg_x g_y & yg_x^2 & yg_x g_y \\ xg_x g_y & xg_y^2 & yg_x g_y & yg_y^2 \end{pmatrix}$$

$$Z = \begin{pmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{pmatrix}$$

Die Lösung dieses Gleichungssystems ist offensichtlich nicht ganz einfach und mitunter auch nicht nötig, um gute Merkmale für die Extraktion des optischen Flusses zu finden, so dass Shi und Tomasi vorschlagen, die Gleichungsterme, die von der affinen Transformation abhängig sind, zu vernachlässigen und stattdessen das folgende einfachere Gleichungssystem zu betrachten:

$$Zd = e$$

Hierbei bezeichnet e die beiden letzten Vektorkomponenten des ursprünglichen Ausdrucks rechts des Gleichheitszeichens. Im experimentellen Teil von Shis und Tomasis Arbeit stellt sich heraus, dass die Einbeziehung affiner Transformationen bei der Merkmalsauswahl vor allem dann hilfreich ist, wenn die gleichen Merkmale über einen längeren Zeitraum verfolgt werden sollen. Bei kleinen Zeitschritten, so wie es bei der Bestimmung des optischen Flusses der Fall ist, ist die obige Vereinfachung sinnvoll, was allerdings voraussetzt, dass dann für jeden Zeitschritt neue Merkmale berechnet werden.
Ein Bildpunkt kann also gut verfolgt werden, wenn das obige Gleichungssystem eine Lösung besitzt. Dies ist der Fall wenn Z wohl konditioniert ist und die Einträge über dem Rauschniveau liegen. Für die beiden Eigenvektoren von Z bedeutet dies, dass sie sowohl möglichst groß als auch von der gleichen Größenordnung sein sollen.

- Zwei kleine Eigenwerte bedeuten, dass sich der betrachtete Pixel in einer Fläche gleicher Intensität befindet.
- Ein großer und ein kleiner Eigenwert entspricht der Tatsache, dass eine Textur in eine Richtung vorliegt.
- Zwei große Eigenwerte entsprechen Pixeln an Ecken und anderen Texturen, die gut geeignet sind, verfolgt zu werden.

Abschließend kann man festhalten, dass ein Bildpunkt gut zu verfolgen ist, wenn für die Eigenwerte λ_1, λ_2 der entsprechenden Matrix Z gilt:

$$\min(\lambda_1, \lambda_2) > \lambda$$

wobei λ ein festzulegender Schwellwert ist. Abbildung 3.5 zeigt die Ergebnisse des Algorithmus beispielhaft an Eingabebildern aus zwei Büros.

Bestimmung des optischen Flusses

Nachdem so geeignete Bildpunkte bestimmt wurden, die leicht in zwei aufeinanderfolgenden Bildern einander zugeordnet werden können, soll nun der Algorithmus nach Lucas und Kanade [LK81] vorgestellt werden, der dies bewerkstelligt. Es handelt sich dabei um ein iteratives Verfahren, das zum Verschiebungsvektor konvergiert.

Abbildung 3.5: *Bestimmung von Bildpunkten, die für die Extraktion des optischen Flusses geeignet sind*

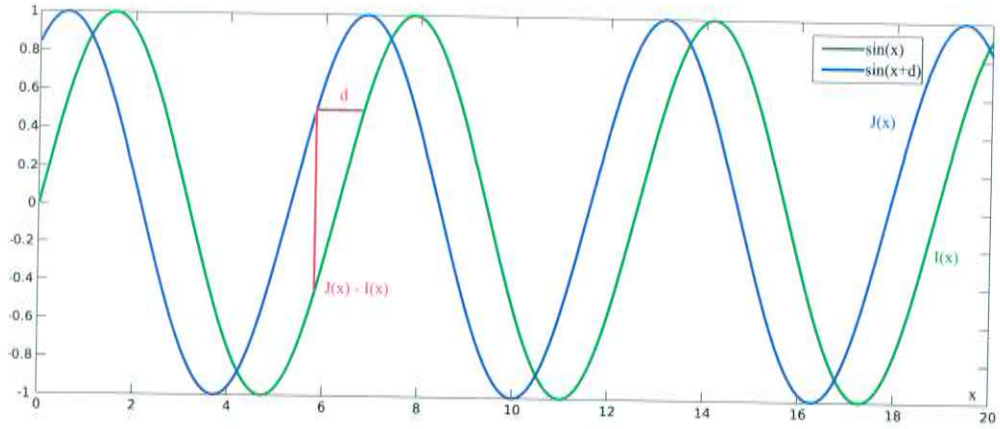

Abbildung 3.6: *Veranschaulichung des Registrierungsproblems im eindimensionalen Fall*

Ein Vorteil dieses Ansatzes besteht unter anderem darin, dass er eine mehrstufige Suche von grob nach fein zulässt. Die verwendete Implementierung von Bouget geht so vor, dass die Verschiebung zunächst auf niedrig aufgelösten Bildern berechnet wird. Das Ergebnis wird dann als Initialisierung auf der nächst höheren Auflösung verwendet und durch mehrfache Anwendung des nachfolgend beschriebenen Algorithmus schrittweise verfeinert. Der Anschaulichkeit wegen soll dies zunächst im eindimensionalen Fall erfolgen.
Die beiden Eingabesignale sollen weiterhin mit $I(x)$ und $J(x)$ bezeichnet werden, wobei gelten soll: $J(x) = I(x) + d$. Abbildung 3.6 zeigt dies veranschaulicht für $I(x) = \sin(x)$ mit $d = 1$. Für hinreichend kleine d gilt in diesem Fall:

$$I'(x) \approx \frac{J(x) - I(x)}{d}$$

und damit

$$d \approx \frac{J(x) - I(x)}{I'(x)}$$

Um diese Schätzung für verschiedene Positionen von x zu kombinieren, wird vorgeschlagen über alle Schätzungen eines Bereiches von x zu mitteln. Diese Schätzung kann noch dadurch verbessert werden, dass eine Gewichtungsfunktion $w(x)$ eingeführt wird, die die Einzelschätzungen in Abhängigkeit der Näherungsgüte von $I'(x)$ bewertet. Hierfür wird die Funktion

$$w(x) = \frac{1}{|J'(x) - I'(x)|}$$

vorgeschlagen. Insgesamt ergibt sich somit für die Schätzung der Verschiebung folgendes Verhältnis:

$$d \approx \frac{\sum_x \frac{w(x)[J(x)-I(x)]}{I'(x)}}{\sum_x w(x)}$$

Daraus lässt sich das folgende iterative Verfahren zur Bestimmung von d ableiten:

$$\begin{aligned} d_0 &= 0 \\ d_{k+1} &= d_k + \frac{\sum_x \frac{w(x)[J(x)-I(x+d_k)]}{I'(x+d_k)}}{\sum_x w(x)} \end{aligned}$$

Bei genauerer Betrachtung stellt man jedoch leider fest, dass dieser Algorithmus schlecht auf höhere Dimensionen verallgemeinert werden kann und dass es zu Problemen kommt falls $I'(x) = 0$. Verwendet man stattdessen die Näherung um den quadrierten Verschiebungsfehler

$$E = \sum_x (I(x+d) - J(x))^2$$

zu minimieren, so erhält man folgenden Algorithmus:

$$\begin{aligned} d_0 &= 0 \\ d_{k+1} &= d_k + \frac{\sum_x w(x) I'(x+d_k)[J(x) - I(x+d_k)]}{\sum_x w(x) I'(x+d_k)^2} \end{aligned}$$

Verallgemeinert auf mehrere Dimensionen erhält man schließlich die Näherung:

$$d_{k+1} \approx d_k + \left[\sum_{x \in R} w(x) \left.\frac{\partial I}{\partial x}\right|_{x+d_k} [J(x) - I(x+d_k)]\right] \left[\sum_{x \in R} w(x) \left(\frac{\partial I}{\partial x}\right) \left(\frac{\partial I}{\partial x}\right)^T\right]^{-1}$$

und einen entsprechenden Algorithmus wie oben. R bezeichnet dabei ein zweidimensionales Fenster um den Merkmalspunkt, für den der optische Fluss bestimmt werden soll. Schließlich sollte angemerkt werden, dass der Konvergenzbereich des Iterationsverfahrens noch vergrößert werden kann, indem hochfrequente Bildanteile unterdrückt werden, was einem Glätten des Bildes gleich kommt. Auf Grund der pyramidischen Implementierung neigt das Verfahren leider allerdings auch dazu, fatal falsche Verschiebungsvektoren zu liefern, falls bereits auf kleinen Skalen ein Schätzungsfehler unterläuft. Um dem Abhilfe zu schaffen, soll im Weiteren immer der Median des extrahierten optischen Flusses aus einem Nachbarschaftsbereich verwendet werden. Abbildung 3.7 zeigt die erfolgreiche Extraktion des optischen Flusses für einige Beispiele, Abbildung 3.8 den angesprochenen Fehlerfall.
Abschließend sollte bemerkt werden, dass für die Implementierung sowohl für die Merkmalsauswahl, als auch für Extraktion des optischen Flusses die Methoden der *Open Source Computer Vision Library* (OpenCV) [Int] benutzt wurden.

3.3 Lokale Merkmalsmodelle

Offensichtlich ist es wenig sinnvoll, die visuellen Merkmale für ein gesamtes Eingabebild global zu bestimmen, da dies eine schlechte Beschreibung des Bildinhalts darstellt. Würde man außerdem die Merkmale eines jeden Bildpunktes zu

Abbildung 3.7: *Extraktion des optischen Flusses für beispielhafte Bildausschnitte aus allen Büros; die linke Spalte zeigt die gewählten Merkmalspunkte, die rechte eine vergrößerte Darstellung der Merkmalsverschiebung mit Pfeilen*

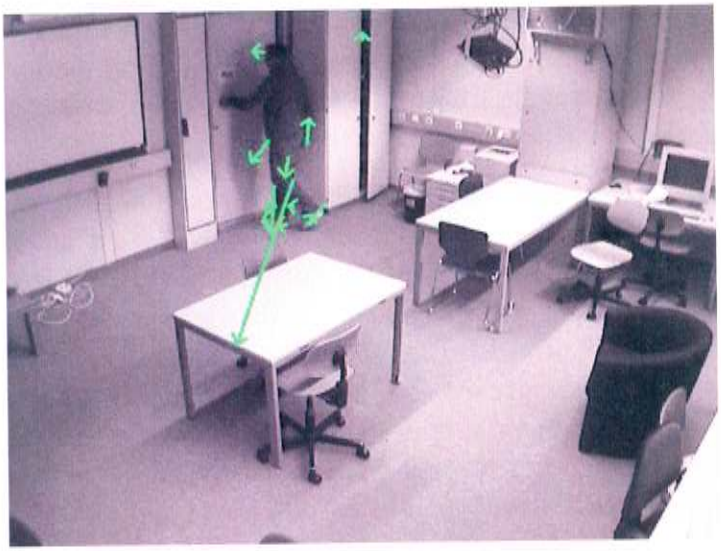

Abbildung 3.8: *Beispiel einer fehlerhaften Extraktion des optischen Flusses*

einem Eingabevektor für ein Hidden Markov Model zusammenfassen, so hätte man ein sehr hochdimensionales Lernproblem. Dies würde eine hohe Anzahl an zu lernenden Parametern nach sich ziehen, wofür wiederum eine große Menge an Trainingsdaten benötigt würde. Allein deshalb muss die Darstellung der Merkmale vereinfacht oder in anderen Worten komprimiert werden.
Grundsätzlich gibt es dafür meherere mögliche Herangehensweisen, die hier kurz erwähnt werden sollen:

- Es werden Algorithmen zur Verfolgung bewegter Bereiche eingesetzt und die Trajektorienpunkte als Merkmale weiterverwendet. Hierbei stellt sich allerdings die Frage, wie die Spuren mehrerer Personen zu einem Merkmalsvektor fester Länge kombiniert werden sollen. Des Weiteren gestaltet sich die Verfolgung in dieser unkontrollierten Versuchsumgebung wegen der einfachen Sensorausstattung außerordentlich schwierig. Vorwiegend auf Grund dieser beiden Punkte wurde auf den Einsatz von Verfolgungstechniken verzichtet.

- Die für eine Aktivität wichtigen Bildbereiche werden mit Hilfe einer Hauptachsentransformation bestimmt. Hierbei wird zunächst der Vordergrundbereich für die jeweilige Aktion bestimmt und das Vordergrundbild dann auf eine sinnvolle und vor allem berechenbare Größe verkleinert. Anschließend werden für die Trainingsbeispiele mit Hilfe der Hauptachsentransformation die Bildbereiche bestimmt, die von Bedeutung sind. Problematisch an diesem Ansatz ist leider allerdings, dass gleichzeitig stattfindende Aktivitäten nur schlecht voneinander getrennt werden können, was dazu führt, dass a-priori Abhängigkeiten zwischen diesen auftreten.

- Wichtige Bildbereiche werden von Hand markiert. Daran anschließend werden die Merkmale nur bezogen auf diese Ausschnitte extrahiert. Bei der

Auswahl stellt sich dabei die Frage, welche Bereiche eines Bildes von semantischer Bedeutung sind. Die Bestimmung dieser ist nicht immer ganz einfach und erfordert das manuelle Eingreifen eines Menschen, welchem der Einsatz einer automatischen Methode vorzuziehen ist.

- Zu guter Letzt kann dieser Einschränkung mit dem nachfolgenden, datengetriebenen Ansatz abgeholfen werden, der für diese Arbeit entwickelt wurde. Man trifft die Annahme, dass sich bedeutungsvolle Bildbereiche mit Hilfe einer Mixtur von Gaußverteilungen darstellen lassen, wobei Vordergrundbereiche hier als bedeutungsvoll gelten sollen. Anschließend werden die bestimmten Merkmale jeweils für eine Mixturkomponente kombiniert und als Merkmale für den Lernalgorithmus verwendet. Dieses Vorgehen hat den Vorteil, dass das Erlernen der lokalen Merkmalsbereiche automatisch vonstatten geht und nur ein Parameter, die Anzahl von Gaußverteilungen, gesetzt werden muss. Des Weiteren ermöglicht es die gleichzeitige Erkennung von Aktivitäten mehrerer Personen und soll aus diesen Gründen in dieser Arbeit zum Einsatz kommen.

Als lokale Bereiche, die von Bedeutung sind, werden also die k Komponenten einer Mixtur von Gaußverteilungen verstanden, die auf einer Menge von Vordergrundpunkten, gelernt werden kann. Dabei werden die Vordergrundpunkte zunächst mit dem in Abschnitt 3.2.1 beschriebenen Algorithmus bestimmt, wobei darauf zu achten ist, dass von jeder Aktivität der ersten Erkennungsstufe, die später erkannt werden soll, die gleiche Menge von Trainingsdaten zur Verfügung steht.
Der verwendete Expectation-Maximization Lernalgorithmus, der die Parameter der Verteilung $\Theta = \{\Sigma_1, \mu_1, w_1, \ldots, \Sigma_k, \mu_k, w_k\}$ lernt, wird in Abschnitt 2.2 erläutert. Als Ergebnis erhält man die Wahrscheinlichkeitsverteilung dafür, dass ein Bildpunkt x zum Vordergrund gehört, angenähert durch die gewichtete Summe von k Normalverteilungen:

$$P_{fg}(X = x|\Theta) = \sum_{i=1}^{k} w_i \cdot \frac{1}{2\pi|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

Hierbei bezeichnen Σ_i, μ_i und w_i die Kovarianz, das Mittel und das entsprechende Gewicht der i-ten Normalverteilung. Um einen besseren Eindruck zu vermitteln, zeigt Abbildung 3.9 die gelernten Verteilungen für die verschiedenen Räume, wobei die Normalverteilungen ins Bild projiziert wurden. Die abgebildeten Ellipsen veranschaulichen die ein-, zwei-, und dreifache Standardabweichung. Im Folgenden sollen die visuellen Merkmale nun jeweils für den Bereich einer Normalverteilung in den Grenzen von drei Standardabweichungen extrahiert werden. Das heißt die Vordergrundbereiche und der optische Fluss werden lokal für alle Bildpunkte einer Mixturkomponente i innerhalb von drei Standardabweichungen berechnet, so dass für die i-te Normalverteilung alle Bildpunkte M_i mit einer Mahalanobisdistanz von kleiner als drei von Bedeutung sind:

$$M_i = \{m \in \mathbb{N}^2 | (m - \mu_i)^T \Sigma_i^{-1} (m - \mu_i) \leq 3\}$$

Abbildung 3.9: *Lokale Merkmalsmodelle; die linke Spalte zeigt beispielhaft für jeden Raum eine Mixturkomponente an markanten Stellen, die rechte Spalte alle Mixturkomponente für einen Raum*

Schließlich müssen die für die Mengen M_i extrahierten Merkmale noch in geeigneter Weise kombiniert werden, so dass der Eingabevektor, der für die unterste Schicht der Hidden-Markov-Model-Hierarchie verwendet wird, aus folgenden Komponenten aufgebaut wird:

- Kumulierte Vordergrundmasse, die sich aus der Summe der Distanzen zum Hintergrundmodell bestimmt:

$$cd_i = \sum_{j=1}^{|M_i|} fg_t(m_j)$$

  Dieses Merkmal charakterisiert wie stark sich innerhalb eines lokalen Bereiches das Hintergrundmodell vom beobachteten Eingabebild unterscheidet. Kommt der Vordergrund nur durch Schattenwurf zustande, wird cd_i kleine Werte annehmen, wohingegen große Werte auftreten wenn sich etwa zwei Personen in einem Bereich befinden.

- Gemeinsame Wahrscheinlichkeitsdichte aus dem lokalen Merkmalsmodell eingeschränkt auf die Vordergrundpunkte $F_i = M_i \cap \{m \in \mathbb{N}^2 | fg_t(m) > 0\}$ der Normalverteilung i:

$$jp_i = \prod_{j=1}^{|F_i|} \frac{1}{2\pi|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x_j-\mu_i)^T\Sigma_i^{-1}(x_j-\mu_i)}$$

  Hiermit wird einerseits nochmals die Anzahl der Vordergrundpunkte innerhalb eines lokalen Merkmalsbereichs charakterisiert und anderseits hat die genaue Position der Vordergrundpunkte innerhalb des Bereichs Einfluss auf den Wert von jp_i.

- Schließlich wird der Median des optischen Flusses jeweils in x und y Richtung (of_i^x und of_i^y) verwendet, welcher in der Lage ist die dominierende Bewegungsrichtung innerhalb eines lokalen Merkmalsbereiches zu beschreiben.

Folglich besteht der visuelle Merkmalsvektor o_t^v, der in die unterste HMM-Schicht geführt wird, aus $k \cdot 4$ Komponenten:

$$o_t^v = \{cd_1, jp_1, of_1^x, of_1^y, \ldots, cd_k, jp_k, of_k^x, of_k^y\}^T$$

Abschließend zur Einführung der Merkmale soll noch die folgende Notation vereinbart werden. Um die Vektoren eines Zeitintervalls von t_1 bis t_2 zu bezeichnen, soll die Schreibweise $o_{t_1:t_2}^v$ für visuelle Merkmale und $o_{t_1:t_2}^a$ für Audiomerkmale verwendet werden.

4 Probabilistische Modelle

Das folgende Kapitel befasst sich mit den eigentlichen Methoden zur Erkennung von Aktivitäten und der Verfolgung von Personen über mehrere Räume hinweg. Zur Erkennung von Büroaktivitäten kommt dabei ein mehrschichtiges Hidden Markov Model zur Anwendung (Abschnitt 4.1), für die Verfolgung von Personen ein Bayes'scher Filter (Abschnitt 4.2).

4.1 Mehrschichtige Hidden Markov Models

Wie bereits aus dem einleitenden Kapitel ersichtlich wurde, sind HMMs ein beliebtes Rahmenwerk zur Erkennung von Aktivitäten. Gleichwohl gibt es eine Vielzahl von Ansätzen, die sich entweder durch die eingesetzten Merkmale unterscheiden oder durch den genauen Aufbau der Modellstruktur.
Einige Arbeiten verwenden etwa gekoppelte Modelle [ORP00] oder modellieren die Aufenthaltsdauer in einem Zustand explizit [DBPV05]. Bei der Wahl der Merkmale gibt es Ansätze, die wie diese Arbeit direkt auf Signalmerkmalen aufbauen, oder aber bereits auf das Ergebnis komplexer Vorverarbeitungen wie zum Beispiel auf Trajektorien zurückgreifen [HHE03]. Auch bezüglich der Lernalgorithmen kann unterschieden werden. Brand *et al.* [BK00] zeigen in ihrer Arbeit, dass ihr Verfahren, das auf Entropieminimierung basiert, Zustände liefert, die von Menschen besser gedeutet werden können und so der Lernprozess besser überprüfbar wird.
Außerdem gibt es verschiedene Ansätze, um mehrere Modalitäten zu kombinieren, wofür McCowen *et al.* [MGPB+05] einen guten Überblick geben. Die einfachste Möglichkeit besteht darin, alle beobachteten Merkmale zu einem großen Merkmalsvektor zusammenzufassen, was in der Fachliteratur als *Early Integration* bezeichnet wird.
Für diese Arbeit soll ein mehrschichtiges Hidden Markov Model zur Anwendung kommen, bei dem die Erkennung in mehrere Teilprobleme zerlegt wird, die aufeinander aufbauen. Auf der untersten Ebene soll das Audiosignal in gesprochene Sprache und Hintergrundgeräusch klassifiziert werden. Für die Videomodalität besteht die Erkennungsaufgabe darin, Personen in verschiedenen Bereichen des Raumes wahrzunehmen und festzustellen, ob jemand den Raum betritt oder verlässt. Abbildung 4.1 verdeutlicht diesen Ansatz nochmals bildlich.

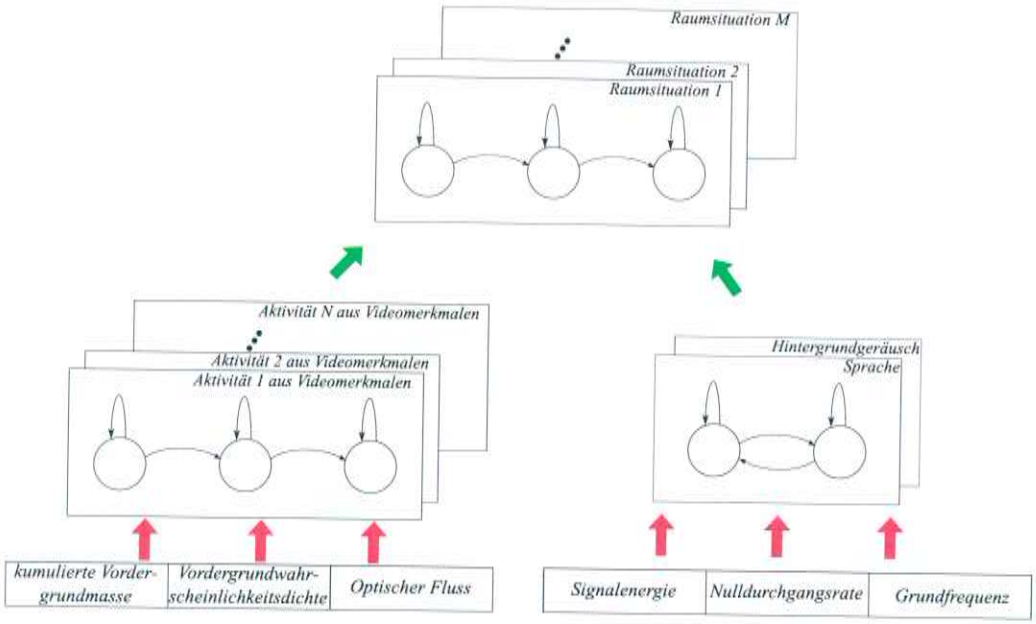

Abbildung 4.1: *Aufbau der mehrschichtigen Hidden-Markov-Model-Architektur*

Die Aufteilung des Merkmalraums in jeweils eine Menge von Audio- und Videomerkmalen hat einerseits den Vorteil, dass die Menge von benötigten Trainingsdaten kleiner wird und anderseits verbessert sich die Intuition bezüglich des Lernprozesses. So werden etwa zunächst auf eher syntaktischer Ebene die Konstellation von Personen im Raum und Sprache erkannt und später werden diese Informationen zu einer semantischen Bedeutung verknüpft.
Weitere Vorteile der Unterteilung des Erkennungsprozesses entstehen dadurch, dass jede Schicht separat mit dem Baum-Welch Algorithmus trainiert werden kann und Veränderungen in einer Schicht sich nur auf darüber liegende Ebenen auswirken. Außerdem ist es möglich, für jede Schicht die Länge der Eingabesequenz, die typischerweise von Schicht zu Schicht wächst, neu zu bestimmen. Niedrige Ebenen verwenden kurze Eingabesequenzen von wenigen Sekunden, während höhere Ebenen Eingabevektoren verwenden, die eine vielfache Länge besitzen.

4.1.1 Allgemeine formale Beschreibung

Auf der untersten Ebene besteht das mehrschichtige HMM einerseits aus einer Menge $V^1 = \{V_1^1, \ldots V_N^1\}$ von Video-HMMs, die dazu dienen, Aktivitäten einzelner Personen visuell zu erkennen. Dabei wird für jede zu erkennende Aktivität ein Hidden Markov Model trainiert. Außerdem wird jeweils ein HMM trainiert, um Sprache und Hintergrundgeräusche zu erkennen. Diese werden entsprechend als A_1^1 und A_2^1 bezeichnet und zur Menge der Audio-HMMs $A^1 = \{A_1^1, A_2^1\}$ zusammengefasst. Die Aktivitäten, die auf der untersten Erkennungsschicht erkannt werden, werden auch als *Ereignisse* bezeichnet.

Die Eingabevektoren für die Menge V^1 bestehen aus $o^v_{t-i_1:t}$, die der Menge A^1 aus $o^a_{t-i_1:t}$. Die Ausgabewahrscheinlichkeiten, die man durch die Auswertung der Eingabesequenzen nach dem in Abschnitt 2.1.4 beschriebenen Verfahren erhält, werden folglich als $P(o^v_{t-i_1:t}|V^1_i)$ beziehungsweise als $P(o^a_{t-i_1:t}|A^1_i)$ benannt. i_1 stellt dabei die zeitliche Fenstergröße des Eingabevektors dar.
Die einzelnen HMM-Schichten sind dadurch verknüpft, dass ein HMM auf der Ebene $l+1$ die Inferenzresultate der Schicht l als Eingabe- oder Beobachtungsvektor erhält. Das heißt für den zweiten Level des vorgestellten Systems dient eine Kombination aus $P(o^v_{t-i_1:t}|V^1_i)$ und $P(o^a_{t-i_1:t}|A^1_i)$ als Beobachtungssequenz. Die Hidden Markov Models der zweiten Schicht, die dazu dienen, *Situationen* eines Raumes zu erfassen, werden im Folgenden mit $S^2 = \{S^2_1, \ldots, S^2_M\}$ notiert. Für die Verknüpfung der Schichten kann zwischen zwei grundsätzlichen Ansätzen unterschieden werden:

Harte Entscheidung: Bei diesem Ansatz wird bei der Weitergabe der Resultate der Schicht l eine harte Entscheidung getroffen. Handelt es sich um eine 1-aus-n Klassifikation, wird die Ausgabe des Modells mit der höchsten Wahrscheinlichkeit auf einen von Null verschiedenen Wert gesetzt und alle anderen Ausgaben zu Null.
Für den hier vorliegenden Fall, bei dem auf der untersten Ebene auch mehrere Aktivitäten gleichzeitig erkannt werden können, bedeutet dieser Ansatz, dass die Ausgabe für alle erkannten Ereignisse auf einen von null verschiedenen Wert gesetzt werden, wobei eine Aktivität als erkannt gilt, wenn deren zugeordnete Ausgabewahrscheinlichkeit einen festzusetzenden Schwellwert übersteigt. Alle entsprechenden Beobachtungswahrscheinlichkeiten nicht erkannter Aktivitäten werden dann zu Null gesetzt.

Weiche Entscheidung: Diese Methode erhält die Verteilung der Beobachtungswahrscheinlichkeiten und gibt diese direkt an die nächste Schicht $l+1$ weiter.

Für das vorliegende System wurde entschieden, weiche Entscheidungen für die Verbindung der beiden HMM-Ebenen zu treffen, da dadurch vermieden wird, auf niedrigen Ebenen bereits Entscheidungen fällen zu müssen, deren Fehler sich in späteren Schichten fortpflanzen könnten. Außerdem entfällt, wie oben erklärt, die Bestimmung von Schwellwerten, die nötig wären, um zu entscheiden ob ein Ereignis stattfindet. Dieses Vorgehen ist bereits aus verwandten Arbeiten [MGPB+05, OHG02] bekannt und hat sich dort bewährt.

4.1.2 Training und Inferenz

Der folgende Abschnitt soll nun die Details des Vorgehens beim Training der Hidden Markov Models darlegen. Dazu sei zunächst auf Tabelle 4.1 verwiesen, die für Büro B die Klassifikationsziele jeweils getrennt nach unterer und oberer Schicht auflistet. Eine vollständige Auflistung aller zu erkennenden Aktivitäten findet sich

in Anhang A. Es sollte bemerkt werde, dass sich die zu erkennenden Klassen von Büro zu Büro auf Grund des jeweils typischen Geschehens unterscheiden. Eine genaue Definition der Klassenbeschreibung findet sich in den jeweiligen Abschnitten zu einer Erkennungsebene in Kapitel 5.

Untere Ebene (Video)
Jemand sitzt am Schreibtisch von Benutzer 3
Jemand sitzt am Schreibtisch von Benutzer 4
Besucher hinter Schreibtisch von Benutzer 3
Besucher hinter Schreibtisch von Benutzer 4
Jemand sitzt auf dem Besucherstuhl
Jemand kommt herein (Haupttüre)
Jemand geht hinaus (Haupttüre)
Jemand kommt herein (Seitentüre)
Jemand geht hinaus (Seitentüre)

Untere Ebene (Audio)
Gespräch findet statt
Hintergrundgeräusche

Obere Ebene
Niemand im Büro
Schreibtischarbeit
Diskussion
Besprechung

Tabelle 4.1: *Klassifikationsziele für Büro B getrennt nach unterer Schicht (linke Spalte) und oberer Schicht (rechte Spalte)*

Topologiewahl: Die Topologie der trainierten HMMs ist abhängig von der zu erkennenden Aktivität und wurde von Hand bestimmt. Die HMMs, die stationäre Aktivitäten ohne zeitlichen Verlauf beschreiben, wurden mit ergodischer Topologie und fester Eingabesequenzlänge trainiert. Beispielhaft für ein Ereignis dieser Art wäre, eine Person an einem Schreibtisch zu detektieren. Das heißt gleichzeitig, dass lange andauernde Aktivitäten in mehrere gleich lange Teilblöcke zerteilt werden mussten. Um bei der Inferenz Aktivitäten zu erkennen, wurden ebenfalls Fenster gleicher zeitlicher Länge verwendet.
HMMs mit Links-Rechts-Topologie, die im Allgemeinen kurze Aktivitäten im zeitlichen Verlauf beschreiben, wie zum Beispiel das Verlassen eines Raumes, wurden mit Eingabesequenzen in der Länge der tatsächlichen Dauer der Aktivität trainiert. Für die spätere Erkennung wurde die mittlere Dauer als Zeitspanne des Eingabevektors gewählt.
Für die Situationen S, die auf der zweiten Ebene erkannt werden sollen, wurden durchweg ergodische Modelle verwendet.

Verarbeitung der Audiomerkmale: Für jeden Raum wurden jeweils zwei ergodische Modelle trainiert, um Sprache von Hintergrundgeräuschen zu unterscheiden. Nachdem aus Gründen der Privatsphäre bei der Datensammlung keine Audiodaten in Rohform, sondern nur extrahierte Merkmale gespeichert

werden konnten, wurden hierfür jeweils zirka 15 Minuten Rohdaten für jedes Zimmer, speziell für das Training der HMMs, gesammelt. Als Eingabe für die nächste Ebene diente dann das Verhältnis der Ausgabewahrscheinlichkeiten

$$\frac{P(o^a_{t-i_1:t}|A^1_1 = \text{Sprache})}{P(o^a_{t-i_1:t}|A^1_2 = \text{Hintergrundgeräusch})}$$

beziehungsweise die Differenz der Log-Wahrscheinlichkeiten:

$$\log P(o^a_{t-i_1:t}|A^1_1 = \text{Sprache}) - \log P(o^a_{t-i_1:t}|A^1_2 = \text{Hintergrundgeräusch})$$

Verbesserung der Separabilität: Da ein einzelnes HMM auf der ersten Ebene oft nicht ausreichte, um bestimmte Aktivitäten befriedigend zu detektieren, wurden für alle Aktivitäten der ersten Ebene neben Modellen, die Ereignisse erkennen, noch zusätzlich HMMs trainiert, die erkennen, dass diese nicht stattfinden. Diese wurden mit Gegenbeispielen trainiert und werden im Weiteren als $\overline{V}_i = \{\overline{V}^1_1, \ldots \overline{V}^1_N\}$ bezeichnet. Auch hier wird wie für die Audioklassifikation das Verhältnis der Ausgabewahrscheinlichkeiten

$$\frac{P(o^v_{t-i_1:t}|V^1_i)}{P(o^v_{t-i_1:t}|\overline{V}^1_i)}$$

beziehungsweise

$$R^v_{i,t} = \log P(o^v_{t-i_1:t}|V^1_i) - \log P(o^v_{t-i_1:t}|\overline{V}^1_i)$$

an die nächste Erkennungsebene weitergereicht.

Training mit Cross-Validation: Da beim Training die Anzahl der zu verwendenden Zustände und die passende Anzahl von Gaußverteilungen pro Zustand a priori unbekannt sind, wurden diese mit Hilfe von *5 fold Cross-Validation* bestimmt. Dabei handelt es sich um eine Technik, mit der in einem Vorversuch diese Unbekannten experimentell bestimmt werden. Hierfür werden jeweils die insgesamt zur Verfügung stehenden Trainingsdaten in fünf gleich große Teilmengen unterteilt. Davon werden vier dazu benutzt, um ein HMM zu trainieren und anschließend die verbleibende dazu, die Erkennungsrate zu bestimmen. Daraufhin wird auf der nächsten Teilmenge evaluiert, solange bis alle Teilmengen einmal zur Evaluation herangezogen wurden. Nachdem man alle in Betracht kommenden HMM-Konfigurationen probiert hat, wird diejenige ausgewählt, die die beste durchschnittliche Erkennungsrate aufweisen kann.
Es gilt allerdings, die Anzahl der Zustände und der Gaußverteilungen pro Zustand zu beschränken, da es sonst zu *Overfitting* kommt, was bedeutet dass das Modell nicht mehr generalisiert, sondern nur einzelne Trainingssequenzen auswendig lernt.

Abbildung 4.2: *Auswahl signifikanter Merkmalskomponenten für die Ereignisse „Jemand sitzt am Schreibtisch von Benutzer 3" (Büro B) und „Jemand kommt herein (Seitentüre)" (Büro A)*

Merkmalsnormierung: Da die Einzelmerkmale auf der untersten Ebene einen sehr unterschiedlichen Wertebereich haben, hat es sich außerdem als günstig herausgestellt, diese zu normieren. Dies geschieht in der Form, dass für jedes Merkmal über alle Trainingsbeispiele m_j eines Ereignisses das Mittel μ_i und die Standardabweichung σ_i für jedes Einzelmerkmal i bestimmt wird. Die ursprünglichen Merkmalseinträge $m_{j,i}$ werden dann wie folgt zu $\hat{m}_{j,i}$ transformiert und für alle weiteren Prozessschritte weiterverwendet:

$$\hat{m}_{j,i} = \frac{m_{j,i} - \mu_i}{\sigma_i}$$

Merkmalsauswahl: Verwendet man für jede Aktivität auf der ersten Ebene den vollständigen Merkmalsvektor, so wie oben beschrieben, stößt man auf das Problem, dass auf Grund der begrenzt vorhandenen Daten a-priori Abhängigkeiten eingelernt werden, was dazu führt, dass seltene Kombinationen von Aktivitäten falsch erkannt werden. Zum Beispiel verhindert eine durch die Tür kommende Person die weitere Erkennung von am Tisch sitzenden Benutzern. Hätte man unendlich viele Daten zur Verfügung, die alle Kombinationen von gleichzeitig auftretenden Aktivitäten beinhalten würden, würde dieses Problem nicht auftreten. Dies lässt sich allerdings dadurch simulieren, dass man die Komponenten des Merkmalsvektors, die für die Erkennung einer Aktivität nicht signifikant sind, mit weißem Rauschen überlagert. Die Mixturkomponenten des lokalen Merkmalsmodells, die diesen entsprechen, werden in einem separaten Vorversuch bestimmt. Dabei werden diejenigen Mixturkomponenten für eine Situation als bedeutungsvoll angesehen, die nötig sind, um mindestens 80 Prozent des segmentierten Vordergrundes abzudecken. Abbildung 4.2 zeigt die ausgewählten Komponenten beispielhaft für zwei Ereignisse.

Inferenz: Die Erkennung von Aktivitäten stellt genau das in Abschnitt 2.1.4 beschriebene Evaluierungsproblem dar. Wie bereits kurz erwähnt, dienen als Eingabe hierfür die Merkmale, die in einem Fenster, das die Merkmale der letzten Zeitschritte umfasst, extrahiert wurden. Die Fenstergröße richtet sich nach den im Punkt *Topologiewahl* beschriebenen Kriterien. Für die Erkennung von Raumsituationen auf dem zweiten Level von HMMs wurde jeweils für alle HMMs der gleiche Merkmalsvektor mit der gleichen Eingabesequenzlänge verwendet. Dadurch sind die ausgegebenen Beobachtungswahrscheinlichkeiten der verschiedenen Modelle vergleichbar und als Klassifikationsergebnis kann einfach die Situation gewählt werden, der das Modell mit der höchsten Beobachtungswahrscheinlichkeit zugeordnet werden kann.

4.2 Bayes'scher Filter

Bayes'sche Filter stellen ein sehr effizientes, statistisches Mittel dar, um Objekte und Personen zu lokalisieren und zu verfolgen. Das Rahmenwerk stellt ein wirkungsvolles Vorgehen gegen Messrauschen und Messausfälle dar und hat sich in verschiedensten Implementierungsformen, sei es als Kalmanfilter [Kal60, May79] oder Partikelfilter [IB98, DdG01] für Verfolgungsaufgaben durchgesetzt. Fox *et al.* [FHL+03] geben hierzu eine gute Einführung und verweisen auf weiterführende Literatur. Der nachfolgende Abschnitt fasst kurz die Eigenschaften eines Bayes'schen Filters zusammen, bevor danach auf die Implementierung in der Büroumgebung eingegangen wird.

4.2.1 Formale Darstellung

Zunächst soll aus Gründen der Einfachheit nur das Verfolgen eines einzelnen Objektes betrachtet werden. Ein Bayes'scher Filter schätzt einen Systemzustand x_t zum Zeitpunkt t auf der Basis der bis dahin gemachten Beobachtungen. Diese werden für gewöhnlich mit $z_0, \ldots, z_t$ bezeichnet. Anders ausgedrückt ist diese Schätzung die bedingte Wahrscheinlichkeit $P(x_t|z_0, \ldots, z_t)$, sich in einem Zustand x_t zu befinden, wenn die Messungen bekannt sind. Da diese für fortschreitende Zeit auf Grund der hohen Anzahl der zu berücksichtigenden Beobachtungen praktisch nicht berechenbar ist, ist man gezwungen die Markov-Annahme zu machen. Diese besagt hier, dass der Systemzustand x_t nur von x_{t-1} und der aktuellen Beobachtung z_t abhängig ist. Daraus lässt sich die rekursive Aktualisierungsregel für die Schätzung eines diskreten Zustands formulieren:

$$p(X_t = x_t|z_t) = k_t \cdot p(z_t|X_t = x_t) \cdot \sum_{x' \in X} p(X_t = x_t|X_{t-1} = x')p(X_{t-1} = x'|z_{t-1})$$

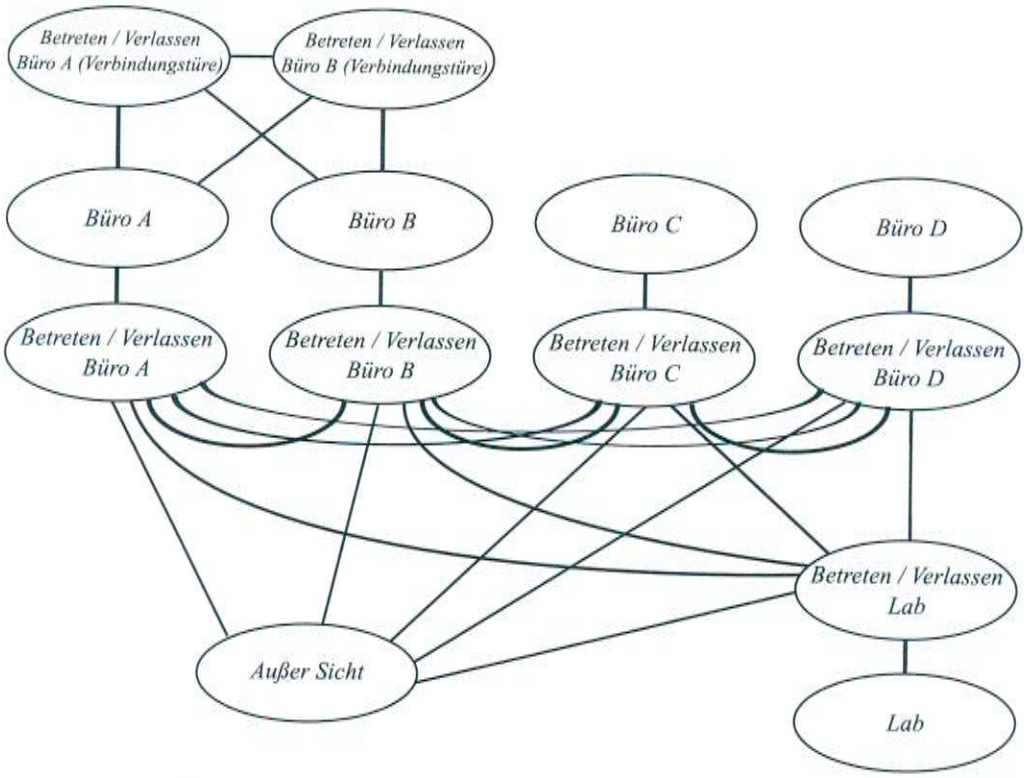

Abbildung 4.3: *Zustandsraum für das Verfolgen von Benutzern in der Büroumgebung; erlaubte Zustandsübergänge sind durch Kanten markiert; Selbstübergänge sind der Übersichtlichkeit wegen nicht eingezeichnet.*

Mit X_t werden hier alle Zustände bezeichnet, die das System einnehmen kann, während k_t den Normierungsfaktor bezeichnet, der notwendig ist, um $p(X_t = x_t|z_t)$ als Wahrscheinlichkeitsdichte zu erhalten. $p(X_t = x_t|X_{t-1} = x')$ bezeichnet hierbei die *Systemdynamik*, also wie sich das System über die Zeit verändert. Bei Verfolgungsaufgaben wird diese auch als das *Bewegungsmodell* bezeichnet, da es angibt, wo sich eine Person bei bekannter Position zum Zeitpunkt $t-1$ zum Zeitpunkt t befindet.
Das *Beobachtungsmodell* wird durch $p(z_t|X_t = x_t)$ beschrieben und setzt die Messungen in Bezug zur Schätzung des dynamischen Modells.
Für die konkrete Anwendung soll der Zustandsvektor x_t für jeden Raum die Wahrscheinlichkeit enthalten, dass ein verfolgter Benutzer sich in diesem aufhält. Zusätzlich wird jeweils für das Passieren von Türen ein Zustand eingefügt. Und schließlich wird noch ein Zustand benötigt, der die Tatsache modelliert, dass ein Benutzer die Umgebung verlassen hat, also nicht beobachtet werden kann. Der gesamte Zustandsraum mit allen erlaubten Zustandsübergängen ist in Abbildung 4.3 gezeigt. Der Beobachtungsvektor z_t setzt sich aus den Beobachtungswahrscheinlichkeiten der ersten Ebene der mehrschichtigen HMM-Architektur zusammen. Details hierzu finden sich im Abschnitt 4.2.3.
Etwas allgemeiner formuliert handelt es sich bei diesem Ansatz um ein Bayes'sches

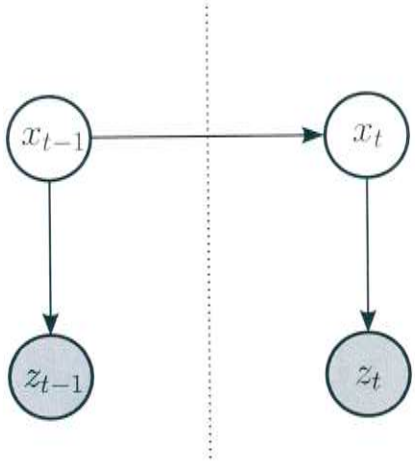

Abbildung 4.4: *Bayes'sches Netz eines Bayes'schen Filters zur Objektverfolgung; beobachtbare Variablen sind grau hinterlegt (nach [WA05]).*

Netz, das in Abbildung 4.4 dargestellt wird. Für die betrachtete Umgebung sind keine Näherungsverfahren wie Partikelfilter von Nöten, da der Zustandsraum sehr klein ist und deshalb obige rekursive Aktualisierungsregel analytisch für jeden Zustand berechnet werden kann.
Dieses Rahmenwerk lässt sich nun auf die Verfolgung beliebig vieler Objekte erweitern. Zunächst ist dabei zu entscheiden, ob für alle Objekteinzelzustände ein gemeinsamer Statusvektor zu verwenden ist oder jedes Objekt mit einem eigenen, von den anderen unabhängigen Statusvektor versehen wird. Im implementierten System wurde letztere Möglichkeit gewählt, da die Modellierung von Abhängigkeiten zwischen Personen im *dynamischen Modell* nicht notwendig ist und somit nur den zu erlernenden Parameterraum vergrößern würde.
Als weiteres Problem stellt sich bei der Erweiterung auf die Verfolgung von mehreren Personen die Zuordnung der Beobachtungen zu den richtigen Personen heraus. Dies ist absolut notwendig, da Beobachtungen, die durch fremde Personen verursacht wurden, kein Rauschen des unterliegenden stochastischen Prozesses darstellen und somit die Verfolgung stören würden. Hierfür kommt ein *Nearest Neighbor Standard Filter* zum Einsatz, der vorsieht, für die Verfolgung einer Person nur die Beobachtungen aus Nachbarzuständen des aktuell geschätzten Systemzustandes heranzuziehen. Bezeichnet man die Zuordnung der Beobachtungen am Zeitpunkt t zu verfolgten Objekten als Zufallsvariable θ_t, so kann die Modellierung als Bayes'sches Netz, wie in Abbildung 4.5 gezeigt, erweitert werden.

4.2.2 Dynamisches Modell

Wie bereits erwähnt, bezeichnet das dynamische Modell das Bewegungsverhalten der einzelnen Benutzer. In dieser Anwendung wurde es anhand von manuell annotierten Daten erlernt. Die Trainingsmenge umfasste dabei vier Tage, für die an je-

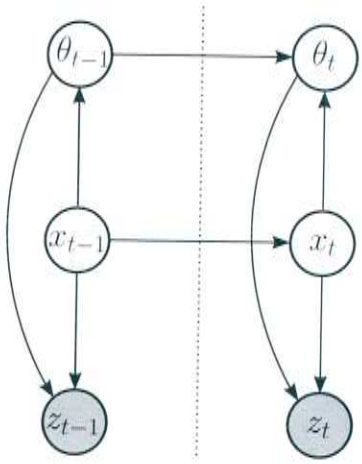

Abbildung 4.5: *Bayes'sches Netz eines Bayes'schen Filters zur Objektverfolgung für mehrere Personen und Datenzuordnung θ_t; beobachtbare Variablen sind grau hinterlegt (nach [WA05]).*

dem aufgenommenen Bild der tatsächliche Aufenthaltsort markiert wurde. Das Bewegungsmodell entsteht dann durch das Abzählen sämtlicher Zustandsübergänge, die stattgefunden haben. Es sei X_t der Zustand zum Zeitpunkt t und X_{t+1} entsprechend der Folgezustand. Die Menge aller Übergänge von Zustand x nach Zustand x' werde mit $c_{xx'} = \{t|X_t = x \wedge X_{t+1} = x'\}$ bezeichnet. Dann errechnet sich die Übergangswahrscheinlickeit wie folgt:

$$p(X_t = x'|X_{t-1} = x) = \frac{|c_{xx'}|}{\sum\limits_{y \in X} |c_{xy}|}$$

Bei der Berechnung wurde für die Zustände, die den Aufenthalt in Räumen und das Verlassen eines Zimmers modellieren, der Durchschnittswert über alle Räume verwendet, um die Wahrscheinlichkeit des Selbstübergangs beziehungsweise des Verlassens eines Raumes zu bestimmen. Für den Zustand „Außer Sicht“ wird bei der Berechnung der Übergangswahrscheinlichkeiten kein Mittelwert gebildet, sondern zwischen den einzelnen Büros unterschieden, da diese auf das individuelle Bewegungsprofil eines Nutzers den meisten Einfluss haben.

4.2.3 Beobachtungsmodell

Wie oben kurz bemerkt, werden für das Beobachtungsmodell die Erkennungswahrscheinlichkeiten der ersten Schicht von HMMs erneut verwendet. Dabei werden die Wahrschlichkeitsverhältnisse $R^v_{n,t}$ mit einem Schwellwert m_n versehen, so dass man einen binären Sensor e_n erhält.

$$e_n = \begin{cases} 1 & \text{falls } R^v_{n,t} > m_n \\ \epsilon & \text{sonst} \end{cases}$$

Folglich besteht der Beobachtungsvektor z_t aus binären Einträgen $\{e_1, \ldots, e_N\}$, die angeben, ob ein Ereignis stattfindet oder nicht. In diesem Fall bezeichnet N die Zahl der insgesamt vorhandenen Sensoren.
Des Weiteren soll für das Beobachtungsmodell die Information genutzt werden, wer an welchem Platz normalerweise arbeitet und wo im Raum sich gewöhnlich Besucher aufhalten. Der Sensor, der misst, ob ein Benutzer sich an seinem Arbeitsplatz befindet, soll mit h bezeichnet werden. Alle Sensoren, die nicht zur Beobachtung von Türereignissen und den Arbeitsplätzen der Benutzer dienen, stehen folglich den Besuchern zur Verfügung und werden mit v_i^k bezeichnet. Dabei bezeichnet i den Raum und k die verschiedenen Besucherplätze innerhalb eines Raumes. Die Menge aller B_i Besucherplatzsensoren in einem Raum i wird als $V_i = \{v_i^1, \ldots, v_i^{B_i}\}$ benannt. Schließlich sei für einen Zustand x die Zustandsmenge $L^x = \{L_1^x, \ldots, L_{T_x}^x\}$ definiert als die Menge aller T_x Zustände, die mit dem Betreten oder Verlassen eines Raumes assoziert sind und direkt mit x verbunden sind.
Im Folgenden sind zwei Fälle zu unterscheiden:

1. Der auszuwertende Zustand, im Folgenden auch *Urzustand* H genannt, gehört zum Raum, in dem der Benutzer arbeitet. Es ist davon auszugehen, dass er, wenn er anwesend ist, auf dem eigenen Platz sitzt, so dass nur der Sensor h und die Türereignisse $e_{L_l^H}$ von Bedeutung sind.

2. Das Beobachtungsmodell soll für einen Zustand ausgewertet werden, der einem Raum entspricht, in dem der verfolgte Benutzer Besucher ist. Die Menge dieser K *Besucherzustände* soll im Folgenden als $\beta = \{\beta_1, \ldots, \beta_K\}$ bezeichnet werden und die Auswertung soll nur die Informationen aus V_i und die Türereignisse $e_{L_l^{\beta_j}}$ verwenden.

Für den Urzustand soll die Beobachtungswahrscheinlichkeit also 1 werden, falls der Sensor h ausgelöst wird und ϵ sonst:

$$p(z_t|X_t = H) = e_h$$

Für die angrenzenden Zustände L^H, die mit dem Passieren einer Türe assoziiert sind, soll die Beobachtungswahrscheinlichkeit abhängig vom erkannten Türereignis und einem Zustandswechsel auf dem Urzustand sein. Nur falls beides eintritt, soll die Beobachtungswahrscheinlichkeit verschieden von ϵ werden. Dies wird mathematisch ausgedrückt durch:

$$p(z_t|X_t = L_l^H) = p(z_{Stuhl,t}|X_t = L_l^H) \cdot p(z_{Tuere,t}|X_t = L_l^H)$$

wobei $p(z_{Stuhl,t}|X_t = L_l^H)$ und $p(z_{Tuere,t}|X_t = L_l^H)$ durch

$$\begin{aligned} p(z_{Stuhl,t}|X_t = L_l^H) &= \begin{cases} 1 & \text{falls } \frac{\partial p(z_t|X_t=H)}{\partial t} \neq 0 \\ \epsilon & \text{sonst} \end{cases} \\ p(z_{Tuere,t}|X_t = L_l^H) &= e_{L_l^H} \end{aligned}$$

bestimmt werden. Die Ableitung $\frac{\partial p(z_t|X_t=H)}{\partial t}$ wird hierbei durch einen Differenzenquotienten approximiert. Die Beobachtungswahrscheinlichkeiten für Besucherzustände β_j und deren angrenzende Türzustände L^{β_j} ergeben sich analog hierzu. Die Messung der verschiedenen Besucherplatzsensoren V_{β_j} werden dabei so kombiniert, dass die Beobachtungswahrscheinlichkeit im entsprechenden Raum dann verschieden von ϵ wird, sobald einer der Sensoren ein Ereignis registriert.

$$p(z_t|X_t = \beta_j) = \max_k e_{v^k_{\beta_j}}$$

Dementsprechend werden die Zustände für das Passieren einer Türe l in Besucherzuständen $L_l^{\beta_j}$ folgendermaßen angepasst:

$$\begin{aligned} p(z_t|X_t = L_l^{\beta_j}) &= p(z_{Stuhl,t}|X_t = L_l^{\beta_j}) \cdot p(z_{Tuere,t}|X_t = L_l^{\beta_j}) \\ p(z_{Stuhl,t}|X_t = L_l^{\beta_j}) &= \begin{cases} 1 & \text{falls } \frac{\partial p(z_t|X_t=\beta_j)}{\partial t} \neq 0 \\ \epsilon & \text{sonst} \end{cases} \\ p(z_{Tuere,t}|X_t = L_l^{\beta_j}) &= e_{L_l^{\beta_j}} \end{aligned}$$

Abschließend sollen noch einige implementierungstechnische Details erwähnt werden, die es zu berücksichtigen gilt, um ein gut funktionierendes System zu konstruieren.

- Auf Grund der Trägheit der Segmentierung und der Zeit, die vergeht bis ein Sensor-HMM mit allen Werten einer Aktivität ausgewertet wird, ist es notwendig, Totzeiten bei einem Zustandswechsel für die entsprechenden Sensoren einzuführen, da die Verfolgung sonst durch falsche Beobachtungen scheitert.
- Die Ausgaben der ersten HMM-Schicht werden geglättet, um Messausfälle, die aus mangelnder Bewegung der Benutzer resultieren, zu verhindern. Dies geschieht in der Form, dass für den Zeitpunkt t ein Ereignis als stattfindend betrachtet wird, wenn das entsprechende Wahrscheinlichkeitsverhältnis in einem Fenster, das die Länge der HMM-Eingabesequenz hat, den zugehörigen Schwellwert überschreitet.
- Verlässt ein Benutzer die Umgebung für mehr als 30 Sekunden oder misslingt die Verfolgung, so wird der Verfolgungsalgorithmus neu initialisiert, sobald der entsprechende Benutzer sich wieder an seinem Arbeitsplatz befindet.
- Der Raum mit der höchsten Aufenthaltswahrscheinlichkeit für einen Benutzer bezeichnet die Schätzung des Aufenthaltsortes.

5 Experimentelle Ergebnisse

Das vorgestellte System soll nun auf möglichst realistischen Daten evaluiert werden, um seine Leistungsfähigkeit zu testen. Abschnitt 5.1 erläutert hierbei zunächst die Randbedingungen, unter denen die Test- und Trainingsdaten gesammelt wurden. Der darauffolgende Abschnitt 5.2 beschreibt die durchgeführten Experimente zur Erkennung von Aktivitäten und geht dabei auf die Erkennungsleistung auf beiden HMM-Ebenen ein. Daran anschließend werden in Abschnitt 5.3 die erzielten Ergebnisse des Personenverfolgungsalgorithmus auf Zimmerniveau präsentiert.

5.1 Datensammlung und Annotation

Um oben beschriebenes System zu evaluieren, wurden vier Büros des ISL (Interactive Systems Lab) mit jeweils einer Kamera und einem Mikrofon ausgestattet. Zusätzlich wurde eine Kamera im Seminarraum (Lab) verwendet, um Ereignisse für die Personenverfolgung zu erkennen. Alle Videobilder hatten eine Größe von 640x480 Bildpunkten und wurde mit einer Rate von 7,5 Hz aufgenommen. Die Kameras waren statisch in den Ecken der Räume knapp unter der Decke angebracht, so dass ein möglichster großer Teil des Raumes überblickt werden konnte. Dies wurde außerdem durch die Verwendung von Weitwinkelobjektiven mit einem Blickwinkel von zirka 90° ermöglicht. Um einen besseren Eindruck zu geben, zeigt Abbildung 5.1 beispielhaft Bilder aus jedem der Räume. Das Audiosignal wurde mit einer Abtastrate von 16 kHz verarbeitet, wobei die Merkmale in Fenstern einer Länge von 20 ms bestimmt wurden. Die Überlappung zweier benachbarter Fenster betrug 10 ms. Es wurden omnidirektionale Mikrofone verwendet, um zu gewährleisten, dass Geräuschquellen aus allen Richtung wahrgenommen werden können. Das Signal musste vor der weiteren Verarbeitung vorverstärkt werden, wobei ein Tiefpassfilter mit $f_0 = 80$ Hz zum Einsatz kam, um Netzbrummen zu unterdrücken. Um die Privatsphäre aller Benutzer zu gewährleisten, wurde allerdings nicht das unverarbeitete Audiosignal gespeichert, sondern nur die Merkmale Signalenergie, Nulldurchgangsrate und Grundfrequenz, welche zur Laufzeit extrahiert wurden. Um die Synchronisation zwischen Audio- und Videomodalität sowie der Sensordaten zwischen den verschiedenen Räumen zu ermöglichen, wurden alle Eingabeströme mit Zeitstempeln versehen.
Insgesamt wurden an sechs Tagen jeweils zirka 7 Stunden Daten gesammelt, ohne

(a) Büro A

(b) Büro B

(c) Büro C

(d) Büro D

(e) Seminarraum (Lab)

Abbildung 5.1: *Beispielhafte Kamerabilder aus jedem der überwachten Räume*

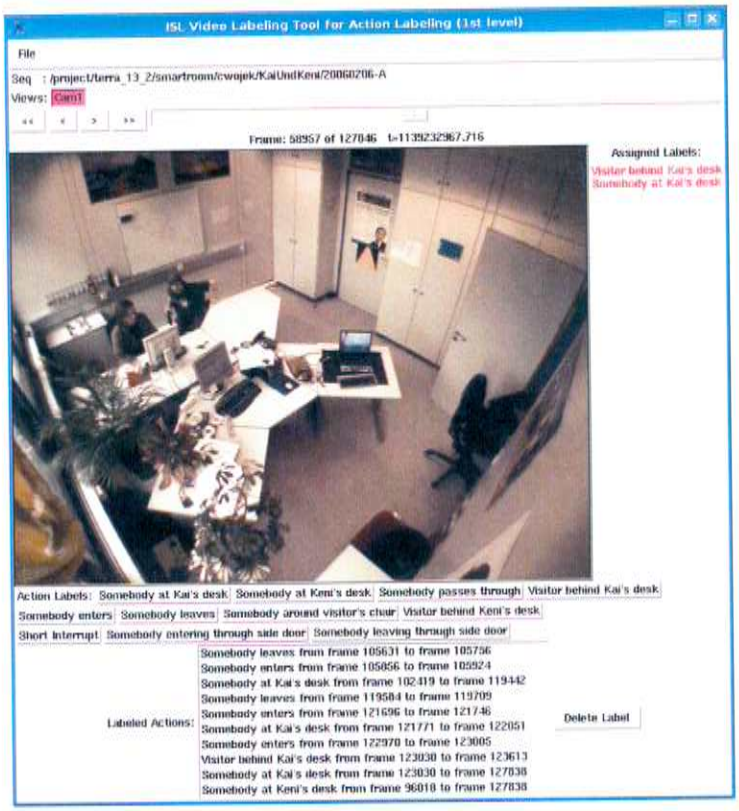

Abbildung 5.2: *Bildschirmfoto des Annotationswerkzeugs für die unterste Schicht von HMMs*

dabei expliziten Einfluss auf das Geschehen zu nehmen. Die Daten von vier Tagen wurden zum Training der Modelle verwendet; die restlichen für die experimentelle Evaluation der vorgestellten Aktivitätenerkennungsverfahren. Da dieser Datensatz wenige Ereignisse enthielt, bei dem die Benutzer die Zimmer wechselten, war er schlecht geeignet, um den Personenverfolgungsalgorithmus zu evaluieren. Außerdem enthielt er viele Ausschnitte, bei denen die Benutzer nicht vollständig durch die Türe gingen, so dass dies nicht als Ereignis detektiert werden konnte. Weiterhin stellte die Tatsache, dass Besucher meist nicht Platz nahmen, sondern an beliebigen Stellen im Raum standen, ein schwieriges Erkennungsproblem dar.
Deshalb wurde zusätzlich ein zweiter Datensatz mit vorgegebener Ereignisabfolge von der ungefähren Länge einer Stunde aufgenommen, der der alleinigen Evaluation des Verfolgungsproblems diente.
Um die tatsächlich stattfindenden Aktivitäten zu bestimmen, die als Referenzwerte für das Training und Testen des Algorithmus dienten, wurden mit Hilfe eines Werkzeugs, das in Abbildung 5.2 dargestellt ist, manuell Annotationen angebracht. Dies geschah jeweils separat für die beiden Schichten, also zunächst für die zu erkennenden Ereignisse der ersten HMM-Schicht und danach für die Raumsituation der zweiten Schicht. Dabei ist zu bemerken, dass Ereignisse auch zeitgleich stattfinden konnten und nur Aktivitäten, die wirklich charakteristisch

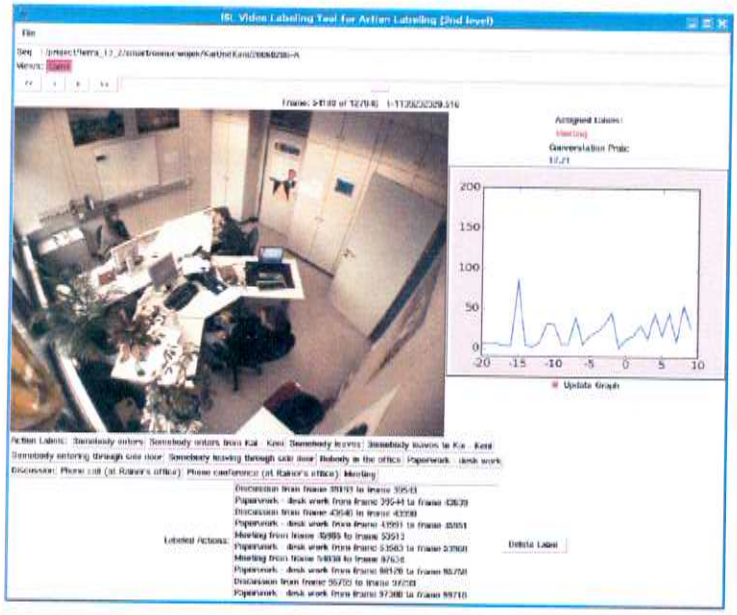

Abbildung 5.3: *Bildschirmfoto des Annotationswerkzeugs für die obere HMM-Ebene; mit Hilfe eines Diagramms wird die Wahrscheinlichkeit dafür angegeben, ob gesprochen wird.*

für eine Klasse sind, verwendet wurden. Für die Annotation der Daten mit den semantischen Situationen zum Training der zweiten HMM-Ebene ergab sich das Problem, dass die Information, ob gesprochen wurde oder nicht, durch die alleinige Speicherung von Merkmalen verloren ging. Dies erschwerte die Annotation insofern, als dass bestimmte Klassen nicht allein auf Basis des Videobildes voneinander unterschieden werden können. Um dem Abhilfe zu verschaffen, wurde speziell für die Annotation ein Paar von Audio-HMMs mit der Sequenzlänge einer Sekunde trainiert, das mit Hilfe der gespeicherten Merkmale rekonstruieren konnte, ob im Raum gesprochen wurde oder nicht (siehe Abbildung 5.3). Als Trainingsdaten wurden hier ebenfalls die 15-minütigen Datensätze der separat aufgenommenen Audiorohdaten verwendet.

5.2 Aktivitätenerkennung

Das System zur Aktivitätenerkennung soll in zwei Schritten evaluiert werden. Zunächst soll in Abschnitt 5.2.1 die Erkennungsleistung der HMM-Schicht, die die visuellen Informationen verarbeitet, gemessen werden. Auf die Evaluation der auf dem Audioeingabestrom arbeitenden HMMs muss leider verzichtet werden, da auf Grund der Tatsache, dass nur die Merkmale gespeichert werden konnten, keine Referenzklassen bekannt sind. Im anschließenden Abschnitt 5.2.2 werden die experimentellen Ergebnisse für die automatische Bestimmung von Situationen in einem Raum präsentiert.

5.2.1 Erste Inferenzschicht

Für die Messung der Erkennungsleistung musste zunächst empirisch ein Schwellwert m_i für $R^v_{i,t}$ für jedes zu erkennende Ereignis bestimmt werden. Bei Überschreitung dieses Schwellwerts wurde das Ereignis als erkannt gewertet. Zusätzlich wurden die Ausgabewahrscheinlichkeiten wie folgt geglättet:

$$R^v_{i,t}{}' = \max_{j=t}^{t+L_i} R^v_{i,j}$$

Hierbei bezeichnet L_i die Länge der Eingabesequenz für Ereignis i. Ein Ereignis wurde also als erkannt betrachtet, falls in einem Fenster der Länge L_i der Schwellwert m_i für $R^v_{i,t}{}'$ überschritten wurde. Da für alle Büroräume in etwa gleichwertige Ergebnisse erzielt wurden, soll auf eine lange Diskussion aller Räume an dieser Stelle verzichtet werden. Stattdessen sollen für die Büros B und D die Ergebnisse detailliert vorgestellt und für die restlichen Räume auf den Anhang B verwiesen werden. In Büro B, das in Abbildung 5.1(b) zu sehen ist, arbeiten normalerweise zwei Mitarbeiter an jeweils einer Seite eines großen Schreibtisches. An dessen Kopfende befindet sich außerdem ein Stuhl für einen Besucher. Da es sich bei den Besuchern oftmals um betreute Studenten handelt, nehmen diese nicht immer auf diesem Platz, sondern befinden sich direkt am Arbeitsplatz der Mitarbeiter, um an deren Computerbildschirm Ergebnisse zu betrachten oder um Sachverhalte zu diskutieren.
Büro D (Abbildung 5.1(d)) ist das Arbeitszimmer eines einzelnen Mitarbeiters. Der große Schreibtisch, an dem dieser normalerweise arbeitet, nimmt etwa ein Drittel des Raumes ein. Für Besucher existiert ein separater Tisch mit Sitzgelegenheiten am anderen Ende des Zimmers. Allerdings passiert es auch hier, dass diese der besseren Diskussionsmöglichkeit wegen direkt an den Schreibtisch kommen.
Die Länge der klassifizierten Eingabesequenzen auf dieser Ebene bewegt sich im Bereich von ungefähr 5 Sekunden für die Ereignisse, die an Türen stattfinden und 10 Sekunden für die restlichen Aktivitäten.
Die Erkennungsraten sowie der Anteil der falsch erkannten Ereignisse einer Klasse sind in den Tabellen 5.1 und 5.2 nebst dem Gesamtanteil an den Daten eines Büros dargestellt. Es ist ersichtlich, dass die Detektion von Benutzern, die an ihren Schreibtischen sitzen, sehr gut funktioniert. Die wenigen Fehler, die gemacht werden, resultieren aus der Trägheit des adaptiven Hintergrundmodells beim Wechsel zu einer anderen Aktivität. Die Nichterkennung von Ereignissen dieser Art resultiert zumeist aus einem Mangel an Bewegung. Gleiches gilt für die Detektion von Gästen auf den vorgesehenen Besucherstühlen.
Schwieriger ist es offensichtlich, zwei Benutzer, die räumlich nicht weit voneinander getrennt sind, an einem Schreibtisch zu erkennen. Der Hauptunterschied zur vorigen Klasse, in dem nur eine Person detektiert werden sollte, liegt für den gewählten Merkmalsraum in der Größe des Vordergrundbereichs und der Intensität der Bewegung. Der segmentierte Bereich ist erwartungsgemäß größer wenn

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Jemand sitzt am Schreibtisch von Benutzer 3	97,8 %	1,9 %	59,3 %
Besucher hinter Schreibtisch von Benutzer 3	78,1 %	14,8 %	1,5 %
Jemand sitzt am Schreibtisch von Benutzer 4	94,5 %	4,2 %	69,9 %
Besucher hinter Schreibtisch von Benutzer 4	64,8 %	22,2 %	2,5 %
Jemand sitzt auf dem Besucherstuhl	98,9 %	11,6 %	2,2 %
Jemand kommt herein (Haupttüre)	100,0 %	3,8 %	0,3 %
Jemand geht hinaus (Haupttüre)	98,3 %	5,2 %	0,3 %
Jemand kommt herein (Seitentüre)	92,8 %	2,9 %	0,2 %
Jemand geht hinaus (Seitentüre)	91,8 %	2,6 %	0,2 %

Tabelle 5.1: *Erkennungsergebnisse für Ereignisse der unteren HMM Schicht mit visuellen Merkmalen in Büro B*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Jemand sitzt am Schreibtisch von Benutzer 7	97,1 %	4,7 %	73,2 %
Besucher hinter Schreibtisch von Benutzer 7	69,3 %	25,5 %	7,6 %
Jemand sitzt am Besuchertisch (kurze Seite)	93,5 %	2,8 %	0,3 %
Jemand sitzt am Besuchertisch (lange Seite)	100,0 %	3,6 %	0,2 %
Jemand kommt herein	94,7 %	3,4 %	0,3 %
Jemand geht hinaus	95,4 %	2,3 %	0,4 %

Tabelle 5.2: *Erkennungsergebnisse für Ereignisse der unteren HMM Schicht mit visuellen Merkmalen in Büro D*

zwei Personen an einem Schreibtisch sitzen und außerdem entsteht in Gesprächen durch Gestikulieren mehr Bewegung. Folglich sind die falsch erkannten Ereignisse dieser Klassen darauf zurückzuführen, dass eine einzelne Person sich sehr stark bewegt hat. Nichterkennung hingegen resultiert meist aus relativ unbewegten Szenen, bei denen beide Benutzer den Computerbildschirm betrachten.
Die Erkennung von hereingehenden oder den Raum verlassenden Benutzern gelingt mit relativ hoher Zuverlässigkeit. Allerdings bleibt leider festzustellen, dass sehr oft die HMMs beider Ereignisse nahezu gleichzeitig mit hohen Beobachtungswahrscheinlichkeiten reagieren, wobei meist jedoch das richtige HMM zuerst ausschlägt. Dies lässt darauf schließen, dass in diesem eher einfachen Merkmalsraum die Bewegung der Türe selbst schlichtweg einen zu großen Teil des Ereignisses darstellt. Beide HMMs lernen somit eigentlich den Vorgang der sich öffnenden und schließenden Türe. Diese Hypothese wird durch die Beobachtung gestützt, dass eine Person, die die Türe zwar passiert, aber nicht den Raum verlässt, beide Ereignisse nicht auslöst.

5.2.2 Zweite Inferenzschicht

Anschließend an die Evaluation der ersten Schicht soll nun die Erkennung von Situationen auf der zweiten Ebene von HMMs experimentell überprüft werden. Auf dieser wurden im Gegensatz zur ersten Ebene die direkten HMM-Beobachtungswahrscheinlichkeiten zur Klassifikation genutzt. Dabei wird die Situation mit der höchsten Ausgabewahrscheinlichkeit als Klassifikationsergebnis gewählt. Voraussetzung hierfür ist die Verwendung gleicher Merkmalsvektoren für alle zu klassifizierenden Ereignisse sowie eine gleiche, feste Länge L_i der Eingabesequenzen. Als Länge eines Eingabefensters wurde 30 Sekunden bestimmt. Im Merkmalsvektor $R_{i,t}$ befinden sich mit Ausnahme der Wahrscheinlichkeiten, die mit Ereignissen der Türe verbunden sind, alle Wahrscheinlichkeitsverhältnisse $R^v_{i,t}$ und $R^a_{i,t}$ der unteren Inferenzschicht.
Um kurzfristige Schwankungen auszugleichen, wurden die Ausgabewahrscheinlichkeiten $P(R_{i,t}|S_i^2)$ auch auf dieser Ebene wieder geglättet. Dabei wird ein Fenster der Länge 150 Sekunden über das Wahrscheinlichkeitsprofil geschoben und dem Mittelpunkt jeweils die Situation zugewiesen, die in diesem Fenster am häufigsten vorkommt. Außerdem wurden Fehler, die am Anfang und Ende einer Situation entstanden, für die Länge L_i nicht gezählt, da diese aus der festen Eingabesequenzlänge für die Merkmalsvektoren resultieren. Diese ist für die Vergleichbarkeit der Inferenzergebnisse notwendig. Folglich ist der Merkmalsvektor zu Beginn und am Ende einer Situation mit Merkmalen zweier verschiedener Situationen gefüllt, so dass eine sinnvolle Klassifikation nur bedingt möglich sein kann.
In den Tabellen 5.3 und 5.4 sind die Ergebnisse der bereits zuvor diskutierten Büroräume abgebildet. Zusätzlich zu den Erkennungsraten wird außerdem die Vertauschungsmatrix zur besseren Analyse gezeigt. Für die Ergebnisse der restlichen Räume, die sich ähnlich erklären lassen, sei auf Anhang C verwiesen.
Für **Büro B** sollten die Klassen wie folgt definiert sein:

Niemand im Büro: Es befindet sich niemand im Büroraum.

Schreibtischarbeit: Einer oder beide Mitarbeiter arbeiten für sich in Stille an ihrem Schreibtisch.

Diskussion: Die beiden Mitarbeiter, die im Büro arbeiten, diskutieren einen Sachverhalt informell, das heißt sie sitzen weiterhin an ihrem Platz. Dabei befinden sich keine weiteren Personen im Raum.

Besprechung: Mindestens zwei Personen im Raum befinden sich in einer Besprechung, was bedeuten soll, dass sie sich in kurzer Distanz gegenüber sitzen oder gemeinsam eine Fläche wie die Tafel oder den Computerbildschirm betrachten. Bei den Personen kann es sich sowohl um Mitarbeiter, die in Büro B arbeiten, als auch um Besucher handeln.

Betrachtet man das Klassifikationsergebnis, so fällt auf, dass die beiden erstgenannten Klassen sehr zuverlässig erkannt werden können. Es kommt bei *Niemand*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Niemand im Büro	95,5%	0,5%	10,1%
Schreibtischarbeit	90,7%	5,8%	62,4%
Diskussion	73,9%	4,8%	18,4%
Besprechung	69,6%	2,8%	9,1%

Beschreibung		[1]	[2]	[3]	[4]
Niemand im Büro	[1]	**3462**	10	144	11
Schreibtischarbeit	[2]	695	**20341**	723	663
Diskussion	[3]	76	123	**4890**	1524
Besprechung	[4]	0	793	203	**2278**

Tabelle 5.3: *Erkennungsraten und Vertauschungsmatrix (in Sekunden) auf der zweiten HMM Schicht (Büro B)*

im Büro lediglich zu wenigen Verwechslungen, die vermutlich aus relativ lauten Diskussionen im Nachbarbüro resultieren. Für die Klasse *Diskussion* ist die relativ hohe Anzahl an Verwechslungen mit der Klasse *Besprechung* auffällig. Diese sind meist das Ergebnis von Fehlklassifikationen auf der ersten HMM-Schicht. Dabei wird bei einer *Diskussion* soviel Bewegung an einem der Mitarbeiterplätze registriert, dass das System in der ersten Inferenzschicht annimmt, es wäre ein Besucher mit anwesend, so dass es sich um eine *Besprechung* handelt. Allgemein bleibt hier festzustellen, dass die Trennung dieser beiden Klassen - auch für Menschen - sehr schwierig ist, zumal die Klassen fließend ineinander übergehen.
Schließlich lässt sich die hohe Anzahl an Vertauschungen der Klasse *Besprechung* mit *Schreibtischarbeit* wie folgt erklären. Eine der Testsequenzen enthielt einen Abschnitt, in der ein Student mit einem der Mitarbeiter an dessen Computer etwas analysierte. Allerdings kam es hierbei zu mehreren Denkpausen und zusätzlich verhielten sich beide eher untypisch ruhig, so dass das System statt zweier sich unterhaltender Personen von einer am Computer arbeitenden Person ausgehen konnte und somit zum falschen Schluss gelangte.
Ähnlich gestaltet sich die Erklärung der Ergebnisse für **Büro D**, in dem eine Person arbeitet, wobei die Klassen folgendermaßen festgelegt sein sollen:

Niemand im Büro: Es befindet sich niemand im Zimmer.

Schreibtischarbeit: Der Mitarbeiter arbeitet für sich in Stille an seinem Schreibtisch.

Telefonat: Der Mitarbeiter führt ein Telefonat, wobei sich keine weiteren Personen im Raum befinden.

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Niemand im Büro	97,7%	0,5%	22,0%
Schreibtischarbeit	86,2%	6,3%	45,8%
Telefonat	70,1%	5,6%	18,8%
Besprechung	60,0%	5,3%	13,4%

Beschreibung		[1]	[2]	[3]	[4]
Niemand im Büro	[1]	**6995**	17	117	32
Schreibtischarbeit	[2]	659	**12877**	352	1042
Telefonat	[3]	76	977	**4294**	776
Besprechung	[4]	26	1031	685	**2611**

Tabelle 5.4: *Erkennungsraten und Vertauschungsmatrix (in Sekunden) auf der zweiten HMM Schicht (Büro D)*

Besprechung: Mindestens zwei Personen befinden sich im Büro zu einer Besprechung. Dabei können diese entweder am separat bereit gestellten Besuchertisch sitzen oder sich vor dem Computer am Schreibtisch des Mitarbeiters aufhalten.

Auch für dieses Büro funktioniert die Erkennung der Klasse *Niemand im Büro* sehr zuverlässig. Für die Klasse *Schreibtischarbeit* kommt es auf Grund von mangelnder Bewegung während der Arbeit am Computer gelegentlich zu Problemen bei der Erkennung des Mitarbeiters an seinem Schreibtisch. Folglich wird dann beim zweiten Inferenzschritt davon ausgegangen, dass sich niemand im Büro befindet. Anderseits passiert es auch, dass der Mitarbeiter des öfteren für kurze Zeit aufsteht und sich am Schreibtisch bewegt. Dies führt dann auf Grund der starken Bewegung zur Fehlannahme, dass sich zwei Personen am Schreibtisch befinden würden und es sich um eine Besprechung handelt.
Solange der Benutzer spricht, werden Telefonate sehr zuverlässig erkannt. Zu Problemen kommt es erst, sobald er seinem Gesprächspartner über einen längeren Zeitraum zuhört, so dass das Telefonat kaum von Schreibtischarbeit unterschieden werden kann. Wie bereits für die Klasse Schreibtischarbeit diskutiert, passiert es auch während einiger Telefonate, dass der Benutzer aufsteht und entlang seines Schreibtisches läuft, was in diesem Fall auch wieder aus oben genannten Gründen zu einer Verwechslung des Telefonats mit einer Besprechung führt.
Die Fehlklassifikationen der Klasse Besprechung lassen sich im Wesentlichen wie für Büro B erklären. Zu wenig Bewegung oder lange Denkpausen führen zu einer fehlerhaften Erkennung auf der ersten HMM-Ebene, die eine Fehlklassifikation auf zweiter Ebene nach sich zieht. Auch hier verhält es sich wie für Büro B, dass

eine Besprechung gut erkannt wird, wenn der Besucher auf dem separat bereit gestellten Stuhl Platz nimmt.

5.3 Personenverfolgung auf Zimmerniveau

Zum Abschluss des Evaluationskapitels soll nun noch die Leistungsfähigkeit des Algorithmus zum Verfolgen von Personen auf Raumniveau analysiert werden. Wie bereits erwähnt, wurde hierfür ein zweiter Datensatz von der ungefähren Länge einer Stunde aufgenommen, da die Daten, die zur Aktivitätenklassifikation benutzt wurden, zu wenig Ereignisse enthielten, bei denen Benutzer den Raum wechselten. Dabei wurde bei der Aufnahme darauf geachtet, dass nicht zu viele Übergänge gleichzeitig stattfanden und diese ausreichenden zeitlichen Abstand voneinander besaßen. Leider ließ es sich hier nicht vermeiden, dass sich zwei Personen im Seminarraum aufhielten, die nicht verfolgt wurden und somit Störeinfluss auf das System hatten.
Auch für diesen Abschnitt sollen beispielhaft nur die Ergebnisse von zwei der Nutzer diskutiert werden. Die übrigen sind in Anhang D zu finden.
In Tabelle 5.5 ist das Ergebnis der Personenverfolgung für Benutzer 4 aufgelistet.

Referenztrajektorie			**Geschätzte Trajektorie**		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	31	Büro B	Büro B	0	41
45	67	Lab	Lab	49	75
76	799	Büro B	Büro B	81	809
809	1109	Büro D	Büro D	810	1102
1117	2194	Büro B	Büro B	1103	2197
2194	2484	Büro A	Büro A	2198	2496
2495	2660	Büro D	Büro D	2497	2671
2667	3248	Büro B	Büro B	2672	3263
3248	3388	Außer Sicht	Außer Sicht	3264	3382
3388	3685	Büro B	Büro B	3383	3687
3685	3698	Lab	Lab	3688	3705
3708	3719	Büro A	Büro D	3709	3925
3719	3925	Büro B			

Tabelle 5.5: *Geschätzte Trajektorie für Benutzer 4 (Zeiten sind in Sekunden angegeben)*

Dabei wird die tatsächliche Referenztrajektorie auf der linken Seite mit der vom Verfolgungssystem erstellten Schätzung verglichen. Die Verzögerung beim Wechsel von Räumen resultiert dabei aus der bereits angesprochenen Trägheit der Segmentierung. Des Weiteren sollte erwähnt werden, dass bei der Darstellung der Zustand *Außer Sicht* der Übersichtlichkeit wegen nicht mit aufgelistet wird, sofern er für

weniger als 30 Sekunden aktiv ist.
Es wird deutlich, dass die Verfolgung bis zum Zeitpunkt 3705 Sekunden sehr gut funktioniert und sogar vom System erkannt wird, dass der verfolgte Benutzer sich für einige Minuten aus dem Umfeld der Versuchsumgebung entfernt hat. Danach wird die Spur allerdings leider verfälscht, was damit zusammen hängt, dass Benutzer 7 ungefähr ab dem Zeitpunkt 3200 Sekunden seine Unterlagen ordnet und deshalb nicht mehr auf seinem Stuhl sitzt. Folglich erhält man für dieses Büro keine zuverlässigen Beobachtungen mehr. Dabei werden einige Ereignisse auf dem Besucherplatz und für das Passieren der Türe fälschlicherweise ausgelöst, was auch bei der Verfälschung der Spur von Benutzer 4 der Fall ist, da genau, als dieser auf dem Flur an der Türe vorbei läuft, ein solches Ereignis erzeugt wird. Auf Grund der Datenzuordnung mit dem *Nearest Neighbor Standard Filter* wird dieses dem verfolgten Benutzer fälschlicherweise zugewiesen und die Trajektorie auf Büro D abgelenkt.
Tabelle 5.6 zeigt die Trajektorie von Benutzer 5, bei dem die Verfolgung weniger

Referenztrajektorie			**Geschätzte Trajektorie**		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	1419	Büro C	Büro C	0	1414
1428	1599	Büro A	Büro A	1429	1599
1599	1610	Büro B	Büro B	1600	1683
1610	1743	Außer Sicht	Lab	1684	1725
1743	2152	Büro C	Büro D	1732	2229
2152	2322	Außer Sicht	Lab	2230	2233
			Büro B	2242	2246
2322	3929	Büro C	Außer Sicht	2304	2369
			Büro C	2370	3925

Tabelle 5.6: *Geschätzte Trajektorie für Benutzer 5 (Zeiten sind in Sekunden angegeben)*

gut funktioniert hat. Bis zum Zeitpunkt 1610 Sekunden funktioniert die Verfolgung sehr gut, dann allerdings verlässt der Benutzer die überwachte Umgebung in Richtung Küche. Dort kann er nicht mehr wahrgenommen werden und der Verfolgungsalgorithmus erhält keine Beobachtungen mehr, um eine Hypothese zu stützen. Gleichzeitig ist möglich, dass der Benutzer vom Flur aus potentiell in jedem Raum wieder auftauchen kann. Folglich wird mit dem gegebenen Algorithmus das erste Ereignis, bei dem eine Tür durchschritten wird, Benutzer 5 zugeordnet und somit seine Spur verloren. Erst als der Algorithmus zum Zeitpunkt 2370 Sekunden wieder an dessen Arbeitsplatz reinitialisiert wird, gelingt die Verfolgung wieder zuverlässig. Daraus wird ersichtlich, dass die Art der Datenassoziation noch verbessert werden kann, um dadurch Fehler dieser Art zu vermeiden.
Alles in allem funktioniert der Verfolgungsalgorithmus also gut, solange die Benutzer im Sichtfeld einer Kamera beobachtbar sind. Ist dies nicht der Fall, kann

Benutzer	*Genauigkeit*	*Übergänge insgesamt*	*Erkannte Übergänge*
Benutzer 1	94,5 %	4	4
Benutzer 2	93,8 %	6	5
Benutzer 3	97,6 %	6	6
Benutzer 4	91,5 %	10	8
Benutzer 5	82,3 %	6	3
Benutzer 6	89,5 %	5	3
Benutzer 7	91,0 %	7	7
Insgesamt	91,5 %	44	82,0 % ≙ 36

Tabelle 5.7: *Übersicht der Ergebnisse zur Personenverfolgung auf Raumniveau*

es zu Problemen kommen, sobald sich ein weiterer Benutzer auch außerhalb des Sichtfeldes auf dem Flur bewegt. Dadurch kann nur schwer entschieden werden, wem die Beobachtungen des nächsten Passierens einer angrenzenden Türe zuzuordnen sind.
Innerhalb der Räume funktioniert die Erkennung der Benutzer gut, solange sie sich in den vorher als signifikant erlernten Bereichen aufhalten, so dass die Verfolgung zuverlässig funktioniert. Kritisch sind jeweils die Übergänge in einen anderen Raum, insbesondere wenn das Sichtfeld aller Kameras für längere Zeit verlassen wird. Schließlich fasst Tabelle 5.7 noch die Ergebnisse aller verfolgten Personen zusammen. Dabei werden neben dem Anteil des richtig geschätzten Aufenthaltsortes auch die richtige Erkennung der Zustandsübergänge beziehungsweise der Raumwechsel angegeben. Diese sind für das gegebene Szenario wesentlich aussagekräftiger, da die Benutzer meist eher statisch an ihrem Arbeitsplatz sitzen und die Verfolgung damit hauptsächlich durch das Wechseln des Raumes interessant wird. Es ist dabei sowohl die Gesamtzahl der Raumwechsel als auch die Zahl der davon richtig erkannten angegeben. Durchschnittlich wird der Aufenthaltsort in 91,5 % der Zeit richtig geschätzt, wobei 82,0 % der Übergänge richtig erkannt werden.
Abschließend bleibt zu bemerken, dass für die Personenverfolgung nur die Ereignisse aus der Aktivitätenerkennung, sowie das a-priori Wissen über den Aufenthaltsort verwendet wurden und keinerlei personenspezifische Merkmale herangezogen wurden. In Zukunft sollten deshalb noch weitere Merkmale wie Farbe, Personengröße, sowie Ergebnisse eines Gesichtserkenners oder einer Sprecheridentifikation integriert werden, um weitere Verbesserungen zu erzielen.

6 Zusammenfassung und Ausblick

Die Erkennung von Aktivitäten stellt eine der wichtigsten Grundlagen dar, um in einer intelligenten Umgebung weitere Dienste zur Verfügung stellen zu können und um auf Situationen angemessen reagieren zu können. Für eine sinnvolle Auswahl von Reaktionen ist es außerdem oftmals notwendig, den Aufenthaltsort sämtlicher Benutzer zu kennen. Damit lässt sich zum Beispiel ein System realisieren, das in der Lage ist, auf externe Anrufe entsprechend der Verfügbarkeit und dem Aufenthaltsort von Personen dynamisch zu reagieren. Einerseits kann so abgeschätzt werden, wann eine nicht erreichbare Person wieder verfügbar ist und anderseits kann zum Beispiel eine kurze Nachricht an den zu erreichenden Benutzer weitergeleitet werden, ohne diesen in der gegenwärtigen Situation zu stören.
Die vorliegende Arbeit widmete sich sowohl der Erkennung von Aktivitäten in einer Büroumgebung mit mehreren Benutzern, als auch der Lokalisierung der Benutzer. Dabei stellte sich das besondere Problem, dass ein Teil der Umgebung nicht beobachtbar war.
Bei den in vier Büroräumen erkannten Klassen handelte es sich um NIEMAND IM BÜRO, SCHREIBTISCHARBEIT, DISKUSSION, TELEFONAT und BESPRECHUNG. Zusätzlich werden die sieben Benutzer noch in einem der Büros oder dem Seminarraum auf Zimmerniveau lokalisiert. Die Evaluation des vorgestellten Systems geschah dabei auf einem Datensatz, der unter realistischen Bedingungen mit einer sehr einfachen Sensorausstattung gewonnen wurde. Es wurde pro Raum jeweils nur eine Kamera und ein omni-direktionales Mikrofon verwendet.
Die Erkennung von Aktivitäten baut dabei auf einer Hierarchie von Hidden Markov Models auf, die auf oberster Ebene die zu erkennenden Situationen als Klassifikationsziel haben. Die Erkennung funktioniert abhängig von der Klasse recht gut, wobei mit zunehmender Anzahl von beteiligten Personen die Schwierigkeit steigt und die Erkennungsleistung zurückgeht. Die Erkennungsraten betragen dabei im Mittel zirka 80 % was auf Grund der einfachen Merkmale beachtlich ist und durch den Einsatz lokaler Merkmalsmodelle ermöglicht wird.
Die Verfolgung von Personen über mehrere Räume hinweg basiert auf einem Bayes'schen Filter und verwendet die geglätteten Inferenzergebnisse der untersten Schicht der HMM-Architektur. Das Verfolgungsproblem wird trotz der Verwendung einfacher Merkmale sehr zufriedenstellend gelöst, wobei zirka 80 % der Zustandsübergänge richtig erkannt werden und in etwa 90 % der Zeit der Aufenthaltsort eines Benutzers korrekt bestimmt wird. Probleme treten meist dann auf, wenn die zu verfolgende Person gleichzeitig mit anderen das Zimmer wechselt, da dabei die richtige Zuordnung von Beobachtungen zu verfolgten Personen

problematisch wird. Eine Fehlentscheidung hierbei kann schwerwiegende Folgen nach sich ziehen, weil Daten aus zwei unterschiedlichen stochastischen Prozessen vermischt werden. Die Schwierigkeit ergibt sich vorwiegend dann, wenn mehrere Personen sich außerhalb des beobachtbaren Bereichs auf dem Flur aufhalten und sich die Trajektorien dadurch kreuzen.
Um das Erkennungs- und Verfolgungssystem weiter zu verbessern, sollten folgende Punkte zur Verbesserung der Aktivitätenerkennung in Erwägung gezogen werden:

Trainingsschema für HMM: Ein Ansatz um das Training von HMMs einfacher und effizienter zu gestalten, ist die Bestimmung der Modellparameter durch Entropieminimierung wie von Brand und Kettnaker [BK00] vorgestellt. Dieser ist verwandt mit dem *Minimum Description Length* Prinzip und ersetzt den *Maximization*-Schritt des bereits vorgestellten Baum-Welch Algorithmus. Die Vorteile dieser Trainingsmethode sind, dass die Zahl der Zustände und der verwendeten Gaußverteilungen automatisch bestimmt werden kann und außerdem weniger Trainingsdaten benötigt werden. Folglich wäre damit auch das Problem gelöst, dass Overfitting auftreten kann, bei dem das Hidden Markov Model nicht mehr generalisiert. Zudem findet der modifizierte Algorithmus besser die entscheidenden Merkmale heraus und liefert interpretierbare Zustände, so dass der Vorverabeitungsschritt zur Auswahl von Merkmalen für jede Aktivität eventuell entfallen könnte.

Merkmalsauswahl: Des Weiteren wurde bei der experimentellen Evaluation ebenfalls deutlich, dass es schwierig ist, die Anzahl von Personen in einem Raum nur basierend auf Bewegung zu bestimmen. Da die Ausnutzung von Farbinformationen auf Grund der stark variierenden Beleuchtung nicht möglich ist, könnte der Einsatz von Gesichtsdetektoren Abhilfe schaffen. Das wohl bekannteste Vorgehen basiert auf der Arbeit von Viola und Jones [VJ01], bei der dank Boosting die Kombination mehrerer einfacher Merkmale ausreicht, um Gesichter zu erkennen. Grundlage hierfür wäre allerdings die passende und zeitaufwändige Auswahl von segmentierten Kopfbildern, so dass die Trainingsdaten für die Haar Kaskaden, Kopfansichten aus allen Richtungen enthalten, so wie diese aus den Kameraperspektiven auch zu erwarten sind. Außerdem könnte die genaue Position von Personen in Bildpunkten mit Hilfe von Personenverfolgungsalgorithmen bestimmt werden und ebenfalls ein wichtiges Merkmal darstellen.

Erweiterungen: Darüber hinaus ist eine Erweiterung der Menge der zu erkennenden Klassen denkbar. Beispielsweise könnten Klassen wie DISKUSSION AN DER TAFEL oder TELEFONKONFERENZ mit dem vorliegenden System vermutlich erkannt werden. Auf Grund mangelnder Trainingsdaten für diese Klassen konnte dies bisher jedoch nicht verwirklicht werden.
Des Weiteren könnte die Weitergabe der Inferenzwahrscheinlichkeiten vom ersten in die zweite Schicht nach dem Prinzip der harten Entscheidung näher

untersucht werden. Außerdem ist der Ausbau des Systems zu einer Echtzeitanwendung möglich, da die Dauer der Berechnungen, also der Merkmalsextraktion und Inferenz, weit hinter der tatsächlichen Länge der Sequenzen zurückbleibt. Schließlich sollte untersucht werden, ob die Einbeziehung des optischen Flusses beim Clustern des Vordergrundes zum lokalen Merkmalsmodell weitere Verbesserungen bringt.
Da weiterhin zu beobachten ist, dass bei der Erkennung von Personen, die das Zimmer betreten oder verlassen, oftmals zuerst die richtige Klasse erkannt wird, danach aber zusätzlich die falsche, könnte versucht werden, dies auszunutzen, um diese beiden Ereignisse auf der höheren Ebene besser zu unterscheiden und damit den Anteil der Fehlklassifikationen zu reduzieren.

Andere Lernverfahren auf höheren Inferenzebenen: Schließlich sei darauf verwiesen, dass nicht nur Hidden Markov Models zur Klassifikation der stattfindenden Situationen in Betracht kommen. Unter anderem gibt es einen Ansatz von Shi und Bobick [SB03, SHM+04], der versucht, mit einem stochastischen Zustandsautomaten – so genannten *P-Nets* – und der Modellierung der Aufenthaltsdauer in Zuständen, länger währende Aktivitäten, wie zum Beispiel Besprechungen, zu erkennen.
Darüber hinaus wäre auf höheren Inferenzebenen auch der Einsatz anderer Klassifikatoren, wie Entscheidungsbäumen anstatt von Hidden Markov Models, denkbar. Allerdings ist für die Erkennung der Elementarereignisse auf der untersten Ebene nicht zu erwarten, dass die Erkennungsleistung von HMMs um Größenordnungen übertroffen werden kann.

Bezogen auf das Personenverfolgungsproblem ergeben sich für zukünftige Arbeiten folgende Verbesserungsansätze:

Beobachtungsmodell: Zunächst einmal kann für die Zuordnung der Daten zu den verfolgten Personen jeweils ein Farbmodell angelernt werden. Basierend auf Farbmodellen für Kopf, Oberkörper und Beine könnten Beobachtungen den einzelnen verfolgten Personen besser zugeordnet werden. Für eine sinnvolle Anwendung ist allerdings auf Grund der wechselnden Lichtverhältnisse und der unterschiedlichen Kameratypen eine Farbkalibrierung aller verwendeten Kameras notwendig. Hierzu geben Renno *et al.* [RMEJ05] einen Überblick über bestehende Verfahren.

Erweiterungen: Außerdem ist die manuelle Bestimmung der Schwellwerte, um auf Basis der Inferenzergebnisse der ersten Ebene binäre Beobachtungen zu generieren, sehr zeitaufwendig. Eine automatische Bestimmung der Schwellwerte mit Normalverteilungen auf der Basis annotierter Daten würde dies überflüssig machen.

Datenzuordnung: Abschließend besteht auch bei der Zuordnung von Beobachtungsdaten zu Personen die Möglichkeit zu Verbesserungen. So schlagen Wilson und Atkeson [WA05] vor, den Aufenthaltsort und die Zuordnung von

Daten in einem gemeinsamen Zustandsraum zu betrachten. Dabei wird die Zuweisung der Daten mit $\theta_t(i, j)$ bezeichnet und nimmt nur den Wert 1 an, falls das Ereignis i der Person j zugeordnet wird und 0 sonst. Die Verwendung eines Rao-Blackwellised Partikelfilters [DdFMR00] erlaubt dann die Aufteilung des Beobachtungsmodells in einen analytisch zu lösenden Teil, der den Zustand gegeben der Beobachtungen und deren Zuordnung aktualisiert und einen mit Partikelfilter genähertem Anteil, der die Zuordnung der Daten zu Personen bewerkstelligt. Für den *Resampling*-Schritt des Partikelfilters muss allerdings eine passende Heuristik gewählt werden, um diesen effektiv einsetzen zu können. Diese nutzt dabei den vorher geschätzten Aufenthaltsort aller Personen und die geschätzte Zuordnung des letzten Zeitschritts.

Abschließend bleibt zu bemerken, dass sich mit der Überwachung von Räumen mittels audio-visueller Methoden auch soziale und rechtliche Fragestellungen ergeben, die nicht Gegenstand dieser Arbeit waren, aber dennoch für zukünftige Anwendungen zum Beispiel im Rahmen einer Benutzerstudie untersucht werden sollten.

A Vollständige Liste aller Aktivitäten

Dieser Abschnitt listet alle zu erkennenden Aktivitäten auf, um einen vollständigen Überblick zu gewähren.

Untere Ebene (Video)
Jemand steht am Drucker
Jemand kommt herein
Jemand geht hinaus

Tabelle A.1: *Klassifikationsziele für Lab für die untere Schicht*

Untere Ebene (Video)
Jemand sitzt am Schreibtisch von Benutzer 1
Jemand sitzt am Schreibtisch von Benutzer 2
Jemand sitzt am Besuchertisch
Jemand steht in der Seitentüre
Jemand kommt herein (Haupttüre)
Jemand geht hinaus (Haupttüre)
Jemand kommt herein (Seitentüre)
Jemand geht hinaus (Seitentüre)

Obere Ebene
Niemand im Büro
Schreibtischarbeit
Diskussion

Untere Ebene (Audio)
Gespräch findet statt
Hintergrundgeräusche

Tabelle A.2: *Klassifikationsziele für Büro A getrennt nach unterer Schicht (linke Spalte) und oberer Schicht (rechte Spalte)*

Untere Ebene (Video)
Jemand sitzt am Schreibtisch von Benutzer 3
Jemand sitzt am Schreibtisch von Benutzer 4
Besucher hinter Schreibtisch von Benutzer 3
Besucher hinter Schreibtisch von Benutzer 4
Jemand sitzt auf dem Besucherstuhl
Jemand kommt herein (Haupttüre)
Jemand geht hinaus (Haupttüre)
Jemand kommt herein (Seitentüre)
Jemand geht hinaus (Seitentüre)

Obere Ebene
Niemand im Büro
Schreibtischarbeit
Diskussion
Besprechung

Untere Ebene (Audio)
Gespräch findet statt
Hintergrundgeräusche

Tabelle A.3: *Klassifikationsziele für Büro B getrennt nach unterer Schicht (linke Spalte) und oberer Schicht (rechte Spalte)*

Untere Ebene (Video)
Jemand sitzt am Schreibtisch von Benutzer 5
Jemand sitzt am Schreibtisch von Benutzer 6
Jemand kommt herein
Jemand geht hinaus

Obere Ebene
Niemand im Büro
Schreibtischarbeit
Diskussion

Untere Ebene (Audio)
Gespräch findet statt
Hintergrundgeräusche

Tabelle A.4: *Klassifikationsziele für Büro C getrennt nach unterer Schicht (linke Spalte) und oberer Schicht (rechte Spalte)*

Untere Ebene (Video)
Jemand sitzt am Schreibtisch von Benutzer 7
Besucher hinter Schreibtisch von Benutzer 7
Jemand sitzt am Besuchertisch (kurze Seite)
Jemand sitzt am Besuchertisch (lange Seite)
Jemand kommt herein
Jemand geht hinaus

Obere Ebene
Niemand im Büro
Schreibtischarbeit
Telefonat
Besprechung

Untere Ebene (Audio)
Gespräch findet statt
Hintergrundgeräusche

Tabelle A.5: *Klassifikationsziele für Büro D getrennt nach unterer Schicht (linke Spalte) und oberer Schicht (rechte Spalte)*

B Visuelle Erkennungsleistung auf der ersten Inferenzschicht

In diesem Abschnitt werden sämtliche Ergebnisse bezüglich der Erkennungsleistung des implementierten Systems mit der ersten Schicht von HMM-Modellen dargestellt. Die Diskussion der Ergebnisse findet sich in Abschnitt 5.2.1.

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Jemand sitzt am Schreibtisch von Benutzer 2	97,2 %	4,7 %	71,2 %
Jemand sitzt am Schreibtisch von Benutzer 1	92,0 %	15,3 %	35,6 %
Jemand sitzt am Besuchertisch	1,9 %	1,5 %	3,6 %
Jemand steht in der Seitentüre	100,0 %	16,7 %	2,6 %
Jemand kommt herein (Haupttüre)	99,0 %	1,6 %	0,2 %
Jemand geht hinaus (Haupttüre)	93,1 %	1,2 %	0,2 %
Jemand kommt herein (Seitentüre)	98,9 %	3,8 %	0,2 %
Jemand geht hinaus (Seitentüre)	100,0 %	5,0 %	0,2 %

Tabelle B.1: *Erkennungsergebnisse für Ereignisse der unteren HMM Schicht mit visuellen Merkmalen in Büro A*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Jemand sitzt am Schreibtisch von Benutzer 3	97,8 %	1,9 %	59,3 %
Besucher hinter Schreibtisch von Benutzer 3	78,1 %	14,8 %	1,5 %
Jemand sitzt am Schreibtisch von Benutzer 4	94,5 %	4,2 %	69,9 %
Besucher hinter Schreibtisch von Benutzer 4	64,8 %	22,2 %	2,5 %
Jemand sitzt auf dem Besucherstuhl	98,9 %	11,6 %	2,2 %
Jemand kommt herein (Haupttüre)	100,0 %	3,8 %	0,3 %
Jemand geht hinaus (Haupttüre)	98,3 %	5,2 %	0,3 %
Jemand kommt herein (Seitentüre)	92,8 %	2,9 %	0,2 %
Jemand geht hinaus (Seitentüre)	91,8 %	2,6 %	0,2 %

Tabelle B.2: *Erkennungsergebnisse für Ereignisse der unteren HMM Schicht mit visuellen Merkmalen in Büro B*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Jemand sitzt am Schreibtisch von Benutzer 6	97,1 %	7,1 %	42,6 %
Jemand sitzt am Schreibtisch von Benutzer 5	96,2 %	4,6 %	40,1 %
Jemand kommt herein	92,2 %	5,3 %	0,4 %
Jemand geht hinaus	76,6 %	5,3 %	0,3 %

Tabelle B.3: *Erkennungsergebnisse für Ereignisse der unteren HMM Schicht mit visuellen Merkmalen in Büro C*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Jemand sitzt am Schreibtisch von Benutzer 7	97,1 %	4,7 %	73,2 %
Besucher hinter Schreibtisch von Benutzer 7	69,3 %	25,5 %	7,6 %
Jemand sitzt am Besuchertisch (kurze Seite)	93,5 %	2,8 %	0,3 %
Jemand sitzt am Besuchertisch (lange Seite)	100,0 %	3,6 %	0,2 %
Jemand kommt herein	94,7 %	3,4 %	0,3 %
Jemand geht hinaus	95,4 %	2,3 %	0,4 %

Tabelle B.4: *Erkennungsergebnisse für Ereignisse der unteren HMM Schicht mit visuellen Merkmalen in Büro D*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Jemand steht am Drucker	89,2 %	7,5 %	1,1 %
Jemand kommt herein	96,6 %	6,7 %	0,5 %
Jemand geht hinaus	96,9 %	8,7 %	1,0 %

Tabelle B.5: *Erkennungsergebnisse für Ereignisse der unteren HMM Schicht mit visuellen Merkmalen im Seminarraum (Lab)*

C Visuelle Erkennungsleistung auf der zweiten Inferenzschicht

In diesem Abschnitt werden sämtliche Ergebnisse bezüglich der Erkennungsleistung des implementierten Systems mit der zweiten Schicht von HMM-Modellen dargestellt. Die Diskussion der Ergebnisse findet sich in Abschnitt 5.2.2.

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Niemand im Büro	98,4 %	0,4 %	27,5 %
Schreibtischarbeit	94,6 %	2,9 %	53,9 %
Diskussion	78,7 %	4,0 %	18,6 %

Beschreibung		[1]	[2]	[3]
Niemand im Büro	[1]	**11472**	190	0
Schreibtischarbeit	[2]	826	**21631**	410
Diskussion	[3]	626	1056	**6202**

Tabelle C.1: *Erkennungsraten und Vertauschungsmatrix (in Sekunden) auf der zweiten HMM Schicht (Büro A)*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Niemand im Büro	95,5 %	0,5 %	10,1 %
Schreibtischarbeit	90,7 %	5,8 %	62,4 %
Diskussion	73,9 %	4,8 %	18,4 %
Besprechung	69,6 %	2,8 %	9,1 %

Beschreibung		[1]	[2]	[3]	[4]
Niemand im Büro	[1]	**3462**	10	144	11
Schreibtischarbeit	[2]	695	**20341**	723	663
Diskussion	[3]	76	123	**4890**	1524
Besprechung	[4]	0	793	203	**2278**

Tabelle C.2: *Erkennungsraten und Vertauschungsmatrix (in Sekunden) auf der zweiten HMM Schicht (Büro B)*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Niemand im Büro	66,3 %	2,4 %	7,0 %
Schreibtischarbeit	89,4 %	8,4 %	79,7 %
Diskussion	96,1 %	0,5 %	13,3 %

Beschreibung		[1]	[2]	[3]
Niemand im Büro	[1]	**1110**	371	192
Schreibtischarbeit	[2]	186	**16988**	1822
Diskussion	[3]	5	120	**3052**

Tabelle C.3: *Erkennungsraten und Vertauschungsmatrix (in Sekunden) auf der zweiten HMM Schicht (Büro C)*

Beschreibung	*Erkennungsrate*	*Falsch-Positiv-Rate*	*Datenanteil*
Niemand im Büro	97,7 %	0,5 %	22,0 %
Schreibtischarbeit	86,2 %	6,3 %	45,8 %
Telefonat	70,1 %	5,6 %	18,8 %
Besprechung	60,0 %	5,3 %	13,4 %

Beschreibung		[1]	[2]	[3]	[4]
Niemand im Büro	[1]	**6995**	17	117	32
Schreibtischarbeit	[2]	659	**12877**	352	1042
Telefonat	[3]	76	977	**4294**	776
Besprechung	[4]	26	1031	685	**2611**

Tabelle C.4: *Erkennungsraten und Vertauschungsmatrix (in Sekunden) auf der zweiten HMM Schicht (Büro D)*

D Ergebnisse zur Personenverfolgung auf Raumniveau

In diesem Abschnitt werden die geschätzten Trajektorien aller Benutzer zusammen mit den tatsächlichen Referenztrajektorien auf der Testsequenz zur Personenverfolgung dargestellt. Die Diskussion der Ergebnisse findet sich in Abschnitt 5.3.

Referenztrajektorie			**Geschätzte Trajektorie**		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
			Büro A	0	104
0	1089	Büro A	Büro B	105	237
			Büro A	238	1087
1089	1236	Büro B	Büro B	1088	1237
1236	3029	Büro A	Büro A	1238	3040
3047	3105	Lab	Lab	3051	3110
3118	3927	Büro A	Außer Sicht	3111	3142
			Büro A	3143	3925

Tabelle D.1: *Geschätzte Trajektorie für Benutzer 1 (Zeiten sind in Sekunden angegeben)*

Referenztrajektorie			Geschätzte Trajektorie		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	105	Büro A	Büro A	0	104
105	234	Büro B	Büro B	105	237
234	507	Büro A	Büro A	238	516
520	539	Lab	Lab	525	546
548	3409	Büro A	Büro A	550	3408
3409	3524	Außer Sicht	Lab	3409	3415
			Büro D	3416	3525
			Büro A	3526	3559
			Büro C	3565	3574
3524	3927	Büro A	Büro D	3575	3634
			Außer Sicht	3635	3666
			Büro A	3667	3925

Tabelle D.2: *Geschätzte Trajektorie für Benutzer 2 (Zeiten sind in Sekunden angegeben)*

Referenztrajektorie			Geschätzte Trajektorie		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	663	Büro B	Büro B	0	666
663	1048	Büro A	Büro A	667	1052
1048	1670	Büro B	Büro B	1053	1683
1681	1720	Lab	Lab	1684	1726
1728	2229	Büro D	Büro D	1732	2229
2237	2778	Büro B	Lab	2230	2234
			Büro B	2242	2791
2790	2822	Lab	Lab	2792	2829
2832	3925	Büro B	Büro B	2837	3925

Tabelle D.3: *Geschätzte Trajektorie für Benutzer 3 (Zeiten sind in Sekunden angegeben)*

Referenztrajektorie			Geschätzte Trajektorie		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	31	Büro B	Büro B	0	41
45	67	Lab	Lab	49	75
76	799	Büro B	Büro B	81	809
809	1109	Büro D	Büro D	810	1102
1117	2194	Büro B	Büro B	1103	2197
2194	2484	Büro A	Büro A	2198	2496
2495	2660	Büro D	Büro D	2497	2671
2667	3248	Büro B	Büro B	2672	3263
3248	3388	Außer Sicht	Außer Sicht	3264	3382
3388	3685	Büro B	Büro B	3383	3687
3685	3698	Lab	Lab	3688	3705
3708	3719	Büro A	Büro D	3709	3925
3719	3925	Büro B			

Tabelle D.4: *Geschätzte Trajektorie für Benutzer 4 (Zeiten sind in Sekunden angegeben)*

Referenztrajektorie			Geschätzte Trajektorie		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	1419	Büro C	Büro C	0	1414
1428	1599	Büro A	Büro A	1429	1599
1599	1610	Büro B	Büro B	1600	1683
1610	1743	Außer Sicht	Lab	1684	1725
1743	2152	Büro C	Büro D	1732	2229
2152	2322	Außer Sicht	Lab	2230	2233
			Büro B	2242	2246
2322	3929	Büro C	Außer Sicht	2304	2369
			Büro C	2370	3925

Tabelle D.5: *Geschätzte Trajektorie für Benutzer 5 (Zeiten sind in Sekunden angegeben)*

Referenztrajektorie			Geschätzte Trajektorie		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	422	Büro C			
426	687	Büro D	Büro C	0	2925
692	2917	Büro C			
2922	3426	Büro B	Büro B	2926	3403
			Büro A	3404	3408
3426	3557	Büro A	Lab	3409	3412
			Büro D	3416	3525
			Büro A	3526	3564
3563	3929	Büro C	Büro C	3565	3925

Tabelle D.6: *Geschätzte Trajektorie für Benutzer 6 (Zeiten sind in Sekunden angegeben)*

Referenztrajektorie			Geschätzte Trajektorie		
Beginn	*Ende*	*Raum*	*Raum*	*Beginn*	*Ende*
0	189	Büro D	Büro D	0	196
200	317	Büro A	Büro A	203	326
326	1228	Büro D	Büro D	327	1235
1228	1341	Außer Sicht	Außer Sicht	1236	1342
1341	3147	Büro D	Büro D	1343	3154
3154	3196	Lab	Lab	3155	3198
			Büro D	3199	3525
3197	3767	Büro D	Außer Sicht	3579	3618
			Büro D	3619	3711
			Büro A	3712	3720
3776	3790	Büro B	Büro B	3721	3925

Tabelle D.7: *Geschätzte Trajektorie für Benutzer 7 (Zeiten sind in Sekunden angegeben)*

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[BAWPR02] Ben-Arie, Jezekiel, Zhiqian Wang, Purvin Pandit und Shyamsundar Rajaram: *Human Activity Recognition Using Multidimensional Indexing.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(8):1091–1104, August 2002.

[BD01] Bobick, Aaron und James Davis: *The Recognition of Human Movement Using Temporal Templates.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(3):257–267, März 2001.

[Bil97] Bilmes, Jeff: *A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.* Technischer Bericht, University of Berkeley, 1997.

[BK00] Brand, Matthew und Vera Kettnaker: *Discovery and Segmentation of Activities in Video.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):844–851, 2000.

[BME05] Black, James, Dimitrios Makris und Tim Ellis: *Validation of blind region learning and tracking.* In: *International Workshop on Performance Evaluation of Tracking and Surveillance*, Seiten 9–16, 2005.

[CBM02] Collobert, Ronan, Samy Bengio und Johnny Mariéthoz: *Torch: a modular machine learning software library.* Technischer Bericht, IDIAP, 2002.

[DBPV05] Duong, Thi, Hung Bui, Dinh Phung und Svetha Venkatesh: *Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model.* In: *IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 838–845, 2005.

[DdFMR00] Doucet, Arnaud, Narido de Freitas, Kevin Murphy und Stuart Russell: *Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks.* In: *Conference on Uncertainty in Artificial Intelligence*, Seiten 176–183, 2000.

[DdG01] Doucet, Arnaud, Nando de Freitas und Neil Gordon: *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, New York, 2001.

[DFB+05] Danninger, Maria, Gopi Flaherty, Keni Bernardin, Hazim Kemal Ekenel, Tobias Kluge, Robert Malkin, Rainer Stiefelhagen und Alexander Waibel: *The connector: facilitating context-aware communication.* In: *International Conference on Multimodal Interfaces*, Seiten 69–75, 2005.

[dK02] de Cheveigné, Alain und Hideki Kawahara: *YIN, a fundamental frequency estimator for speech and music.* Journal of the Acoustical Society of America, 111:1917–1930, 2002.

[DKR+06] Danninger, Maria, Tobias Kluge, Erica Robles, Leila Takayama, QianYing Wang, Rainer Stiefelhagen, Cliff Nass und Alexander Waibel: *The Connector Service - Predicting Availability in Mobile Contexts.* In: *Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Mai 2006.

[DLR77] Dempster, Arthur, Nan Laird und Donald Rubin: *Maximum-Likelihood from incomplete data via EM algorithm.* Journal Royal Statistical Society, Series B, 39:1–38, 1977.

[DR04] Dielmann, Alfred und Steve Renals: *Dynamic Bayesian Networks for Meeting Structuring.* In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seiten 76–86, 2004.

[DTK+02] Demirdjian, David, Konrad Tollmar, Kimberle Koile, Neal Checka und Trevor Darrell: *Activity maps for location-aware computing.* In: *IEEE Workshop on Applications of Computer Vision*, Seiten 70–75, 2002.

[FHL+03] Fox, Dieter, Jeffrey Hightower, Lin Liao, Dirk Schulz und Gaetano Borriello: *Bayesian Filtering for Location Estimation.* IEEE Pervasive Computing, 2(3):24–33, Juli–September 2003.

[GX03] Gong, Shaogang und Tao Xiang: *Recognition of group activities using dynamic probabilistic networks.* In: *International Conference on Computer Vision*, Seiten 742–749, 2003.

[HBN00] Hongeng, Somboon, Francois Brémond und Ramakant Nevatia: *Bayesian Framework for Video Surveillance Application.* In: *International Conference on Pattern Recognition*, Band I, Seiten 164–170, 2000.

[HHE03] Hamid, Raffay, Yan Huang und Irfan Essa: *ARGMode - Activity Recognition using Graphical Models.* In: *IEEE Workshop on Detection and Recognition of Events in Video*, Band 4, Seiten 38–44, Juni 2003.

[IB98] Isard, Michael und Andrew Blake: *Condensation – conditional density propagation for visual tracking.* International Journal of Computer Vision, 29(1):5–28, 1998.

[IB00] Ivanov, Yuri und Aaron Bobick: *Recognition of Visual Activities and Interactions by Stochastic Parsing.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):852–872, August 2000.

[Int] Intel Cooperation: *Open Source Computer Vision Library (OpenCV).* Erhältlich unter: `http://www.intel.com/research/mrl/research/opencv/`.

[JH96] Johnson, Neil und David Hogg: *Learning the Distribution of Object Trajectories for Event Recognition.* Image and Vision Computing, 14(8):609–615, August 1996.

[Kal60] Kalman, Rudolph Emil: *A New Approach to Linear Filtering and Prediction Problems.* Transactions of the ASME–Journal of Basic Engineering, 82(Series D):35–45, 1960.

[LK81] Lucas, Bruce und Takeo Kanade: *An iterative image registration technique with an application to stereo vision.* In: *DARPA Image Understanding Workshop*, Seiten 121–130, 1981.

[May79] Maybeck, Peter (Herausgeber): *Stochastic models, estimation, and control*, Band 1. Academic Press, New York, 1979.

[MEB04] Makris, Dimitrios, Tim Ellis und James Black: *Bridging the gaps between cameras.* In: *IEEE Conference on Computer Vision and Pattern Recognition*, Band II, Seiten 205–210, 2004.

[MGPB+05] McCowan, Iain, Daniel Gatica-Perez, Samy Bengio, Guillaume Lathoud, Mark Barnard und Dong Zhang: *Automatic Analysis of Multimodal Group Actions in Meetings.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(3):305–317, März 2005.

[OHG02] Oliver, Nuria, Eric Horvitz und Ashutosh Garg: *Layered Representations for Human Activity Recognition.* In: *International Conference on Multimodal Interfaces*, Seiten 3–8, 2002.

[ORP00] Oliver, Nuria, Barbara Rosario und Alex Pentland: *A Bayesian Computer Vision System for Modeling Human Interactions.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):831–843, August 2000.

[Rab89] Rabiner, Lawrence: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.* Proceedings of the IEEE, 77(2):257–286, 1989.

[RMEJ05] Renno, John-Paul, Dimitrios Makris, Tim Ellis und Graeme Jones: *Application and Evaluation of Colour Constancy in Visual Surveillance.* In: *International Workshop on Performance Evaluation of Tracking and Surveillance*, Seiten 301–308, 2005.

[SB03] Shi, Yifan und Aaron Bobick: *P-Net: A Representation for Partially-Sequenced, Multi-stream Activity.* IEEE Workshop on Detection and Recognizing Events in Video, 4:40, 2003.

[SFH03] Schulz, Dirk, Dieter Fox und Jeffrey Hightower: *People Tracking with Anonymous and ID-Sensors Using Rao-Blackwellised Particle Filters.* In: *International Joint Conferences on Artificial Intelligence*, Seiten 921–928, 2003.

[SG00] Stauffer, Chris und Eric Grimson: *Learning Patterns of Activity Using Real-Time Tracking.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):747–757, 2000.

[SHM+04] Shi, Yifan, Yan Huang, David Minnen, Aaron Bobick und Irfan Essa: *Propagation Networks for Recognition of Partially Ordered Sequential Action.* In: *IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 862–869, 2004.

[ST93] Shi, Jianbo und Carlo Tomasi: *Good Features to Track.* Technischer Bericht, Cornell University, November 1993.

[ST94] Shi, Jianbo und Carlo Tomasi: *Good Features to Track.* In: *IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 593–600, Juni 1994.

[VJ01] Viola, Paul und Michael Jones: *Robust Real-Time Face Detection.* In: *International Conference On Computer Vision*, Seiten 747–747, Juli 9–12 2001.

[WA05] Wilson, Daniel und Christopher Atkeson: *Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors.* In: *International Conference on Pervasive Computing*, Seiten 62–79, 2005.

[WSStCPC04] Waibel, Alexander, Hartwig Steusloff, Rainer Stiefelhagen und the CHIL Project Consortium: *CHIL - Computers in the Human Interaction Loop.* In: *International Workshop on Image Analysis for Multimedia Interactive Services*, 2004.

[ZGPBM05] Zhang, Dong, Daniel Gatica-Perez, Samy Bengio und Iain McCowan: *Semi-Supervised Adapted HMMs for Unusual Event Detection.* In: *IEEE Conference on Computer Vision and Pattern Recognition*, Band I, Seiten 611–618, 2005.

[ZGPBM06] Zhang, Dong, Daniel Gatica-Perez, Samy Bengio und Iain McCowan: *Modeling Individual and Group Actions in Meetings With Layered HMMs.* IEEE Transactions on Multimedia, 8(3):509–520, Juni 2006.