UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FAKULTÄT FÜR
ELEKTROTECHNIK,
INFORMATIK UND
MATHEMATIK

Akustische Szenenanalyse für die ambiente Kommunikation im vernetzten Haus

Zur Erlangung des akademischen Grades

DOKTORINGENIEUR (Dr.-Ing.)

der Fakultät für Elektrotechnik, Informatik und Mathematik
der Universität Paderborn
genehmigte Dissertation
von

Dipl.-Ing. Jörg Schmalenströer
Paderborn

Referent: Prof. Dr.-Ing. Reinhold Häb-Umbach
Korreferent: Prof. Dr.-Ing. Gernot A. Fink

Tag der mündlichen Prüfung: 09.03.2010

Paderborn, den 15.03.2010
Diss. EIM-E/264

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit im Fachgebiet Nachrichtentechnik der Universität Paderborn. Sie wurde im Rahmen des europäischen Forschungsprojektes Amigo (IST 004182) gefördert.

Mein besonderer Dank gilt dem Fachgebietsleiter Herrn Prof. Dr.-Ing. Reinhold Häb-Umbach für die Betreuung dieser Arbeit. Die vielen gemeinsamen Diskussionen und der rege Ideenaustausch führten zu einer sehr guten Arbeitsatmosphäre und trugen entscheidend zum Erfolg bei. Herrn Prof. Dr.-Ing. Gernot A. Fink möchte ich für die Übernahme des Korreferates und die interessanten Gespräche danken.

Den wissenschaftlichen Mitarbeitern des Fachgebietes Nachrichtentechnik danke ich für die gemeinsame Zeit und ihre fachliche Unterstützung. Insbesondere gilt mein Dank Herrn Dr.-Ing. Valentin Ion und Herrn Dr.-Ing. Ernst Warsitz für die vielfältigen Diskussionen über meine Arbeit. Des Weiteren danke ich meinen Kollegen Herrn Dipl.-Math. Alexander Krüger, Herrn Dipl.-Inf. Sven Peschke, Herrn Dipl.-Ing. Maik Bevermeier, Herrn Dipl.-Ing. Dang Hai Tran Vu und Herrn Dipl.-Ing. Volker Leutnant für ihre konstruktiven Kommentare und ihre Unterstützung. Allen Studierenden, deren Arbeiten ich in den letzten Jahren betreuen durfte, danke ich für ihre motivierte Mitarbeit.

An dieser Stelle möchte ich noch meinen Freunden, speziell Nicole Fröhleke, Björn Kehl und Romina Kehl, für die vielen hilfreichen Kritiken zu meiner Arbeit danken.

Meiner Frau Nicole danke ich für den Rückhalt und die Unterstützung, welche mir gerade in anstrengenden und schwierigen Zeiten viel Kraft für meine Arbeit gab. Meiner Tochter Lea danke ich für ihre Liebe und die vielen kleinen Ablenkungen. Besonders möchte ich meinen Eltern danken, deren Vertrauen und kontinuierliche Unterstützung mich während des Studiums und der anschließenden Arbeit an meiner Promotion bestärkte. Sie haben für mich diesen Weg erst ermöglicht.

Inhaltsverzeichnis

“Ambient intelligence refers to the presence of a digital environment that is sensitive, adaptive, and responsive to the presence of people. Within a home environment, ambient intelligence will improve the quality of life of people by creating the desired atmosphere and functionality via intelligent, personalized inter-connected systems and services.”

Emile Aarts, Philips Research [Aar09]

“This technology will recognize us, notice our habbits, learn our likes and dislikes, and adapt its behaviour and the services it offers us accordingly.”

Stefano Marzano über Intelligente Dienste [AM04]

1 Einleitung

Im Rahmen dieser Arbeit wird ein neues Verfahren zur Informationsgewinnung aus akustischen Signalen vorgestellt. Die gewonnenen Informationen geben Aufschluss über anwesende Personen und stattgefundene Ereignisse sowie deren Position im Raum. Anschließend wird die Integration dieser Informationsquelle in eine vernetzte Hausumgebung gezeigt und in den Kontext der ambienten Intelligenz gesetzt. Aufbauend auf den Informationsquellen der Hausumgebung wird abschließend ein audio-visuelles Kommunikationssystem vorgestellt. Dieses nutzt die im Haus vorhandenen Informationsquellen zur Realisierung einer kontextbewussten Steuerung der Kommunikation.

Das Paradigma ambiente Intelligenz (AI) formuliert das Konzept einer vernetzten Umgebung, welche intelligent auf Personen und Ereignisse reagiert. Dabei soll das System sensitiv gegenüber Wünschen und Bedürfnissen der Nutzer sein und auf diese adaptiv reagieren, so dass eine Steigerung des Komforts und der Lebensqualität für den Nutzer erfahrbar wird [AM04]. Diese weitreichende Definition von ambienter Intelligenz umfasst Forschungsthemen sowohl im Bereich der Hardware- als auch der Softwareentwicklung. Verwandte Forschungsbereiche mit starken Überschneidungen im Aufgabenspektrum sind *Ubiquitous Computing* bzw. *Pervasive Computing* [Wei99]. Beide Begriffe beschreiben eine Vernetzung und Durchsetzung alltäglicher Gegenstände mit Mikroprozessoren und Sensoren, wobei der Begriff des *Pervasive Computing* vornehmlich durch die Industrie geprägt wurde. Geräte sollen sich automatisch untereinander vernetzen und eine allgegenwärtige Kapazität an Rechenleistung bereitstellen. Diese hardwareorientierte Sichtweise unterscheidet das *Ubiquitous Computing* von dem Paradigma ambiente Intelligenz. Im Sinne der ambienten Intelligenz ist eine vernetzte Hardware eine notwendige Grundlage für ein System, jedoch soll diese in den Hintergrund treten und möglichst aus dem Wahrnehmungsfeld des Nutzers verschwinden. Die Funktionen und Dienste der Hardware sollen bei diesem Prozess erhalten bleiben. Zusätzlich soll eine starke Orientierung auf den Benutzer erfolgen. Die Nutzung einer Funktion soll intuitiver werden, so dass dem Nutzer das Erlernen eines Bedienschemas abgenommen wird, indem das System sich „intelligent“ verhält [Ami06].

Die zentralen Eigenschaften der ambienten Intelligenz sind durch Integration, Kontextbewusstsein, Personalisierung, Adaptivität und Antizipation gegeben [AM04]. Zunächst soll ein System aus dem Wahrnehmungsbereich der Nutzer entfernt werden, indem die Hardware in die Umgebung oder Dinge des täglichen Lebens vollständig integriert wird. Diese unauffällige Bereitstellung von Funktionen und Diensten führt zu einer verbesserten Akzeptanz der Technik, da sie dem Nutzer weniger aufdringlich erscheint. Das Kontextbewusstsein ist der Schlüssel zu einer aus der Sicht des Nutzers als „intelligent“ wahrgenommenen Umgebung. Ein kontextbewusstes System ist dadurch gekennzeichnet, dass es entsprechend der verfügbaren Informationen Entscheidungen trifft und auf aktuelle Ereignisse reagiert. Das Verhalten des Systems ist somit nicht nur abhängig von den Eingaben des Nutzers, sondern

auch von dem aktuellen Kontext, in dem das System genutzt wird. Da das System kontextbewusst handeln soll, muss es folglich Regeln beinhalten, die entweder vom Nutzer vorgegeben oder selbstständig gelernt werden. Diese Personalisierung ist eine aus dem Paradigma ambiente Intelligenz abgeleitete Notwendigkeit, da das System sich dem Nutzer anpassen soll und nicht umgekehrt. Damit verbunden ist die Eigenschaft der Adaptivität, welche die Fähigkeit beschreibt, auf den Benutzer zu reagieren und sich seinem Verhalten anzupassen. Somit wird die Adaption auf den Benutzer zwangsläufig zu einer Personalisierung führen. Die sicherlich am schwierigsten zu realisierende Eigenschaft der ambienten Intelligenz ist die Antizipation. Das System soll die Absichten und Wünsche von Benutzern prognostizieren und vorausschauende Entscheidungen treffen. Dies bedingt zunächst eine große Menge an Informationen über den aktuellen Kontext und eine entsprechende Beschreibung der möglichen zukünftigen Ereignisse basierend auf den vorhandenen Informationen. Häufige Fehlentscheidungen und dadurch ausgelöste Reaktionen des Systems werden zwangsläufig zu einer Ablehnung des Systems durch den Nutzer führen, da aus der Wahrnehmung des Nutzers heraus das System „irrational“ agiert. Die Realisierung von ambienter Intelligenz bedingt somit grundsätzlich eine Verfügbarkeit von verlässlichen Informationen.

Die Europäische Union unterstützt die Forschung im Bereich ambienter Intelligenz im Rahmen der *Information Society Technologies* (*IST*) Projekte. Das 6. Rahmenprogramm beinhaltete unter anderem das mittlerweile abgeschlossene Projekt Amigo [Ami06], dessen Untertitel „*Ambient Intelligence for the networked home environment*“ die Zielvorgaben des Projektes verdeutlicht. Das Projekt Amigo hatte das Ziel, die Vorteile einer vernetzten Umgebung für den Benutzer erfahrbar zu machen, indem intelligente Dienste auf Basis einer *Middleware* entwickelt wurden. Eine *Middleware* ist dabei eine Software, welche im Hintergrund, d. h. vor dem Anwender verborgen, Systemkomponenten miteinander verknüpft. Die vorliegende Arbeit stellt Teile der Forschungsergebnisse aus dem Bereich der akustischen Szenenanalyse und der ambienten Kommunikation vor und gibt einen Einblick in die Mechanismen der Amigo *Middleware*.

Obwohl schon häufiger prognostiziert, haben Systeme zur Realisierung von ambienter Intelligenz den Weg in den Massenmarkt noch nicht gefunden. Im Projekt Amigo wurde als eines der Haupthindernisse hierfür die fehlende Interoperabilität von Geräten unterschiedlicher Hersteller identifiziert. Trotz fortschreitender Entwicklung im Bereich der Vernetzung entwickeln viele Hersteller isolierte Lösungen, welche auf das eigene Produktportfolio abgestimmt sind. Infolgedessen sind die in einem Haushalt vorhandenen Geräte, welche sich in die Kategorien Haushaltsgeräte, Unterhaltungselektronik, mobile Geräte und Personal Computer einteilen lassen, oft isoliert voneinander anstatt einen Verbund darzustellen [Ami06]. Aktuelle Entwicklungen führen zwar vermehrt zur Vernetzung von Geräten, wie z. B. zwischen Computern und Unterhaltungselektronik, jedoch ist dies kein Weg zur allgemeinen Interoperabilität, sondern eine eher harte Verknüpfung über proprietäre Protokolle. Im Projekt Amigo wurde daher eine quelloffene, standardisierte und interoperable *Middleware* entwickelt, welche mit den auf dem Markt etablierten *Middleware*-Technologien sowohl interagieren als auch diese miteinander verknüpfen kann.

Das Bindeglied der ambienten Intelligenz ist eine *Middleware*, welche die im Haus vorhandenen Sensoren, Geräte, Dienste und Applikationen untereinander verbindet. Folglich sorgt sie dafür, dass die in den Sensoren und Diensten gewonnenen Informationen im gesamten Netz verfügbar sind. Neben Messwert nehmenden Sensoren, wie z. B. Temperaturfühlern, sind in der vernetzten Hausumgebung auch komplexere Sensoren in Form von

Mikrophonen und Kameras vorstellbar. Diese erfordern im Vergleich zu Messwertsensoren spezielle Analyseverfahren zur Auswertung der aufgenommenen Daten. Im Falle von Mikrophondaten ist dies die akustische Szenenanalyse und für die Videodaten sind dies Verfahren zur visuellen Personen- oder Objekterkennung. Die akustische Szenenanalyse hat das Ziel, die in einem akustischen Signal enthaltenen Quellen zu identifizieren und alle nutzbaren Daten zu extrahieren. Entstanden ist dieses Forschungsgebiet aus dem Bestreben, die automatische Spracherkennung zu verbessern, indem eine bessere Identifikation der Störquellen vorgenommen wird [RO98]. Betrachtet man die akustische Szenenanalyse aus dem Blickwinkel der ambienten Intelligenz, so kann diese als eine wertvolle Informationsquelle für kontextuelle Zusammenhänge gesehen werden. Vorteilhaft hierbei ist, dass Mikrophone als Sensoren unauffällig in die Umgebung integriert werden können. Dabei erfassen sie den gesamten Raum und sind unabhängig von den Beleuchtungsverhältnissen, wodurch sie Informationen liefern, die durch Kamerasysteme nicht erfassbar sind. Die in der akustischen Szenenanalyse gewonnenen Daten geben Aufschluss über Benutzer, deren Aktivitäten und auftretende Ereignisse.

Mikrophone sind als Sensoren für die akustische Szenenanalyse notwendig, jedoch ist die Nutzung nicht auf die reine Informationsgewinnung beschränkt. In Kombination mit Lautsprechern und Netzwerktechnik ist der Aufbau verteilter Kommunikationssysteme möglich. Orientiert sich solch ein System an den Ideen der ambienten Intelligenz, so wird es durch den Begriff „ambiente Kommunikation“ charakterisiert. Die Grenzen zwischen der „klassischen“ Kommunikation über Internetprotokolle (engl. *Voice over Internet Protocol*, *VoIP*) und der „ambienten Kommunikation“ sind fließend, da in beiden Verfahren vergleichbare Komponenten eingesetzt werden.

Ein Merkmal der ambienten Kommunikation ist die nicht vorhandene Bindung des Gesprächs an ein dediziertes Gerät, wie z. B. ein Telefon. Der Nutzer muss nicht mehr ein Gerät für die Funktion der Kommunikation aufsuchen, stattdessen tritt die Hardware in den Hintergrund und die reine Funktionalität bleibt bestehen. Folglich kann der Nutzer jederzeit eine Kommunikation starten und sich währenddessen frei bewegen. Das System setzt somit eine Freisprechfunktion und einen über mehrere Räume verteilten Aufbau voraus.

Ein weiteres Merkmal der ambienten Kommunikation resultiert aus den Benutzerstudien des Projektes Amigo [M$^+$05]. Vielfach wurde durch die Testpersonen der Wunsch geäußert, eine „intelligente Umgebung“ solle den Kontakt zu Freunden und nahen Verwandten unterstützen. Hieraus entstand die Idee einer kontinuierlichen Verbindung zwischen räumlich entfernten, jedoch emotional nahe stehenden Personen, die ein Gefühl des „Verbunden-Seins“ erzeugen soll. Hierbei ist die Menge der ausgetauschten Informationen zwischen den Personen über die Zeit betrachtet geringer als bei einem klassischen Telefongespräch. Die Kommunikation ist fortlaufend aktiv und die Personen hören, was der entfernte Partner macht. Somit entsteht bei beiden das Gefühl, dass der jeweils andere sich im Nebenraum befindet. Denkbar ist zum Beispiel, dass das System automatisch die Verbindung zwischen zwei Personen etabliert, sobald beide von der Arbeit nach Hause kommen und jeweils, entsprechend der persönlichen Systemkonfigurationen, bei bestimmten Ereignissen die Verbindung automatisch trennt.

Die Kommunikation kann sowohl durch explizite wie auch implizite Benutzereingaben kontrolliert werden. Die explizite Interaktion beinhaltet die klassische Steuerung der Kommunikation durch den Benutzer, die durch direkte Eingaben, z. B. über einen berührungsempfindlichen Bildschirm, gekennzeichnet ist. Die implizite Steuerung versucht das System

intuitiver für den Benutzer zu gestalten, indem aus dem Verhalten des Nutzers die implizierten Befehle ermittelt werden. Vorstellbar ist zum Beispiel der automatische Aufbau einer Kommunikation, wenn sich der Nutzer auf ein Bild des gewünschten Kommunikationspartners zubewegt.

Die vorliegende Arbeit behandelt Aspekte aus den Themengebieten akustische Szenenanalyse, *Middleware* und ambiente Kommunikation und gliedert sich in die folgenden Kapitel: In Kap. 2 wird ein Überblick über den aktuellen Stand der Forschung in den Bereichen akustische Szenenanalyse, ambiente Intelligenz, *Middleware* und ambiente Kommunikation gegeben. Die wissenschaftlichen Ziele dieser Arbeit werden im darauffolgenden Kap. 3 definiert. Das Kap. 4 stellt die Verfahren zur akustischen Szenenanalyse vor und fasst die experimentellen Ergebnisse in diesem Bereich zusammen. In Kap. 5 werden Aspekte der akustischen Ereignisdetektion als ein spezieller Teil der akustischen Szenenanalyse näher untersucht. Das Amigo System und die Verknüpfung der akustischen Szenenanalyse mit der Amigo *Middleware* werden in Kap. 6 erläutert. Anschließend wird in Kap. 7 gezeigt, wie das Amigo System zur Realisierung eines kontextbewussten Dienstes genutzt werden kann. Das hierbei betrachtete Beispiel der ambienten Kommunikation verwendet sowohl akustische als auch visuelle Daten. Eine Zusammenfassung der Ergebnisse dieser Arbeit erfolgt abschließend in Kap. 8.

2 Stand der Forschung

Diese Arbeit behandelt die Themengebiete akustische Szenenanalyse, ambiente Intelligenz, *Middleware* und ambiente Kommunikation. Dabei sollen die für die ambiente Intelligenz zu entwickelnden Komponenten der *Middleware* die Informationsgewinnung mittels der akustischen Szenenanalyse mit der Anwendung, der ambienten Kommunikation, verknüpfen. Im Folgenden wird ein Überblick über den Stand der Forschung in den einzelnen Themengebieten gegeben.

2.1 Akustische Szenenanalyse

Die akustische Szenenanalyse ist auf Grund der unterschiedlichen Anwendungsgebiete ein weit gefächertes Forschungsgebiet. Zunächst wurde es durch die *DARPA* im Rahmen der „*Rich Transcription Task*“ gefördert. Das vorgegebene Ziel war hierbei, eine automatische Zuordnung von Zeitabschnitten zu Sprechern (sog. Annotation) in Rundfunksendungen, Telefongesprächen und Besprechungen durchzuführen [NIS08b, TR06].

Bedingt durch die Verfügbarkeit neuer Datenquellen, welche in Besprechungsräumen und intelligenten Umgebungen zu finden sind, wandelten sich die Ansätze von unimodalen zu multimodalen Signalverarbeitungssystemen (vgl. Abb. 2.1). Waren in Telefongesprächen

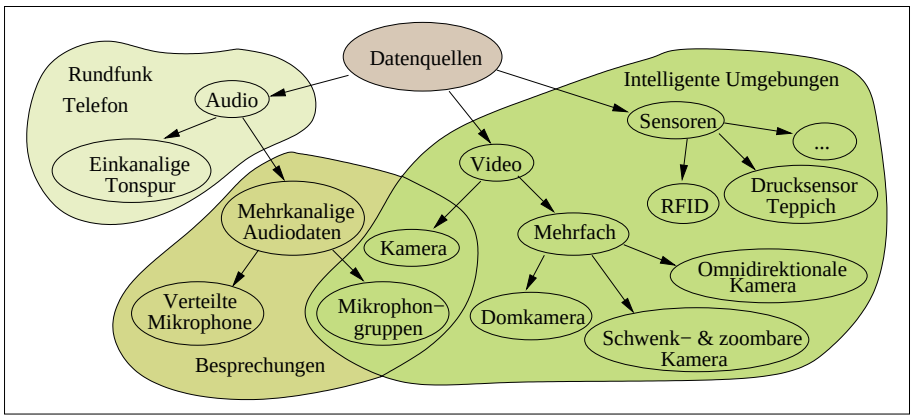


Abbildung 2.1: Datenquellen und Anwendungsgebiete der akustischen Szenenanalyse

und Rundfunksendungen nur einkanalige akustische Aufnahmen vorhanden, so bieten viele Datenbasen von aufgezeichneten Besprechungen schon mehrkanalige Aufnahmen. Eine erneute Steigerung der Vielfalt der Sensoren ist in intelligenten Umgebungen zu verzeichnen. Dabei kann die Ausstattung der Umgebungen stark variieren, wodurch eine Anpassung der Systeme und Algorithmen zur Datenverarbeitung an die gegebene Sensorik notwendig ist.

Am deutlichsten wird dies bei den visuellen Daten, wo neben Kameras mit festen Blickwinkeln auch schwenk- und zoombare (engl. *Pan Tilt Zoom*, *PTZ*) Kameras oder omnidirektionale Kameras eingesetzt werden. Somit findet eine Spezialisierung der Systeme auf die vorhandene Sensorik und den Verwendungszweck statt.

Aktuelle Projekte, wie das *IST* Projekt *CHIL* (*Computer in the Human Interaction Loop*) [CHI04] oder das Projekt *AMI* (*Augmented Multi-Party Interaction*) [AMI04], erforschen professionelle Arbeitsumgebungen, wie zum Beispiel Seminar- oder Besprechungsräume. Ziele sind unter anderem die Verbesserung der automatischen Spracherkennung, die akustische und visuelle Lokalisation von Personen, sowie die Identifikation von Personen und Ereignissen [OSBC06, TMZ+06, B+05b]. Des Weiteren wird im Projekt *DIRAC* (*Detection and Identification of Rare Audiovisual Cues*) [DIR06] an der Detektion und Identifikation seltener akustischer und visueller Ereignisse gearbeitet.

Anwendungsgebiete der akustischen Szenenanalyse mit multimodalen Daten sind beispielsweise verbesserte Video-Konferenzsysteme, automatische Überwachungssysteme und Systeme zur Unterstützung älterer oder behinderter Menschen [KTVL07]. Ein weiteres Forschungsgebiet ist die automatische Annotation von Videomaterial aus Fernsehsendungen [KMK07, MMF+06]. Im Folgenden wird ein Überblick über die grundlegenden Komponenten eines Systems zur akustischen Szenenanalyse und deren Stand der Forschung gegeben.

Eine Lokalisierung von Personen durch aufgenommene, akustische Signale kann durch die Schätzung der Signallaufzeitdifferenzen zwischen Mikrophonpaaren erfolgen. Hierzu werden z. B. im „*Generalized Cross Correlation*“-Verfahren die Korrelationen zwischen den Signalen berechnet und durch das Wissen über die Position der Mikrophone eine Positionsschätzung durchgeführt [KC76]. Zusätzlich kann nach Bedarf eine modellbasierte Nachfilterung durch Kalman- oder Partikelfilter erfolgen, um die Genauigkeit der Positionsschätzung zu erhöhen [WPH04].

Eine Identifikation von Sprechern und Ereignissen basiert zumeist auf einer Modellierung der Klassen durch Gauß’sche Mischungsverteilungen (engl. *Gaussian Mixture Model*, *GMM*) [Cam97]. Diese können einzeln, also für jede Klasse unabhängig, trainiert oder aber von einem gemeinsamen (universellen) Hintergrundmodell adaptiert werden [RQD00]. Die Modellbildung durch ein universelles Hintergrundmodell (engl. *Universal Background Model*, *UBM*) bietet den Vorteil, dass weniger Daten für das Training benötigt werden und eine rudimentäre Erkennung von unbekannten Klassen erfolgen kann. Das Training wird mit dem „*Expectation Maximization*“-Algorithmus (*EM*-Algorithmus) oder der Bayes’schen Adaption durchgeführt [DHS01].

Neuere Verfahren zur Parameterschätzung der Klassenmodelle mit dem Ziel der Reduktion der Fehlerrate stammen aus dem Bereich diskriminativer Lernverfahren. Bekannte Ansätze sind das „*Minimum Classification Error*“-Training (*MCE*-Training) und das „*Maximum Mutual Information*“-Training (*MMI*-Training). Sowohl das *MCE*- als auch das *MMI*-Training finden erfolgreich Anwendung im Bereich der automatischen Spracherkennung [LP96], der Sprecheridentifikation [KYM+05] und Sprecherverifikation [MC03]. Dabei können die diskriminativen Lernverfahren sehr langsam konvergieren oder im Extremfall auch divergieren, falls keine geeigneten Gegenmaßnahmen getroffen werden [NCM91].

Bevor jedoch eine Identifikation von Sprechern erfolgen kann, muss zunächst eine Einteilung der akustischen Daten in homogene Abschnitte, die sog. Segmentierung, erfolgen. Ein homogener Abschnitt beinhaltet dabei nur Daten einer Klasse und kann folglich eindeutig klassifiziert werden. Die zur Segmentierung verwendeten Verfahren nutzen häufig

das Bayes'sche Informationskriterium (engl. *Bayesian Information Criterion*, *BIC*), welches auf einem Hypothesentest basiert [CG98, DW00]. Hierbei wird die Hypothese, dass an einem Punkt im beobachteten Zeitabschnitt ein Sprecherwechsel vorliegt und somit der erste Teil des Zeitfensters aus einer Klasse und der zweite Teil des Zeitfensters aus einer anderen Klasse stammt, der Hypothese gegenübergestellt, dass das gesamte Fenster aus einer Klasse stammt. Verfahren, die auf *BIC*-Ansätzen basieren, nehmen dabei immer eine Abwägung zwischen den Aspekten Genauigkeit, Verlässlichkeit und Latenz der Segmentierung vor [LZ02, DY08].

Fasst man Sprecherwechseldetektion und Sprecheridentifikation als eine Aufgabe auf, so wird dies als Sprecherprotokollierung (engl. *speaker diarization*) bezeichnet [PAW07]. Dabei wird versucht, durch eine automatische Annotation vorhandene Audio- oder Videodaten so aufzubereiten, dass sie mit textbasierten Suchalgorithmen erfasst werden können [TR06]. Die Kombination einer Identifikation von Sprechern mit einer automatischen Spracherkennung und die Auswertung der Metadaten des Videomaterials liefern die Information „Wer spricht Wann und Was?". Hierbei können akustische Modelle für verschiedene Sprecher vorab trainiert werden, um deren Anteile in den Audiodaten zu finden, wie es zum Beispiel die Protokollierung von Besprechungsdaten erfordert. Alternativ kann auch die Aufgabe gestellt sein, dass alle Anteile eines Sprechers durch eine eindeutige Kennung gekennzeichnet werden sollen, ohne die Anzahl der Sprecher oder deren Identität vorab zu kennen [SML+08, RT05].

In der Sprecherprotokollierung sind iterative Verfahren mit variierender Komplexität weit verbreitet, die zwei unterschiedliche Ansätze verwenden. Die eine Möglichkeit („*top-down*") ist, die gesamten Daten an den wahrscheinlichsten Sprecherwechselpunkten, z. B. durch eine Detektion des Sprechergeschlechts, aufzuteilen und somit mehrere Teile zu erhalten. Anschließend werden die Teile erneut auf Sprecherwechselpunkte untersucht und aufgeteilt [MMF+06]. Die andere Möglichkeit („*bottom-up*") ist, die sehr feine Vorsegmentierung der Daten in kleinste, homogene Abschnitte und das anschließende Clustern der Segmente, so dass zusammenhängende Abschnitte eines Sprechers wieder in einem Segment zusammengefasst werden [STGW05]. In beiden Verfahren werden Schwellwerte oder Grenzen festgelegt, die das iterative Verfahren stoppen, sobald die vermutlich optimale Segmentierung gefunden ist. Verfahren zur Sprecherprotokollierung, die auf Datenströmen arbeiten, verwenden beispielsweise *Hidden Markov Models* (*HMM*) zur Modellierung der Sprechergruppe. In [MMF+06] wird ein Verfahren vorgestellt, in dem je ein Zustand eines *HMM* einen Sprecher repräsentiert und das bei einem neu auftretenden Sprecher um einen weiteren Zustand erweitert wird. Die Transitionswahrscheinlichkeiten des *HMM* werden in diesem Fall aus Trainingsdaten geschätzt und sind für jeden Zustandsübergang fest vorgegeben. Die Grundlage einer jeden Identifikation ist eine Menge von Sprechermodellen, welche entweder vorab trainiert oder während des Betriebs geschätzt werden. Eine echtzeitfähige Bildung von Sprechermodellen auf fortlaufenden Datenströmen wird in [LZ02] vorgestellt. Verfahren, die auf Datenströmen arbeiten, haben jedoch im Vergleich zu iterativen Ansätzen immer den Nachteil, dass keine Korrekturen vergangener Entscheidungen durch erneute Iterationen oder Clusterungen möglich sind.

Eine Sprecherprotokollierung kann durch Nahbereichsmikrophone oder durch entfernte Mikrophone erfolgen, die in Gruppen angeordnet oder auf einem Tisch verteilt sind. Dabei kann entweder eine Auswahl des besten Mikrophonsignals oder eine Signalverbesserung durch strahlformende Algorithmen verwendet werden, um die Leistungsfähigkeit des Sys-

tems zu steigern [AWH07].

Ein aktuelles Thema in der Forschung ist die multimodale Signalverarbeitung in „intelligenten Umgebungen“, wo neben Mikrophonen und Kameras auch andere Sensoren verfügbar sind. Ein Schwerpunkt dieses Forschungsthemas liegt bei der Positionsschätzung von Personen durch akustische und/oder visuelle Daten [KFH+08]. Die Positionsinformationen von Sprechern können dann direkt für die Segmentierung von Audiodaten genutzt werden [AFI+08] oder aber integriert in den akustischen Merkmalsvektor zu einer Verbesserung der Sprecherprotokollierung führen [PAW06, APW06]. Die Positionsinformationen können wahlweise aus Laufzeitschätzungen zwischen Mikrophonen [PAW06], Kamerasystemen [SML+08] oder anderen Systemen, wie dem in [CSJ07] vorgeschlagenen „*Radio Frequency Identification*“-System (*RFID*-System), stammen. Ansätze für die Sprecherprotokollierung mit Audio- und Videodaten können in [NK07] und [FHY09] gefunden werden. Entsprechend der verfügbaren Hardware in den Räumen unterscheiden sich die Systeme und Verfahren deutlich. In [SML+08] wird z. B. ein System mit fest installierten Kameras genutzt, bei dem das Gesicht eines Nutzers beim Betreten des Raumes mit einer Kamera identifiziert und anschließend die Position des identifizierten Nutzers über andere Kameras verfolgt wird. Der Ansatz in [BS07] verwendet im Kontrast dazu schwenkbare Kameras und versucht kontinuierlich die im Raum befindlichen Personen zu identifizieren. Ein weiterer Aspekt der Sprecherprotokollierung in „intelligenten Umgebungen“ ist die Verfügbarkeit multipler Datenquellen, die im Falle von Mikrophonen eine Auswahl oder Kombination von Kanälen erfordert. Hierzu wurde in [GAW06] ein Ansatz mit einem Viterbi-Dekodierer vorgeschlagen, der eine automatische Kanalauswahl durchführt. Alternativ gibt es eine Vielzahl von Ansätzen zur Gewichtung, Normierung und Kombination multimodaler Informationen, wovon einige in [EFJS07] untersucht wurden.

2.2 *Middleware* und ambiente Intelligenz

Die Entwicklung von Anwendungen und Diensten in der vernetzten Hausumgebung setzt vermehrt auf dienstorientierte Architekturen. Diese sind in der Lage, in heterogenen Umgebungen Geräte und Dienste miteinander zu verbinden und so die Inkompatibilitäten zwischen unterschiedlichen Herstellern zu überwinden [MKGI07, Car08].

Die Verwendung von *Webservices* wird hierbei als eine mögliche Schlüsselkomponente gesehen, da die aus dem Bereich des *World Wide Web* (*WWW*) bekannt gewordenen Dienste offene und standardisierte Schnittstellen und Beschreibungen bieten [PTDL07]. Der Datenaustausch zwischen den Softwarediensten erfolgt dabei durch das offene *Simple Object Access Protocol* (*SOAP*) [G+07]. Des Weiteren können die Dienste und die zugehörigen Schnittstellen durch die *Web Services Description Language* (*WSDL*) [C+07] beschrieben werden. Damit Dienste einander in einem gemeinsamen Netz finden, ist ein zentraler Anlaufpunkt im System notwendig, der in Form eines Verzeichnisdienstes, wie z. B. dem *Lightweight Directory Access Protocol* (*LDAP*) [Z+06], realisiert werden kann.

Im Bereich *Middleware* gibt es verschiedene Standards mit unterschiedlichen Verbreitungsgraden, wobei *Universal Plug and Play* (*UPnP*) [UPn08] eine weit verbreitete Technologie ist. *UPnP* bietet Mechanismen zur Lokalisierung, Beschreibung, Steuerung und Ereignismeldung von Diensten und Geräten. Ein Anwendungsgebiet ist die Verteilung von Medieninhalten und die Steuerung von Unterhaltungselektronik. Im Bereich Gebäudeauto-

matisierung sind Bussysteme wie der *European Installation Bus* (*EIB*) [EIB09] verbreitet, wobei der Einsatz aus Kostengründen meist auf professionelles Gebäudemanagement beschränkt ist. Beide *Middleware*-Technologien sind zwar führend in ihrer Domäne, jedoch sind sie zueinander inkompatibel und nur durch spezielle Verfahren miteinander verknüpfbar [RBH03].

Insgesamt wird die Entwicklung von ambienter Intelligenz durch die Inkompatibilität zwischen Diensten und Systemen unterschiedlicher Hersteller gehemmt [Ami06]. Dies ist einer der Gründe für die Unterstützung des Projektes Amigo durch die Europäische Union. Die Entwicklung von ambienter Intelligenz beinhaltet ein breites Spektrum an offenen Fragestellungen im Bereich der Software- und Hardwareentwicklung [FCP+05]. Aktuelle Systeme können zwar einzelne Aufgabenstellungen in vernetzten Umgebungen handhaben, jedoch verwenden diese Ansätze zur Lösung der Problemstellungen feste von den Herstellern vorgegebene Ansätze mit eingeschränkter Flexibilität [EK05]. Ein Beispiel hierfür ist der *EIB*, welcher die Möglichkeit bietet, physikalische Informationen von Sensoren, wie z. B. Lichtsensoren, zu sammeln und Komponenten mit aktorischen Fähigkeiten, wie z. B. Türschließsysteme, anzusteuern [EIB09].

2.3 Ambiente Kommunikation

Die ambiente Telefonie, wie sie in [Här07] vorgestellt wurde, beschreibt eine neue Form der Kommunikation, welche auf der Kombination von *VoIP*-Technologien und Freisprechtechnologien basiert. Die Verbreitung von Breitbandanschlüssen ermöglicht unbegrenzt Gespräche über *VoIP*-Technologien zu führen, wobei die Kosten auf einen festen Betrag für den Breitbandanschluss begrenzt sind[1]. Die damit verbundene Abkehr von Verbindungspreisen hin zu festen Grundpreisen für die Versorgung mit Datenanschlüssen beeinflusst das Verhalten der Benutzer derart, dass Verbindungen im Vergleich zur Festnetztelefonie länger, wenn nicht sogar praktisch unbegrenzt, geführt werden [GDJ06]. Infolgedessen tritt das intensive Gespräch zwischen zwei Menschen während der Kommunikation in den Hintergrund und die Menge an ausgetauschten Informationen pro Zeit wird geringer. Der Charakter einer Verbindung wandelt sich vom reinen Medium zum mündlichen Informationsaustausch zum System, das zwei räumlich getrennte Orte verbindet [BFGP08].

Die hierzu benötigten Technologien verwenden häufig das *Real-Time Transport Protocol* (*RTP*) [S+03] zur Datenübertragung und das *Session Initialization Protocol* (*SIP*) [R+02] zum Sitzungsaufbau und zur Sitzungsverwaltung. Des Weiteren existieren eine Vielzahl von Audiokompressionsverfahren, um die Datenrate für eine Verbindung zu senken. Viele Verfahren, wie z. B. das durch die *International Telecommunication Unit* (*ITU*) standardisierte Verfahren G.711, sind auf einen geringen Bandbreitebedarf optimiert und verwenden daher eine Abtastrate von $8\,\mathrm{kHz}$ [Wik09b]. Dazu wird das Signal zunächst auf einen Frequenzbereich zwischen $300\,\mathrm{Hz}$ bis $3400\,\mathrm{Hz}$ begrenzt, wodurch Teile der Sprache und tieffrequente Umgebungsgeräusche unterdrückt werden. Paketorientierte Übertragungsverfahren, wie z. B. *RTP*-Datenströme, verwenden das verbindungslose *Universal Datagram Protocol* (*UDP*), um Verbindungen mit niedrigen Latenzen zu realisieren. Die hiermit verbundenen

[1] Aktuell wird auf Grund des steigenden Kostendrucks ein Umbau der Telekommunikationsnetze zu einer paketvermittelnden Netzinfrastrukur betrieben (engl. *Next Generation Networks*), wodurch auch in der Festnetztelefonie Festpreise für Telefonate ermöglicht werden [NGN09].

Paketverluste sind abhängig von der Netzqualität und werden von einigen Audiokompressionsverfahren automatisch durch eine Fehlerverschleierung (engl. *Packet Loss Concealment*, *PLC*) kompensiert [Spe09].

Einige neuere Kompressionsverfahren, wie z. B. das quelloffene Verfahren *Speex* [Spe08], besitzen die Option breitbandige Signale, d. h. Signale mit einer Abtastrate von $16\,\mathrm{kHz}$ oder sogar $32\,\mathrm{kHz}$, zu komprimieren. Sie bieten somit ein besseres Klangbild als schmalbandige Verfahren. Die höheren Datenraten (z. B. *Speex* $16\,\mathrm{kHz}$: $32\,\mathrm{kBit/s}$ Datenrate je Kanal) stellen keinen Nachteil dar, da aktuelle *ADSL*-Anschlüsse in privaten Wohnungen und Häusern eine genügend hohe Bandbreite bieten [Wik09a]. Ein weiterer Vorteil dieser breitbandigen Audiosignalübertragung ist die Möglichkeit, die neben dem Sprachsignal übertragenen anderen akustischen Ereignisse besser erkennen zu können. Der lokale Sprecher hört nicht nur die Stimme des entfernten Sprechers, sondern auch die Umgebungsgeräusche, welche durch die Aktivitäten des entfernten Sprechers entstehen, wodurch der Charakter der ambienten Kommunikation zusätzlich unterstützt wird [SLH08].

Eine Freisprecheinrichtung erfordert zwingend die Verwendung von Echokompensations- oder Echounterdrückungsverfahren sowie Ansätze zur optionalen Unterdrückung von stationären Störquellen. Ansonsten entstehen störende Rückkopplungen oder Pfeifgeräusche, welche die Qualität des Kommunikationssystems stark beeinträchtigen [BH03]. Ein Ansatz hierfür besteht aus einem vorgeschalteten adaptiven Filter zur Kompensation des ersten Anteils der unbekannten Raumimpulsantwort und einem nachgeschalteten Nachfilter zur Restecho- und Störgeräuschunterdrückung [LK07].

Die Adaptionssteuerung von Filtern zur Echokompensation benötigt neben der zu treffenden Entscheidung ob das wiedergegebene Signal einen aktiven Sprecher enthält, auch Informationen über die Aktivität des lokalen Sprechers [MH00]. Diese sog. *Double-Talk*-Detektion kann durch die Berechnung der Kreuzkorrelation zwischen dem wiedergegebenen Signal und dem aufgenommenen Signal, sowie dem Wissen über die geschätzte Raumimpulsantwort realisiert werden [BMC00]. Eine Sprecheraktivitätsdetektion für den entfernten Sprecher kann durch die Berechnung von Kurzzeit- und Langzeitmittelwerten der Signalenergie implementiert werden [RS04].

3 Wissenschaftliche Ziele

Ziel dieser Arbeit ist die Realisierung einer akustischen Szenenanalyse, deren Informationen über eine *Middleware* an ein System zur ambienten Kommunikation weitergegeben werden. Zunächst werden die Möglichkeiten der akustischen Szenenanalyse zur Informationsgewinnung innerhalb einer vernetzten Hausumgebung untersucht. Anschließend wird das Konzept der Amigo *Middleware* erläutert, speziell die Aspekte des Datenaustausches und der Dienstinteraktion. In diesem Rahmen wird auch die Einbindung der akustischen Szenenanalyse als Informationsquelle im *Middleware*-Konzept herausgestellt. Darauf aufbauend werden die notwendigen Komponenten der ambienten Kommunikation diskutiert und der gesamte Systemaufbau vorgestellt. Im Folgenden werden aufgeschlüsselt nach den Themengebieten akustische Szenenanalyse, *Middleware* und ambiente Kommunikation die einzelnen Aufgabenstellungen näher definiert.

3.1 Akustische Szenenanalyse

Die ambiente Intelligenz in einem Haus soll aktiv und gleichzeitig unauffällig die Bewohner eines Hauses in ihrem täglichen Leben unterstützen und somit den Komfort steigern [AM04]. In dieser Arbeit dienen akustische Signale als Informationsquellen. Sie sollen fortlaufend mit möglichst geringer Latenz ausgewertet werden, um Änderungen im Systemverhalten aufgrund eines detektierten Ereignisses unmittelbar nach dessen Eintritt vornehmen zu können. Der Prozessablauf, von der Signalaufnahme durch die Mikrophone, über die Entstörung und die abschließende Klassifikation, muss zeitlich möglichst schnell erfolgen, so dass die gewonnenen Informationen sofort über die *Middleware* an die ausführenden Applikationen weitergegeben werden können. Eine zu große Verzögerung in der Verarbeitungskette würde die Reaktionen des Systems mit einer Latenz versehen, welche die hilfreichen Intentionen der Applikationen ins Negative verkehren könnte.

Als Beispiel für die negativen Folgen von zu großen Latenzen kann eine einfache Lichtsteuerung durch Sprachbefehle in Kombination mit der akustischen Positionsschätzung betrachtet werden. Angenommen werde ein großes Wohnzimmer mit Essecke und angeschlossenem Kochbereich, so dass sich mehrere Beleuchtungsszenarien ergeben. Ein Benutzer gibt den Befehl zum Anschalten des Lichtes, während er im Kochbereich steht. Das System entscheidet nun anhand der akustischen Positionsschätzung, dass sich der Benutzer im Küchenbereich aufhält und schaltet das Licht dort ein. Reagiert das System langsamer als das Betätigen eines Schalters dauert, so ist der Vorteil der schalterlosen Lichtsteuerung für den Benutzer nicht mehr gegeben, da für ihn die Unannehmlichkeit des Wartens überwiegt.

Aktuelle Verfahren trennen die Aufgabe der Lokalisation von der Identifikation der Sprecher und führen abschließend die Ergebnisse zusammen. Im Rahmen dieser Arbeit wird ein

neuer Ansatz zur kombinierten Identifikation und Positionsschätzung entwickelt, der den zuvor genannten zeitlichen Anforderungen gerecht wird. Dabei wird die Positionsinformation direkt mit in den Identifikationsprozess einbezogen, so dass eine Reduktion der Fehlerrate erzielt wird.

Des Weiteren wird für die Lokalisation ein geeignetes Verfahren auf Basis von strahlformenden Algorithmen ausgewählt, das sowohl eine Verbesserung der Signalqualität als auch eine Positionsschätzung ermöglicht. Dieses wird im Kontext einer vernetzten Hausumgebung hinsichtlich der Genauigkeit mit einem aktuellen Verfahren verglichen.

Da die akustischen Signale sowohl zur Positionsschätzung als auch zur Identifikation von Personen und Ereignissen verwendet werden, wird der Einfluss der akustischen Strahlformung auf den Klassifikationsprozess untersucht. Die hierzu benötigten Merkmale werden sowohl für die Sprecheridentifikation als auch für die Ereignisdetektion verwendet, um der Prämisse der Ressourcen schonenden Verfahren Rechnung zu tragen. Dabei wird untersucht, ob die in der Sprach- und Sprechererkennung verbreiteten Merkmale für eine Identifikation von akustischen Ereignissen verwendet werden können.

Diskriminative Lernverfahren zum Training von Modellen zur Sprecheridentifikation und Spracherkennung erzielen signifikante Verbesserungen durch die Reduktion der Fehlerrate. Dies ist möglich durch das Einbeziehen aller Klassen zum Training jeder einzelnen Klasse, wodurch fehlerhafte Annahmen in der Modellierung und Näherungen kompensiert werden können. Ein Vergleich zwischen diskriminativen Lernverfahren und *ML*-Trainingsverfahren wird zeigen, inwieweit eine Verbesserung der Klassifikationsleistung durch diese erreicht werden kann und wo die Grenzen der Verfahren liegen.

Zusammenfassend kann das Ziel der hier zu entwickelnden akustischen Szenenanalyse als Beantwortung der Frage „Wer spricht Wann und Wo, während Was passiert?“ beschrieben werden, während frühere Ansätze zur Sprecherprotokollierung lediglich die Beantwortung der Frage „Wer spricht Wann?“ zum Ziel hatten.

Auf der einen Seite stellt die ambiente Kommunikation als Echtzeitanwendung hohe Anforderungen an die Latenz der Informationsgewinnung durch die akustische Szenenanalyse. Auf der anderen Seite bietet eine audio-visuelle Kommunikation über die aufgenommenen Videodaten eine weitere Datenquelle zur Verbesserung der akustischen Szenenanalyse. Daher wird im Rahmen dieser Arbeit auch die multimodale Sprecherprotokollierung als Fusion von akustischen und visuellen Daten betrachtet.

3.2 *Middleware* und ambiente Intelligenz

Eine Entscheidung in einem intelligenten System kann nur so gut sein, wie die Menge an Informationen, auf deren Grundlage sie getroffen wurde. Folglich ist ein offenes System zum Informationsaustausch eine wichtige Komponente für die ambiente Intelligenz. Grundgedanke bei der Entwicklung des Amigo Kontextmanagementsystems ist die Annahme, dass in einer heterogenen Umgebung, wie dem vernetzten Haus, eine Vielzahl von zur Zeit ungenutzten Informationsquellen vorhanden ist, durch deren Nutzung die Qualität der ambienten Intelligenz signifikant verbessert werden kann. Dabei muss darauf geachtet werden, ein dynamisches System zu entwickeln, welches dem zeitvarianten Charakter einer Hausumgebung gerecht wird. Kontextquellen können in Form von Geräten in das Haus gebracht oder herausgenommen werden, und müssen folglich dynamisch verwaltet werden. Dies steht im

Kontrast zu anderen *Middleware*-Technologien, wie z. B. *EIB*, bei denen Sensoren und Aktoren fest in die Umgebung integriert sind und keine Dynamik aufweisen. Das in Kooperation mit den Projektpartnern von Amigo entwickelte Kontextmanagementsystem basiert auf den in der Amigo *Middleware* implementierten Methoden zur Nutzung von Diensten. Die Einbindung von Informationsquellen in diesen losen Verbund von Quellen ermöglicht eine netzwerkweite Nutzung der Informationen. In dieser Arbeit wird gezeigt, wie die Mechanismen der *Middleware* für das Kontextmanagementsystem genutzt werden und wie die akustische Szenenanalyse als Kontextquelle eingebunden wird.

Die Amigo *Middleware* bildet einen losen Verbund von Diensten, die dynamisch zusammengestellt und verbunden werden. Dies bedeutet jedoch, dass eine aussagekräftige und durch Maschinen verständliche Beschreibung der Dienste und Informationsquellen entwickelt werden muss, so dass eine automatische Komposition von Diensten auf semantischer Ebene erfolgen kann. Die hierzu notwendigen Beschreibungen für die Kontextquelle der akustischen Szenenanalyse werden im Rahmen dieser Arbeit vorgestellt.

3.3 Ambiente Kommunikation

Erste Formen von ambienter Kommunikation wurden durch Aki Härmä (Philips®) und Michael Stanford (Intel®) als eine Art der Kommunikation beschrieben, bei der eine *VoIP*-Verbindung einfach angelassen wurde und somit Gesprächspartner dieser beitreten oder diese verlassen, indem sie in den entsprechenden Raum eintreten oder hinausgehen [Här07]. Eine solche Form der Kommunikation kann natürlich nur zwischen nahestehenden Personen durchgeführt werden, da beide Seiten einen unkontrollierten, zufälligen Einblick in die Privatsphäre des Anderen erhalten. Betrachtet man dieses beschriebene Szenario genauer, so sind nicht alle Aspekte der ambienten Intelligenz mit einbezogen worden. In der hier vorliegenden Arbeit wird die Idee der ambienten Kommunikation unter dem Paradigma der ambienten Intelligenz untersucht, wodurch den in der Einleitung bereits beschrieben Kernelementen, wie z. B. der Orientierung auf den Benutzer, Rechnung getragen wird. Dies bedingt eine Einbindung der ambienten Kommunikation in die ambiente Intelligenz durch die Verwendung einer *Middleware*. Das physikalische Gerät zur Kommunikation, d. h. das Telefon, wird dabei durch einen personalisierten Softwaredienst ersetzt. Es erfolgt somit eine Ablösung der gerätezentrierten Kommunikation durch eine „überall“ verfügbare Möglichkeit zur Kommunikation, in deren Verlauf die Kommunikation dem Nutzer durch das Haus folgt und der Nutzer nicht mehr an einen Ort gebunden ist. Die ambiente Kommunikation verwirklicht folglich die Kernelemente der ambienten Intelligenz:

- Integration: Die Hardwarekomponenten des Systems werden unauffällig in die Umgebung integriert. Der Nutzer muss nicht mehr ein bestimmtes Gerät aufsuchen, sondern der Dienst der Kommunikation steht ihm überall zur Verfügung.

- Kontextbewusstsein: Informationen über die Umgebung, über anwesende Personen und kontextrelevante Ereignisse tragen zur Verbesserung des Kommunikationssystems bei und werden über eine entsprechende Schnittstelle verfügbar gemacht.

- Personalisierung: Die Kommunikation orientiert sich am Benutzer und wird an seine Bedürfnisse und Wünsche angepasst.

- Adaptivität & Antizipation: Das System wird auf aktuelle Ereignisse kontextabhängig reagieren und dem Nutzer so vorhersagbare oder absehbare Handlungen abnehmen. Hierbei wird zudem ein Schutz der Privatsphäre berücksichtigt.

Die Auswahl und Implementierung von Ansätzen und Verfahren orientiert sich an deren Effizienz, die gestellten Anforderungen im System zu erfüllen. Bevorzugt werden Lösungsansätze, die parallel für mehrere Problemstellungen verwendet werden können, um die Leistungsfähigkeit des Systems bei konstantem Ressourcenverbrauch zu steigern.

4 Akustische Szenenanalyse

Das Ziel der akustischen Szenenanalyse ist die Gewinnung von Informationen aus den Signalen von räumlich verteilten Mikrophonen. Die hierbei auftretenden Aufgaben können in mehrere Verarbeitungsschritte aufgeteilt werden. Zuallererst wird eine Verarbeitung der aufgenommenen Signale zum Zweck der Störgeräuschreduktion und der Berechnung von Merkmalen durchgeführt. Hierauf basierend kann im nächsten Schritt eine Lokalisation von Quellen durchgeführt werden. Anschließend kann eine Klassifikation der akustischen Ereignisse anhand einer trainierten Wissensbasis erfolgen. Im letzten Verarbeitungsschritt werden die gewonnenen Informationen zusammengeführt, bewertet und im System für Applikationen bereitgestellt.

4.1 Merkmalsextraktion

Die akustischen Signale im vernetzten Haus werden durch unterschiedliche stationäre und instationäre Störquellen beeinflusst. Somit ist eine effektive Störunterdrückung für die spätere Erkennung nötig. Grundsätzlich lassen sich hierbei zwei Ansätze verfolgen. Zum einen kann das akustische Signal gefiltert werden, um eine Reduktion der Störung zu erreichen. Zum anderen kann zunächst eine Merkmalsextraktion erfolgen und der Merkmalsvektor anschließend entstört werden. Beide Ansätze werden erfolgreich unter anderem in der automatischen Spracherkennung verwendet [ETS02, HS05].

Ein Leitgedanke bei der Entwicklung der akustischen Szenenanalyse ist die Effizienz von Komponenten und deren Wiederverwendbarkeit. Die Entstörung des Zeitsignals anstelle der Merkmale bietet in dieser Hinsicht den Vorteil, dass das entstörte Signal für eine Kommunikation verwendbar ist.

4.1.1 Störgeräuschunterdrückung

Die hier verwendete Störgeräuschunterdrückung ist entwickelt worden aus der 2-stufigen Wiener-Filterung des *Advanced Front-end Feature Extraction* (*AFE*) des *ETSI* [ETS02]. Die Anforderung war, eine Filterung des Eingangssignals durchzuführen, die sowohl gute Ergebnisse für einen menschlichen Hörer (gute Sprachqualität) als auch für eine nachfolgende Klassifikationsaufgabe (z. B. Sprechererkennung) erzielt.

Das *AFE* ist eine aus der Spracherkennung stammende Signalverarbeitungskomponente, die bei geringer Rechenkomplexität einen hohen Gewinn im Signal-zu-Rauschabstand (engl. *Signal to Noise Ratio*, *SNR*) bietet. Nachteilig für die Verwendung im Bereich Kommunikation ist die leicht reduzierte Sprachqualität bei niedrigen *SNR*-Werten. Zudem ist das

2-stufige Wiener-Filter des *AFE* nur für eine Abtastrate von $8\,\mathrm{kHz}$ spezifiziert. Eine Anpassung auf eine Abtastrate von $16\,\mathrm{kHz}$ ist durch eine Verdoppelung der Blockgrößen und der Anpassung einiger Systemparameter möglich. Die Reduktion des *SNR*-Gewinns bei niedrigen *SNR*-Werten der Eingangssignale verbessert die subjektive Qualität des Sprachsignals zu Lasten eines höheren Rauschanteils.

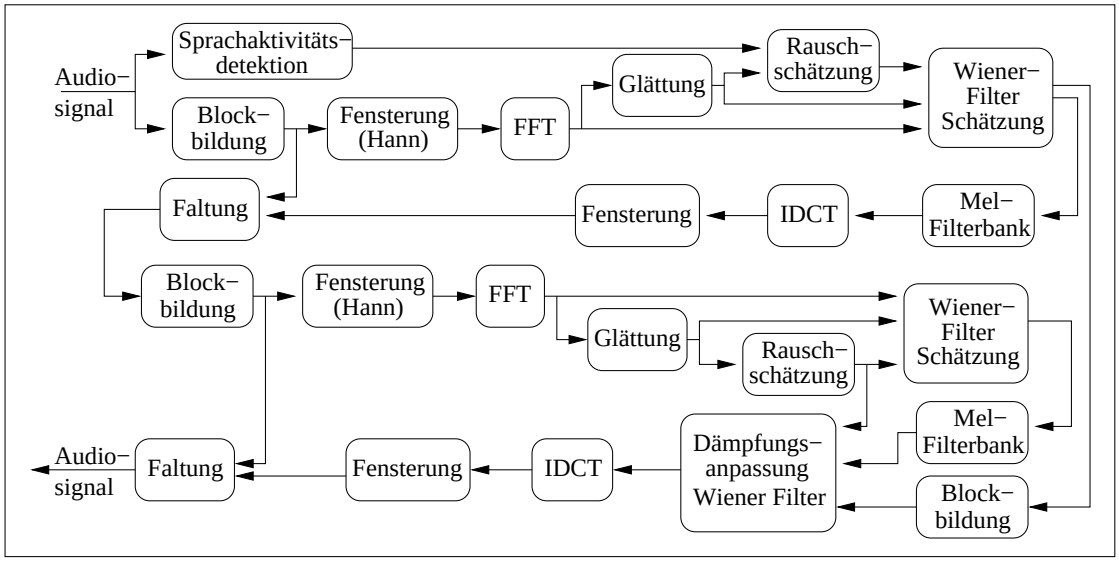


Abbildung 4.1: Blockdiagramm des 2-stufigen Wiener-Filters zur Störgeräuschreduktion

Das Blockschaltbild in Abb. 4.1 zeigt die Komponenten des 2-stufigen Wiener-Filters. Basierend auf einer Sprachaktivitätsdetektion wird auf dem Eingangssignal eine Schätzung des Störgeräuschspektrums durchgeführt. Anschließend wird ein Wiener-Filter zur Reduktion der Störgeräusche geschätzt und mit Hilfe eine Mel-Frequenz-Filterbank gehörorientiert geglättet. Die Filterung selbst wird im Zeitbereich durch den Block Faltung realisiert, da dies dem Entstehen von Störungen (sog. *musical tones*) entgegenwirkt.

Die zweite Stufe des Wiener-Filters führt auf dem Ausgangssignal der ersten Stufe eine erneute Schätzung des verbliebenen Störspektrums durch. Das hieraus berechnete Wiener-Filter wird durch eine Mel-Frequenz Filterbank geglättet und in der Dämpfungsanpassung mit dem Wiener-Filter der ersten Stufe kombiniert. Die Filterung wird erneut im Zeitbereich realisiert.

4.1.2 *Mel-Frequency Cepstral Coefficients*

Die *Mel-Frequency Cepstral Coefficients* (*MFCC*) werden aus dem entstörten Ausgangssignal der 2-stufigen Wiener-Filterung berechnet. Zunächst werden durch eine Hochpassfilterung Gleichanteile im Audiosignal sowie tieffrequente Störungen gedämpft. In einem weiteren Schritt wird in der Vorverstärkung eine Anhebung der Höhen vorgenommen. Das Signal wird dann gefenstert, anschließend in den Frequenzbereich transformiert und mit einer Mel-Frequenz Filterbank geglättet. Die Berechnung der diskreten Cosinus Transformation (DCT) liefert die cepstralen Merkmale, welche in der Nachverarbeitung mit Hilfe der logarithmierten Energie des Audiosignals normalisiert werden. Zuletzt werden näherungsweise die erste

(Delta-Merkmale) und zweite zeitliche Ableitung (Delta-Delta-Merkmale) der Merkmale berechnet und im Multiplexer zu einem Merkmalsvektor zusammengefasst. In Abb. 4.2 ist das Blockschaltbild zur Bestimmung der *MFCC* angegeben.

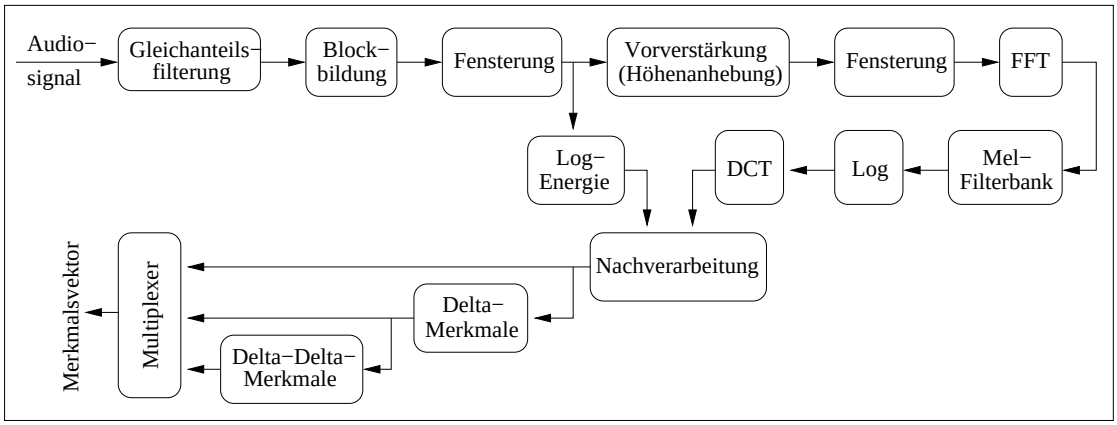


Abbildung 4.2: Blockdiagramm zur Berechnung der *Mel-Frequency Cepstral Coefficients*

4.1.3 *Maximum Autocorrelation Value*

Ein häufig in der Sprechererkennung verwendetes Merkmal ist die Stimmbandgrundfrequenz. Dieses Merkmal besitzt zum einen den Nachteil, dass es nur für stimmhafte Abschnitte der Sprache existiert. Zum anderen kann es für die Erkennung von akustischen Ereignissen, die nicht durch den menschlichen Sprachtrakt hervorgerufen werden, nicht verwendet werden.

In [WP00] wird ein alternatives Merkmal, der *Maximum Autocorrelation Value* (*MACV*), vorgeschlagen, welcher ein Maß für die Periodizität des Signals in einem betrachteten Fenster ist. Vorteil hierbei ist, dass das Merkmal auch für stimmlose Laute existiert und wie in der Literatur [WP00] gezeigt wird, dem Merkmal Stimmbandgrundfrequenz in der Erkennungsleistung überlegen ist. Dieses Merkmal kann außerdem für die akustische Ereignisdetektion verwendet werden, da es nur eine Bewertung der Periodizität des Signals vornimmt, die nicht an das Vorhandensein einer Stimmbandgrundfrequenz gebunden ist.

Zunächst wird für den *MACV* die Autokorrelationsfunktion des gefensterten Eingangssignals $\tilde{x}(n)$ der Länge N mit

$$R(k) = \frac{1}{N} \sum_{n=0}^{N-1-k} \tilde{x}(n)\tilde{x}(n+k) \qquad k = 0, \ldots, N-1 \tag{4.1}$$

berechnet. Anschließend wird die Autokorrelationsfunktion mit dem Koeffizienten $R(0)$ normiert:

$$r(k) = \frac{R(k)}{R(0)}. \tag{4.2}$$

Die Autokorrelationssequenz kann entweder in Q gleich große Blöcke unterteilt werden, so dass für jeden Block das Maximum bestimmt wird und so ein *MACV*-Merkmalsvektor der

Dimension Q entsteht, oder es wird nur ein *MACV*-Wert für den Bereich der Stimmbandgrundfrequenz ($t \in [2{,}5\,\mathrm{ms}, 12{,}5\,\mathrm{ms}] \mathrel{\widehat{=}} k \in [40, 200]$ bei einer Abtastfrequenz von $16\,\mathrm{kHz}$) berechnet.

$$\mathit{MACV}(q) = \max_{(q-1)\frac{N}{Q}<k<q\frac{N}{Q}} \{r(k)\} \qquad q = 0, \ldots, Q-1 \tag{4.3}$$

$$\mathit{MACV} = \max_{40<k<200} \{r(k)\} \tag{4.4}$$

In [ZSN05] wird eine Variation des *MACV* vorgeschlagen, bei der Anstelle von Gl. 4.1 die erwartungstreue Schätzung der Autokorrelationsfunktion

$$R(k) = \frac{1}{N-k} \sum_{n=0}^{N-1-k} \tilde{x}(n)\tilde{x}(n+k) \qquad k = 0, \ldots, N-1 \tag{4.5}$$

verwendet wird.

4.2 Akustische Positionsschätzung

Die Lokalisation von Personen oder Ereignissen anhand von akustischen Signalen setzt das Vorhandensein mehrerer räumlich getrennter Mikrophone bzw. Mikrophongruppen voraus. Hierbei werden die Unterschiede in der Signallaufzeit und das Wissen über die Position der Mikrophone verwendet, um Positionsschätzungen durchzuführen. Das am häufigsten in der Literatur beschriebene Verfahren der *Generalized Cross Correlation with Phase Transformation* (*GCC-PHAT*) nutzt die normalisierte Kreuzkorrelation zwischen zwei Mikrophonsignalen, um die Laufzeitdifferenz zu berechnen. Als Alternative hierzu wird in dieser Arbeit die Positionsbestimmung mittels adaptiver Strahlformung diskutiert.

4.2.1 *Generalized Cross Correlation with Phase Transformation*

Das in [KC76] vorgestellt *GCC-PHAT*-Verfahren berechnet mit Hilfe des normierten Kreuzleistungsdichtespektrums die Laufzeitdifferenz der Signale zwischen zwei Mikrophonen. Es wird im Weiteren angenommen, dass insgesamt $l = 1, \ldots, L$ Mikrophongruppen mit jeweils M_l Mikrophonen in einem Raum vorhanden sind. Die Laufzeitdifferenz zwischen den abgetasteten Mikrophonsignalen $x_{i,l}(n)$ und $x_{j,l}(n)$ (i-tes und j-tes Mikrophon der l-ten Mikrophongruppe) wird geschätzt als das Maximum der Fourier-Rücktransformierten der Kohärenzfunktion. Die Fourier-Rücktransformierte ist mit

$$\phi_{ij,l}^{(GCC)}(\lambda) = \mathrm{IDFT}\left\{ \frac{\mathrm{DFT}\{x_{i,l}(n)\} \cdot \mathrm{DFT}^*\{x_{j,l}(n)\}}{|\mathrm{DFT}\{x_{i,l}(n)\} \cdot \mathrm{DFT}^*\{x_{j,l}(n)\}|} \right\} \tag{4.6}$$

gegeben. Zusätzlich ist es möglich die Fourier-Rücktransformierte zu interpolieren, um eine höhere zeitliche Auflösung zu erzielen:

$$\phi_{ij,l}^{(GCC)}(\lambda) \overset{\text{Interpolation}}{\longrightarrow} C_{ij,l}^{(GCC)}(\tau) = \sum_{\lambda} \phi_{ij,l}^{(GCC)}(\lambda)\,\mathrm{si}\left(\pi \frac{\tau - \lambda T}{T}\right). \tag{4.7}$$

An dieser Stelle sei darauf hingewiesen, dass $C_{ij,l}^{(GCC)}(\tau)$ in der Implementierung ein zeitlich diskretes Signal darstellt, da die Interpolation in einem Digitalrechner durchgeführt wird. Für die Schätzung der Laufzeitdifferenz folgt somit:

$$\tau_{ij,l}^{(GCC)} = \underset{\tau}{\operatorname{argmax}} \left\{ |C_{ij,l}^{(GCC)}(\tau)| \right\}. \tag{4.8}$$

4.2.2 Akustische Strahlformung

Der Zweck der akustischen Strahlformung ist die Ausrichtung der Empfindlichkeit einer Mikrophongruppe auf eine akustische Quelle im Raum. Die Verstärkung der Quelle führt im Ausgangssignal zu einer Verbesserung des *SNR* und somit zu einer Unterdrückung möglicher Störquellen aus anderen Raumrichtungen. Im Folgenden wird das in [WH05] beschriebene Verfahren zur Strahlformung vorgestellt. Es ist ein blindes Verfahren, welches sich auf die stärkste im Raum befindliche Quelle ausrichtet. Um eine Fehlausrichtung der Strahlformung in Sprachpausen zu unterbinden, wird eine Sprachaktivitätsdetektion zur Steuerung der Adaption benötigt.

Gegeben sei eine Mikrophongruppe mit $i = 1, \ldots, M_l$ Mikrophonen. Jedes Mikrophon liefert ein Signal

$$x_i(n) = h_i(n) * s(n) + n_i(n) \tag{4.9}$$

bestehend aus einem Störsignal $n_i(n)$ und dem gewünschten Sprachsignal $s(n)$, welches mit der unbekannten Raumimpulsantwort $h_i(n)$ gefaltet wird. Die Signale $x_i(n), i = 1, \ldots, M_l$ sollen nun durch ein Filter $f_i(n)$ so gefiltert und anschließend summiert werden, dass eine konstruktive Überlagerung des Sprachsignals $s(n)$ erzielt wird:

$$y(n) = \sum_{i=1}^{M_l} f_i(-n) * x_i(n). \tag{4.10}$$

Die Filter $f_i(n)$ seien dabei Filter mit endlicher Filterimpulsantwort (engl. *Finite Impulse Response, FIR*). Eine Implementierung der Filterung im Frequenzbereich führt zu einer Reduktion des Rechenaufwandes und ist der zeitlichen Filterung vorzuziehen. Es folgt für Gl. 4.10, dass

$$Y(k) = \sum_{i=1}^{M_l} F_i^*(k) \cdot X_i(k) \qquad k = 0, \ldots, K-1 \tag{4.11}$$

ist, mit k als dem k-ten Frequenzbin der K langen diskreten Fourier Transformation (DFT). Durch die Einführung der Vektornotation

$$\boldsymbol{F}(k) = [F_1(k), \ldots, F_{M_l}(k)]^T \tag{4.12}$$

$$\boldsymbol{X}(k) = [X_1(k), \ldots, X_{M_l}(k)]^T \tag{4.13}$$

kann Gl. 4.11 mit

$$Y(k) = \boldsymbol{F}^H(k)\boldsymbol{X}(k) \qquad k = 0, \ldots, K-1 \tag{4.14}$$

dargestellt werden. Die Adaption der Filter erfolgt entsprechend [WH07] durch ein deterministisches Gradientenverfahren und liefert die Adaptionsregel

$$\boldsymbol{F}_{m+1}(k) = \boldsymbol{F}_m(k) + \mu \left(\boldsymbol{\Phi}_{xx}(k)\boldsymbol{F}_m(k) - \boldsymbol{F}_m^H(k)\boldsymbol{\Phi}_{xx}(k)\boldsymbol{F}_m(k)\boldsymbol{F}_m(k) \right) \tag{4.15}$$

mit m als Iterationsindex, μ als Schrittweite, $\boldsymbol{\Phi}_{xx}$ als spektrale Kreuzleistungsdichtematrix der Mikrophonsignale und der Nebenbedingung $\boldsymbol{F}^H(m)\boldsymbol{F}(m) = 1$. Dabei liefert die Gleichung Gl. 4.15 den Eigenvektor zum größten Eigenwert der spektralen Kreuzleistungsdichtematrix $\boldsymbol{\Phi}_{xx}$ [WH07]. Diese Verfahren der akustischen Strahlformung wird als *Filter Sum Beamformer* (*FSB*) bezeichnet [WH05].

Die Verwendung von *FIR*-Filtern im *FSB* bietet gegenüber einem *Delay Sum Beamformer* (*DSB*) den Vorteil, dass neben den direkten Schallkomponenten auch frühe Reflexionen mit berücksichtigt werden und somit die Klarheit der Sprache verbessert wird [WH05].

Ein positiver Nebeneffekt der *FSB*-Adaption ist die Möglichkeit, eine Schätzung des Einfallswinkels der akustischen Signale relativ zur Ausrichtung der Mikrophongruppe anhand der Filterimpulsantworten durchzuführen [SH06]. Hierfür wird die Kreuzkorrelation zwischen dem i-ten und j-ten Mikrophon der l-ten Mikrophongruppe mit

$$\phi_{ij,l}^{(FSB)}(\lambda) \quad = \quad f_i(-\lambda) * f_j(\lambda) \tag{4.16}$$

berechnet, wobei $\lambda = m \cdot T$ einem Vielfachen der Abtastperiode entspricht. Da die *FIR*-Filter nicht ganzzahlige Verzögerungen modellieren können, ist eine Interpolation der Kreuzkorrelation zur Steigerung der Auflösung möglich.

$$\phi_{ij,l}^{(FSB)}(\lambda) \quad \overset{\text{Interpolation}}{\longrightarrow} \quad C_{ij,l}^{(FSB)}(\tau) \tag{4.17}$$

Die Verzögerung zwischen den Signalen an den Mikrophonen kann mit

$$\tau_{ij,l}^{(FSB)} \quad = \quad \underset{\tau}{\operatorname{argmax}} \, |C_{ij,l}^{(FSB)}(\tau)| \tag{4.18}$$

bestimmt werden. Analog zur Latenzschätzung des *GCC-PHAT* kann die Kreuzkorrelation der *FIR*-Filter als Fourier-Rücktransformierte der Kohärenzfunktion der Mikrophonsignale angesehen werden.

4.2.3 Lokalisation mittels verteilter Mikrophongruppen

Der Einfallswinkel kann grundsätzlich als Information über eine Position im vernetzten Haus verwendet werden, jedoch steigert die Kombination verteilter Mikrophongruppen zur Schätzung einer Position in kartesischen Koordinaten den Informationsgehalt beträchtlich. Hierzu ist es notwendig, die Position und Anordnung der Mikrophongruppen im Raum zu kennen. Im Folgenden wird das aus der Literatur bekannte Verfahren der Kohärenzfeldanalyse [OSBC06] einer Schnittpunktanalyse gegenübergestellt und hinsichtlich Genauigkeit und Rechenaufwand verglichen.

Kohärenzfeldanalyse

Das verbreitetste Verfahren zur akustischen Positionsbestimmung ist die Kohärenzfeldanalyse (engl. *Global Coherence Field analysis*) [OSBC06], welche äquivalent zum „*Steered*

Response Power“-Verfahren ist [DBA07] und mit dem Begriff „*GCF*-Analyse“ abgekürzt wird. Hierbei wird die Positionsbestimmung im Raum zumeist in zwei Dimensionen durchgeführt, so dass die möglichen Positionen in einer Fläche liegen. Über diese Fläche wird ein Gitter $\boldsymbol{G}$ gelegt, welches durch die diskreten Gitterpunkte $[x, y] \in \boldsymbol{G}$ definiert ist. Zu jedem Zeitschritt wird die globale Kohärenzfunktion für alle Gitterpunkte $[x, y]$ des Raumes mit

$$GCF(x, y) \quad = \quad \frac{1}{L} \sum_{l=1}^{L} \frac{2}{M_l^2 - M_l} \sum_{i=1}^{M_l - 1} \sum_{j=i+1}^{M_l} C_{ij,l} \left(\tau_{ij,l}(x, y) \right) \tag{4.19}$$

berechnet. Hierbei werden die interpolierten Fourier-Rücktransformierten der Kohärenzfunktionen $C_{ij,l}(\tau)$ der $l = 1, \ldots, L$ Mikrophongruppen verwendet, welche entweder mit dem *GCC-PHAT*-Verfahren oder der akustischen Strahlformung geschätzt wurden. Die Laufzeitdifferenz $\tau_{ij,l}(x, y)$ wird berechnet aus der relativen Position und Orientierung der l-ten Mikrophongruppe zum Aufpunkt $[x, y]$ im Raum. Da der Aufwand der Aufpunktsberechnung sich quadratisch zur Quantisierung des Raumes verhält, muss eine Abwägung zwischen dem geduldeten Quantisierungsfehler und der vertretbaren Rechenkomplexität vorgenommen werden.

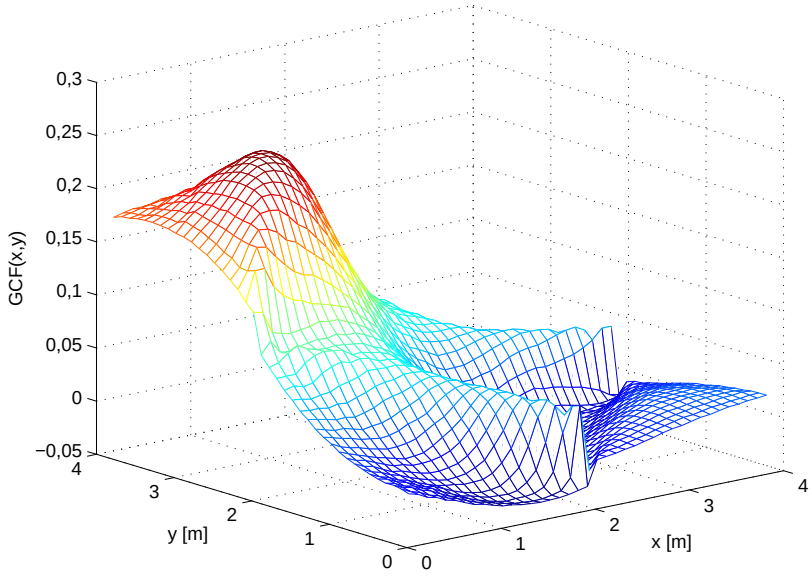


Abbildung 4.3: Beispiel eine *GCF*-Analyse für vier Mikrophongruppen zur akustischen Positionsschätzung durch verteilte Mikrophongruppen

Die Abb. 4.3 zeigt ein Beispiel für eine *GCF*-Analyse für einen Raum der Größe $4\,\mathrm{m} \times 4\,\mathrm{m}$, in dem vier Mikrophongruppen ($\boldsymbol{r}_l = [0, 2]; [4, 2]; [2, 0]; [2, 4]$) jeweils mittig an den Wänden angebracht sind. Das Maximum der globalen Kohärenzfunktion wird als Hypothese für die Sprecherposition verwendet.

Schnittpunktanalyse

Die Schnittpunktanalyse ist ein vereinfachtes Verfahren zur Berechnung einer Sprecherposition, basierend auf den interpolierten Fourier-Rücktransformierten der Kohärenzfunktionen

und dem Wissen über die Position und Anordnung der Mikrophongruppen. Es wird dabei angenommen, dass jede der L Mikrophongruppen eine lineare Anordnung besitzt, so dass die Einfallswinkel $\alpha_{ij,l}$ der akustischen Signale durch

$$\alpha_{ij,l} = \arcsin\left(c \cdot T \cdot \frac{\tau_{ij,l}}{s_{ij,l}}\right) \tag{4.20}$$

berechnet werden können. Dabei ist c die Schallgeschwindigkeit in der Luft, T die Abtastperiode und $s_{ij,l}$ der Abstand zwischen dem i-ten und j-ten Mikrophon der l-ten Mikrophongruppe. Stehen mehr als zwei Mikrophone in einer Gruppe ($M_l > 2$) zur Verfügung, kann eine Mittelung über alle Kombinationen der Mikrophone mit

$$\overline{\alpha}_l = \frac{2}{M_l^2 - M_l} \sum_{i=1}^{M_l-1} \sum_{j=i+1}^{M_l} \alpha_{ij,l} \tag{4.21}$$

erfolgen, falls die räumliche Ausdehnung der Mikrophongruppe nicht zu einer Verletzung der Fernfeldnäherung führt. Die Fernfeldnäherung ist die Annahme, dass das akustische Signal in einer ebenen Wellenfront auf die Mikrophone trifft. Die Laufzeitdifferenz $\tau_{ij,l}$ zwischen den Mikrophonsignalen kann sowohl durch das *GCC-PHAT*-Verfahren ($\tau_{ij,l}^{(GCC)}$) als auch durch den *FSB*-Ansatz ($\tau_{ij,l}^{(FSB)}$) bestimmt werden.

Jede Winkelschätzung $\overline{\alpha}_l$ einer Mikrophongruppe mit der Position $\boldsymbol{r}_l = [x_l, y_l]^T$ wird als Geradengleichung

$$\boldsymbol{g}_l(\nu) = \boldsymbol{r}_l + \nu \cdot \boldsymbol{a}_l(\overline{\alpha}_l, \beta_l) \tag{4.22}$$

dargestellt. Der Richtungsvektor $\boldsymbol{a}_l$ ist dabei abhängig von dem geschätzten Einfallswinkel $\overline{\alpha}_l$ und der Orientierung der Mikrophongruppe im gewählten Koordinatensystem β_l (Winkel zur Ordinate).

In Abb. 4.4 ist ein Beispiel für die Positionsbestimmung durch die Schnittpunktanalyse gegeben. Die Kombination der Geradengleichungen der i-ten und j-ten Mikrophongruppe liefert im Idealfall einen Schnittpunkt $\boldsymbol{\chi}_{ij}$ im Raum, der als Grundlage für die Positionsschätzung verwendet werden kann. Sollte ein Schnittpunkt durch Fehler bei der Schätzung der Winkel außerhalb des Raumes liegen, so wird diese Schätzung verworfen. Die Position $\boldsymbol{P} = [x_p, y_p]^T$ der akustischen Quelle wird als Schwerpunkt aller Schnittpunkte $\boldsymbol{\chi}_{ij}$ mit

$$\boldsymbol{P} = \frac{2}{L^2 - L} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \boldsymbol{\chi}_{ij} \tag{4.23}$$

berechnet.

In Laborversuchen wurde beobachtet, dass die Gewichtung der Schnittpunkte mit einem aus der Kreuzkorrelation berechneten Konfidenzwert die Schätzung positiv beeinflusst. Dies ist auch in den Simulationen in Abb. 4.6 (a) erkennbar. Der Gewichtsfaktor berechnet sich mit

$$\gamma_{ij,l} = \frac{\max\{|\phi_{ij,l}(\lambda)|\}}{\sum\limits_{\lambda'} |\phi_{ij,l}(\lambda')|}, \tag{4.24}$$

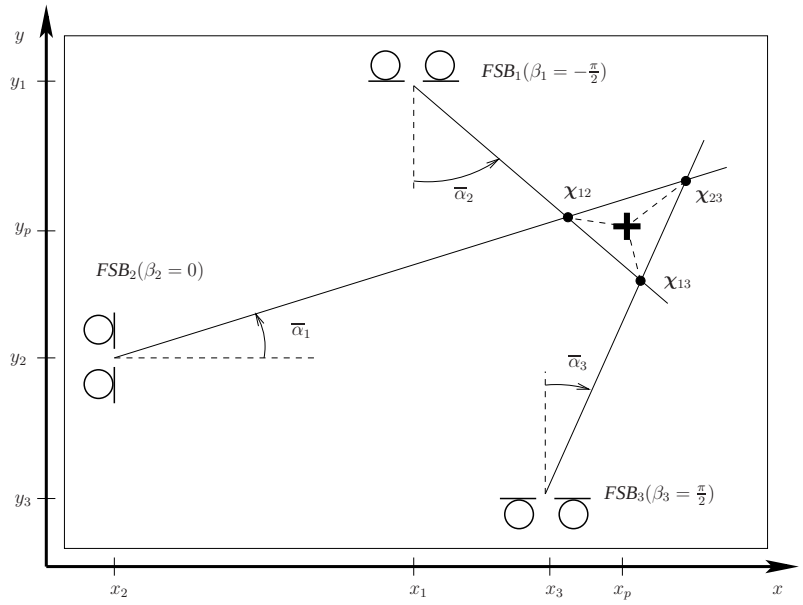


Abbildung 4.4: Beispiel einer akustischen Positionsschätzung mit drei Mikrophongruppen durch die Schnittpunktanalyse

und ist ein Maß für die Impulsförmigkeit der Kreuzkorrelationsfunktion.

Ein Überblick über Verfahren zur Positionsschätzung kann in [WM09] gefunden werden. Unter anderem wird dort auf den *Linear Intersection Estimator* eingegangen, der im dreidimensionalen Raum den minimalen Abstand zwischen zwei Geraden als Positionsschätzung verwendet und als verallgemeinerte Form der Schnittpunktanalyse für drei Dimensionen angesehen werden kann.

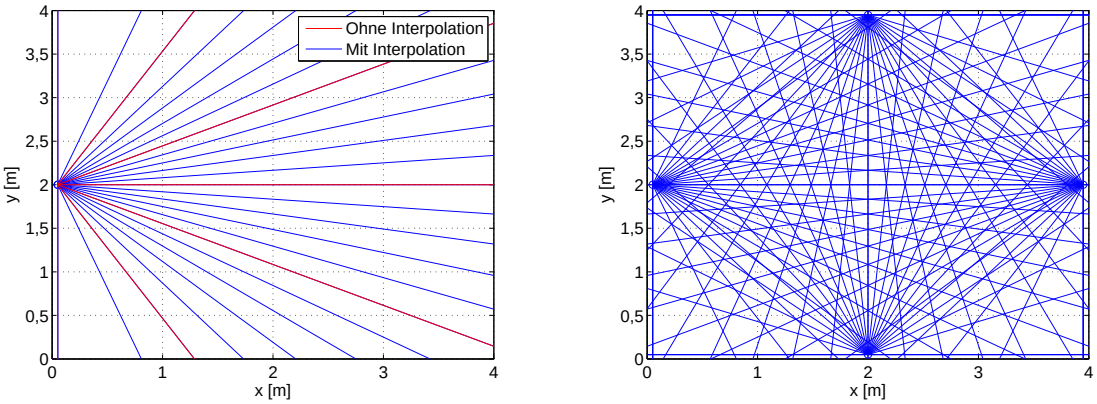
Interpolation

Der Abstand der Mikrophone innerhalb einer Mikrophongruppe hat zum einen Einfluss auf die maximal ohne Aliasingfehler auflösbaren Frequenzen und zum anderen einen Einfluss auf die Anzahl der unterscheidbaren Laufzeitdifferenzen. Je kleiner der Abstand zwischen den Mikrophonen gewählt wird, desto weniger räumliche Aliasingfehler treten auf und desto geringer ist die Anzahl der ohne Interpolation unterscheidbaren Laufzeitdifferenzen.

Die Anzahl der Laufzeitdifferenzen wird bestimmt durch die Abtastperiode der Fourier-Rücktransformierten der Kohärenzfunktion und die gewählte Interpolation (vgl. Gl. 4.7). Ohne Interpolation sind nur ganzzahlige Vielfache der Abtastperiode als Laufzeitdifferenz messbar. Mit Interpolation vervielfacht sich die Anzahl der unterscheidbaren Laufzeitdifferenzen um den Interpolationsfaktor. In beiden Fällen kann nur eine begrenzte Menge an Laufzeitdifferenzen unterschieden werden.

In Abb. 4.5 (a) sind die resultierenden Winkel aus den Latenzschätzungen in rot eingezeichnet. Bei einer angenommenen Abtastrate von $1/T = 16\,\mathrm{kHz}$ und einem Mikrophonabstand von $s_{ij,l} = 0{,}05\,\mathrm{m}$ ergibt sich nach Gl. 4.20 eine maximal messbare Latenz zwischen den Signalen für einen Winkel $\alpha_{ij,l} = \pm\pi/2$ von

$$\lambda_{ij,l}^{(\max)} = \frac{s_{ij,l}}{c \cdot T} = \frac{16\,000\,\frac{1}{\mathrm{s}} \cdot 0{,}05\,\mathrm{m}}{343\,\frac{\mathrm{m}}{\mathrm{s}}} = \lceil 2{,}33 \rceil = 3 \tag{4.25}$$



(a) Auswirkung der Interpolation auf die Winkelauflösung

(b) Räumliche Verteilung der Schnittpunkte bei vier Mikrophongruppen und Interpolation

Abbildung 4.5: Positionsschätzung durch Interpolation von Winkelschätzungen

Abtastwerten. Die maximale ohne Aliasingfehler auflösbare Frequenz kann mit

$$f_{\max} = \frac{c}{s_{ij,l}} = \frac{343\,\frac{\mathrm{m}}{\mathrm{s}}}{0{,}05\,\mathrm{m}} = 6860\,\mathrm{Hz} \tag{4.26}$$

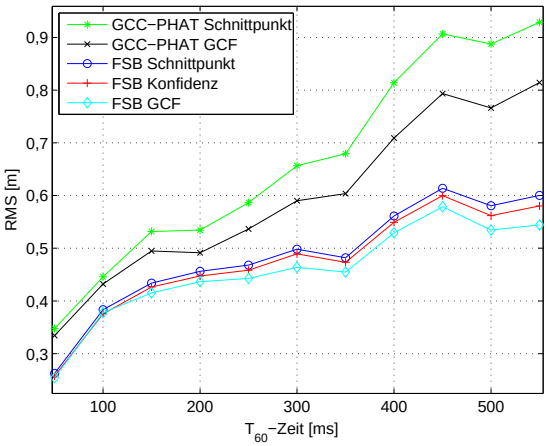
berechnet werden. Da ohne Interpolation nur ganzzahlige Verzögerungen messbar sind, können nur 7 Winkel pro Mikrophongruppe unterschieden werden (vgl. Abb. 4.5 (a), rote Linien). Erst die Interpolation erreicht eine verwertbare Winkelauflösung des Raumes (vgl. Abb. 4.5 (a), rote und blaue Linien). Die Abb. 4.5 (b) zeigt die entstehenden Schnittpunkte für einen Aufbau mit vier Mikrophongruppen und Interpolation. Es ist erkennbar, dass gerade die Ecken gegenüber der Mitte des Raumes eine schlechtere Auflösung besitzen, da dort weniger Schnittpunkte liegen. Auf Grund dieser Beobachtung ist es erforderlich, Systeme zur akustischen Lokalisation so aufzubauen, dass der Bereich mit den meisten Schnittpunkten im vorgesehenen Interaktionsbereich mit den Benutzern liegt.

Experimente

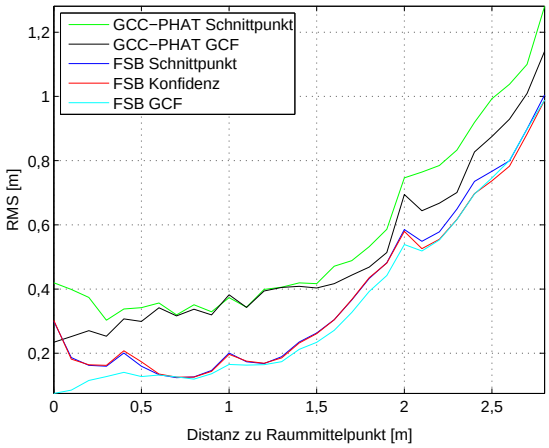
Die folgenden Experimente untersuchen und vergleichen das *GCC-PHAT*-Verfahren mit dem *FSB*-Ansatz zur Positionsschätzung hinsichtlich der Vor- und Nachteile für die Verwendung in der akustischen Szenenanalyse.

In Abb. 4.6 sind die experimentellen Ergebnisse zum Vergleich der Positionsschätzung zwischen *GCC-PHAT* und *FSB* angegeben. Hierzu wurde ein Raum der Größe $4\,\mathrm{m} \times 4\,\mathrm{m}$, mit einer Deckenhöhe von $3\,\mathrm{m}$ und unterschiedlichen Raumnachhallzeiten mit der Spiegelmethode nach [AB79] simuliert. Bei einer Abtastrate von $16\,\mathrm{kHz}$ wurde für jede Nachhallzeit eine $90\,\mathrm{s}$ lange Audiodatei für einen sich zufällig bewegenden Sprecher künstlich verhallt. Insgesamt 8 Mikrophone waren paarweise mittig an den Wänden und im Abstand von $0{,}05\,\mathrm{m}$ zueinander angebracht. Die *FFT*-Länge des *GCC-PHAT*-Verfahrens betrug 2048 Abtastwerte, mit einer anschließenden Interpolation zur Verbesserung der Positionsschätzung. Der *FSB* wurde mit einer Filterlänge von 128 Werten implementiert und das Ergebnis der Filterkorrelation ebenfalls interpoliert.

Abbildung 4.6 (a) zeigt die Wurzel des mittleren quadratischen Fehlers (engl. *Root Mean Square*, *RMS*) für die Positionsschätzung durch den *GCC-PHAT* („*GCC-PHAT* Schnitt-



(a) Vergleich des *RMS* bezogen auf die Nachhallzeit des Raumes



(b) Vergleich des *RMS* bezogen auf die Distanz der Sprecherposition zum Raummittelpunkt

Abbildung 4.6: Experimente zur Positionsschätzung mit dem *FSB*- und dem *GCC-PHAT*-Verfahren

punkt“), den *FSB* ohne Gewichtung der Schnittpunkte („*FSB* Schnittpunkt“) und den *FSB* mit Gewichtung der Schnittpunkte proportional zum Konfidenzwert der Schätzungen („*FSB* Konfidenz“) für ansteigende Nachhallzeiten des Raumes. Des Weiteren sind die *RMS*-Werte für die Positionsschätzung bei Verwendung der Kohärenzfeldanalyse für das *GCC-PHAT*-Verfahren („*GCC-PHAT GCF*“) und den *FSB*-Ansatz („*FSB GCF*“) angegeben.

Die experimentellen Ergebnisse zeigen, dass der *FSB* eine bessere Positionsschätzung ermöglicht als das *GCC-PHAT*-Verfahren. Des Weiteren besitzen die Ausgangssignale des *FSB* ein besseres *SNR* und könnten somit für weitere Verarbeitungsschritte besser geeignet sein als ein einzelnes Mikrophonsignal. Vergleicht man die Ergebnisse der Positionsschätzung des *GCC-PHAT*-Verfahrens mit *GCF*-Analyse („*GCC-PHAT GCF*“) mit denen der einfacheren Schnittpunktanalyse („*GCC-PHAT* Schnittpunkt“), so kann festgestellt werden, dass das *GCC-PHAT*-Verfahren deutlich von der *GCF*-Analyse profitiert. Speziell für längere Raumnachhallzeiten ist die Verwendung der *GCF*-Analyse vorteilhaft, um den Fehler der Positionsschätzung gering zu halten. Im Falle der Positionsschätzung durch den *FSB* ist der Vorteil der *GCF*-Analyse („*FSB GCF*“) gegenüber der Schnittpunktanalyse („*FSB* Schnittpunkt“) weniger ausgeprägt und es kann zu Gunsten einer reduzierten Rechenanforderung darauf verzichtet werden.

Abbildung 4.6 (b) zeigt die Untersuchungen zur Verteilung der Fehler bezogen auf den Abstand der Sprecherposition zum Mittelpunkt des Raumes. Der Fehler steigt mit zunehmender Distanz zum Mittelpunkt des Raumes an und ist am größten in den Ecken, wie es bereits in experimentellen Versuchen im Labor beobachtet wurde. Dies zeigt, dass die Platzierung der Mikrophone die erreichbare Schätzgenauigkeit beeinflusst. Mikrophongruppen sollten immer so angebracht werden, dass sie den Interaktionsbereich des Nutzers gut abdecken und die Gebiete mit großen Fehlern abseits der Nutzungsflächen liegen.

In [WPH04] wird gezeigt, dass die akustische Positionsschätzung durch eine modellbasierte Nachfilterung, wie z. B. Kalman- oder Partikelfilter, verbessert werden kann. Auf eine modellbasierte Nachfilterung wird im Rahmen dieser Arbeit bewusst verzichtet, da die experimentell erreichten Genauigkeiten in realen Umgebungen den Anforderungen genügen und somit eine rechenintensive Filterung unnötig ist. Eine erzielte Genauigkeit von ca.

$0{,}2\,\mathrm{m} - 0{,}5\,\mathrm{m}$ kann als hinreichend für die häusliche Umgebung mit geringem Nachhall (niedrige T_{60}-Zeiten) betrachtet werden.

Der Vergleich der benötigten Rechenzeit in Tab. 4.1 zeigt deutlich den Vorteil der Verwendung des *FSB* gegenüber dem *GCC-PHAT*-Verfahren.[1] Die Positionsbestimmung des *FSB*

Modul	Zeit (μs)
FSB-Strahlformung (2 Mikrophone)	273
FSB-Winkelschätzung (2 Mikrophone)	16
GCC-PHAT (2 Mikrophone)	653
Schnittpunktanalyse (4 Gruppen je 2 Mikrophone)	5
GCF-Analyse (4 Gruppen je 2 Mikrophone, $0{,}1\,\mathrm{m}$ Rasterung)	1457
GCF-Analyse (4 Gruppen je 2 Mikrophone, $0{,}05\,\mathrm{m}$ Rasterung)	5624
FSB mit Schnittpunktanalyse (4 Gruppen je 2 Mikrophone)	1161
FSB mit *GCF*-Analyse (4 Gruppen je 2 Mikrophone, $0{,}1\,\mathrm{m}$ Raster)	2613
FSB mit *GCF*-Analyse (4 Gruppen je 2 Mikrophone, $0{,}05\,\mathrm{m}$ Raster)	6780
GCC-PHAT mit Schnittpunktanalyse (4 Gruppen je 2 Mikrophone)	2617
GCC-PHAT mit *GCF*-Analyse (4 Gruppen je 2 Mikrophone, $0{,}1\,\mathrm{m}$ Raster)	4069
GCC-PHAT mit *GCF*-Analyse (4 Gruppen je 2 Mikrophone, $0{,}05\,\mathrm{m}$ Raster)	8236

Tabelle 4.1: Vergleich der Rechenzeit unterschiedlicher Module zur Positionsschätzung

mittels Schnittpunktanalyse benötigt im Vergleich zur Positionsschätzung des *GCC-PHAT* mit Schnittpunktanalyse nur $44{,}1\,\%$ der Rechenleistung. Noch größer wird der Unterschied, falls die *GCF*-Analyse angewendet wird, da die Schnittpunktberechnung um einen Faktor 1125 schneller ist. Die Experimente zeigen, dass im Falle des *FSB* die Schnittpunktanalyse der *GCF*-Analyse im Bereich Ressourcenbedarf überlegen ist, jedoch die Genauigkeit nur geringfügig niedriger liegt. In der Literatur gibt es Ansätze, den Bedarf an Rechenzeit durch die *GCF*-Analyse zu reduzieren [DBA07], welche hier jedoch nicht weiter betrachtet werden.

4.3 Segmentierung und Sprecheridentifikation

Bei der sequentiellen Vorgehensweise zur Sprecheridentifikation wird zunächst eine Einteilung des Datenstroms in homogene Abschnitte durchgeführt. Diese Abschnitte werden dann durch eine Sprecheridentifikation einem bekannten Sprecher aus der Datenbasis zugeordnet. Demgegenüber steht eine gemeinsame Segmentierung und Sprecheridentifikation, die in dieser Arbeit vorgeschlagen wird. Eine zeitnahe gemeinsame Identifikation von Sprechern in fortlaufenden Datenströmen erfordert Algorithmen, welche eine Segmentierung der Daten in homogene Abschnitte eines Sprechers und eine Klassifikation dieser Segmente mit möglichst geringer Latenz vornehmen.

Zunächst wird die Segmentierung von Daten durch die Anwendung des Bayes'schen Informationskriteriums erläutert und mögliche Ansätze zur Verwendung der Positionsinformationen zur Segmentierung diskutiert. Anschließend werden die Sprecheridentifikation für homogene Sprachsegmente und das Modelltraining vorgestellt. Abschließend werden in Experimenten die Teilkomponenten der Segmentierung und Sprecheridentifikation, sowie das Gesamtsystem getestet.

[1] Simulationsumgebung: Intel T2400@1,83 GHz, 2 GB RAM

4.3.1 Sequentielle Sprecherwechseldetektion und Identifikation

Segmentierung durch Sprecherwechseldetektion

Das Ziel der Segmentierung ist die Einteilung der Daten in homogene Abschnitte, innerhalb derer nur ein Sprecher aktiv ist. Diese Aufgabenstellung wird in der Literatur häufig als Modellselektionsproblem formuliert [DW00, WH06]. Basierend auf den N_w Merkmalsvektoren $\boldsymbol{X}_{1:N_w} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(N_w)]$ in einem betrachteten Fenster werden die folgenden zwei Hypothesen verglichen:

- H_0: Alle Merkmalsvektoren sind eine unabhängige und identisch verteilte Stichprobe der multivariaten Normalverteilung $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, welche den Sprecher beschreibt.

- H_1: Die ersten $N_w/2$ Merkmalsvektoren sind eine unabhängige und identisch verteilte Stichprobe der multivariaten Normalverteilung $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ des Sprechers A und die übrigen eine Stichprobe der multivariaten Normalverteilung $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ des Sprechers B.

Die Modellparameter $\boldsymbol{\Theta}_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, $i = 1, 2$, der Normalverteilungen bestehen aus den Mittelwertvektoren $\boldsymbol{\mu}_i$ und den Kovarianzmatrizen $\boldsymbol{\Sigma}_i$ und sind zunächst unbekannt. Sie werden durch einen *„Maximum Likelihood“*-Schätzer aus den Merkmalsvektoren innerhalb des Fensters bestimmt. Die Bewertung der zwei Hypothesen entsprechend der Definition für *BIC* aus [DW00, NK05] liefert

$$\begin{aligned} \mathit{BIC}(H_i) &= \log p(\boldsymbol{X}_{1:N_w}|H_i) - \xi\frac{m_i}{2}\log N_w && (4.27)\\ &= \sum_{k=1}^{N_w} \log p(\boldsymbol{x}(k)|H_i) - \xi\frac{m_i}{2}\log N_w, && (4.28) \end{aligned}$$

mit $p(\boldsymbol{X}_{1:N_w}|H_i)$ als *Likelihood* [2] der D-dimensionalen Merkmalsvektoren $\boldsymbol{X}_{1:N_w}$ für das parametrische Modell der Hypothese H_i, m_i als Anzahl der Parameter im Modell und N_w als Anzahl der Merkmalsvektoren. Unter der Annahme multivariater Normalverteilungen gilt

$$p\left(\boldsymbol{x}(k)|H_0\right) = \mathcal{N}\left(\boldsymbol{x}(k); \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right) \tag{4.29}$$

für die Dichtefunktion der Hypothese H_0 und

$$p\left(\boldsymbol{x}(k)|H_1\right) = \begin{cases} \mathcal{N}\left(\boldsymbol{x}(k); \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right) & \text{für} \quad k \leq N_w/2 \\ \mathcal{N}\left(\boldsymbol{x}(k); \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right) & \text{für} \quad k > N_w/2 \end{cases} \tag{4.30}$$

[2] An dieser Stelle wird bewusst der englische Begriff *„Likelihood“* verwendet, um zu verdeutlichen, dass die Auswertung der Dichtefunktion für die beobachteten Merkmalsvektoren und somit ein Zahlenwert und nicht die Dichtefunktion betrachtet wird. Eine mögliche Übersetzung mit „Mutmaßlichkeit“, wie in [Hän01] vorgeschlagen, wird zu Gunsten des häufig auch in deutschen Veröffentlichungen verwendeten Begriffs *„Likelihood“* verworfen.

für die Dichtefunktion der Hypothese H_1. Die *Likelihood* der Hypothese H_0 ist unter der Annahme, dass $\boldsymbol{X}_{1:N_w}$ eine unabhängige und identisch verteilte Stichprobe ist, gegeben durch

$$p(\boldsymbol{X}_{1:N_w}|H_0) = \prod_{k=1}^{N_w} \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_0|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)\right)} \tag{4.31}$$

$$= (2\pi)^{-\frac{N_w D}{2}}|\boldsymbol{\Sigma}_0|^{-\frac{N_w}{2}} e^{\left(-\frac{1}{2}\sum\limits_{k=1}^{N_w}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)\right)} \tag{4.32}$$

mit den *ML*-Schätzwerten der Parameter $\boldsymbol{\Theta}_0 = (\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$:

$$\boldsymbol{\mu}_0 = \frac{1}{N_w}\sum_{k=1}^{N_w}\boldsymbol{x}(k) \tag{4.33}$$

$$\boldsymbol{\Sigma}_0 = \frac{1}{N_w}\sum_{k=1}^{N_w}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T. \tag{4.34}$$

Durch Logarithmieren der Dichtefunktion und Verwendung von Gl. 4.33 und Gl. 4.34 folgt entsprechend [WH06] (vgl. Kap. A.1) für die *Likelihood* der Hypothese H_0

$$\log\left(p(\boldsymbol{X}_{1:N_w}|H_0)\right) = -\frac{N_w}{2}\log\left(|\boldsymbol{\Sigma}_0|\right) - \frac{N_w D}{2}\left(1+\log\left(2\pi\right)\right) \tag{4.35}$$

bzw. für die *Likelihood* der Hypothese H_1

$$\log\left(p(\boldsymbol{X}_{1:N_w}|H_1)\right) = -\frac{N_w}{4}\log\left(|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|\right) - \frac{N_w D}{2}\left(1+\log\left(2\pi\right)\right). \tag{4.36}$$

Die Differenz ΔBIC der *BIC*-Werte der Hypothesen wird als Kriterium für Segmentierungspunkt verwendet und kann entsprechend [CW03] als *Generalized Likelihood Ratio* der Hypothesen interpretiert werden.

$$\Delta BIC = BIC(H_1) - BIC(H_0) \tag{4.37}$$

$$= \frac{N_w}{2}\log(|\boldsymbol{\Sigma}_0|) - \frac{N_w}{4}\log(|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|) - \xi\frac{m_0}{4}\log N_w. \tag{4.38}$$

Ein ΔBIC-Wert größer Null zeigt hierbei einen Segmentierungspunkt an, wobei die Empfindlichkeit durch die Konstante ξ eingestellt wird.

Im Folgenden wird der ΔBIC-Wert der Gl. 4.37 um einen Zeitindex k erweitert ($\Delta BIC(k)$), welcher die Mitte des betrachteten Fensters der Länge N_w angibt. Dieses Fenster wird über den Datenstrom der Merkmalsvektoren geschoben, so dass der Wert $\Delta BIC(k)$ zu den Merkmalsvektoren $\boldsymbol{x}(k-N_w/2+1), \ldots, \boldsymbol{x}(k+N_w/2)$ gehört. Daraus resultiert eine Verzögerung der Information über einen Sprecherwechsel aus den ΔBIC-Werten von einer halben Fensterlänge ($N_w/2$).

Experimente unter variierenden Bedingungen, wie z. B. Hintergrundgeräuschen, zeigten die Notwendigkeit, den Parameter ξ aus Gl. 4.38 an die akustischen Umgebungsbedingungen anzupassen. Dieser Nachteil ist in der Literatur bekannt und kann durch eine metrische Entscheidungsregel abgemildert werden. Die Grundidee der metrischen Entscheidungsregel beruht auf der Beobachtung, dass ein Segmentierungspunkt im Zeitverlauf der ΔBIC-Werte

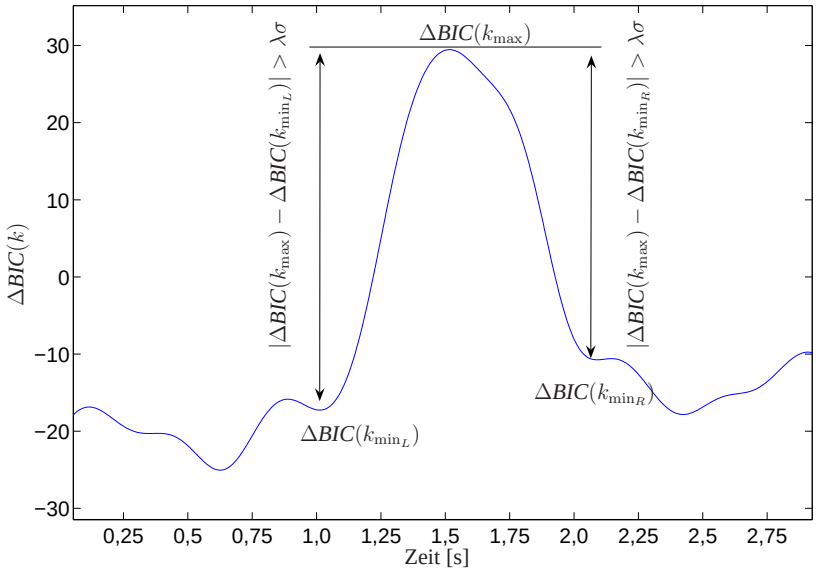


Abbildung 4.7: Metrische Entscheidungsregel zur Segmentierung durch ΔBIC-Werte

durch ein lokales Maximum gekennzeichnet ist (vgl. Abb. 4.7). Ein Segmentierungspunkt wird immer dann angenommen, falls die Differenz zwischen lokalem Minimum und Maximum ein λ-faches der Standardabweichung σ des ΔBIC-Wertes beträgt [DW00, DY08]. Die metrische Entscheidungsregel zeigt folglich einen Segmentierungspunkt an, falls mindestens eine der Bedingungen erfüllt ist:

$$|\Delta BIC(k_{\max}) - \Delta BIC(k_{\min_R})| > \lambda\sigma \tag{4.39}$$

$$|\Delta BIC(k_{\max}) - \Delta BIC(k_{\min_L})| > \lambda\sigma. \tag{4.40}$$

Dabei sei $k_{\max}$ ein Zeitpunkt, an dem ein lokales Maximum im Zeitverlauf der ΔBIC-Werte vorliegt, und $k_{\min_R}$ bzw. $k_{\min_L}$ die zugehörigen Zeitpunkte der lokalen Minima, welche rechts bzw. links vom Maximum liegen (vgl. Abb. 4.7).

Segmentierung mittels Positionsinformationen

Ein Sprecherwechsel geht immer mit einem Wechseln in der geschätzten Sprecherposition einher. Umgekehrt ist eine Veränderung der Sprecherposition jedoch kein sicherer Indikator für einen Sprecherwechsel, da der Sprecher auch nur seine Position geändert haben kann.

In Abb. 4.8 sind die Winkelschätzungen während eines Gesprächs zwischen zwei Personen und die zugehörigen Segmentierungspunkte, d. h. die Zeitpunkte der Sprecherwechsel, dargestellt. Theoretisch kann ein solches Gespräch einzig durch die Positionsinformationen segmentiert werden, weil die Sprecher räumlich gut getrennt und jeweils an einer festen Position sind. Eine solche Voraussetzung ist in einer Hausumgebung nicht gegeben, da sich die Sprecher frei bewegen können. Folglich müssen andere Ansätze zur Verwendung der geschätzten Sprecherposition betrachtet werden.

Ein möglicher Ansatz ist, dass die Position eines Sprechers für die Dauer einer Äußerung als näherungsweise konstant und die Gesprächspartner als räumlich unterscheidbar angenommen werden. Obwohl diese Annahmen in einem Gespräch üblicherweise gegeben sind,

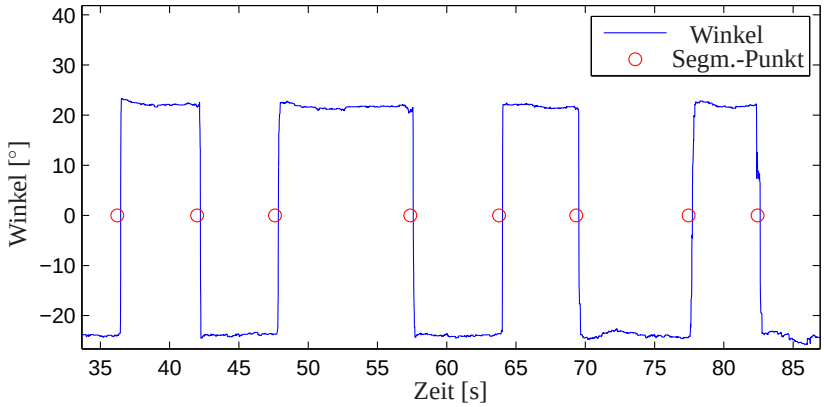


Abbildung 4.8: Vergleich zwischen Positionsinformationen und bekannten Segmentierungspunkten

stellen sie eine Einschränkung der Verwendbarkeit des Systems dar. Zunächst werden Hypothesen für Segmentierungspunkte durch das in Kap. 4.3.1 vorgestellte ΔBIC-Verfahren ermittelt und anschließend anhand der Positionsinformation nachgefiltert. Falls die Position innerhalb eines Zeitfensters konstant ist, so werden Hypothesen für einen Sprecherwechsel innerhalb dieses Zeitfensters verworfen. Hierdurch kann eine erhebliche Reduktion der Fehler erzielt werden, wie die Experimente in Kap. 4.3.3 zeigen.

In Kap. 4.3.2 wird ein alternativer Ansatz ohne die einschränkenden Annahmen vorgestellt, welcher eine kombinierte Segmentierung und Identifikation mit Hilfe der Positionsinformationen durchführt. Da dieser Ansatz Informationen aus dem Modul zur Sprecheridentifikation benötigt, wird im folgenden Kapitel zunächst die Sprecheridentifikation erläutert.

Sprecheridentifikation

Die Problemstellung der Sprecheridentifikation wird allgemein als ein Mustererkennungsproblem formuliert, bei dem eine beobachtete Menge von Merkmalsvektoren einem Sprechermodell zugeordnet werden soll [Cam97]. Dabei wird für jeden der $\mathcal{I}$ Nutzer ein stochastisches Modell aus Trainingsdaten geschätzt. Für den Klassifikationsschritt werden die *Likelihoods* der Merkmalsvektoren für die Dichtefunktionen der Sprechermodelle berechnet und anhand eines Hypothesentests verglichen. Im Folgenden werden die zum Aufbau einer Sprecheridentifikation benötigten Ansätze und Gleichungen entsprechend den Ideen aus [Cam97] und [RQD00] eingeführt, um deren Zusammenhang zur Sprecherprotokollierung herzustellen.

Die *Likelihood* der Merkmalsvektorfolge $\boldsymbol{X}_{1:N} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(N)]$, gegeben das i-te Sprechermodell ($\Omega = i$), ist unter der Annahme unabhängiger und identisch verteilter Merkmalsvektoren durch

$$p(\boldsymbol{X}_{1:N}|\Omega = i) \quad = \quad \prod_{k=1}^{N} p(\boldsymbol{x}(k)|\Omega = i) \tag{4.41}$$

gegeben. Diese *Likelihood* wird auf die *Likelihood* $p(\boldsymbol{X}_{1:N}|\Omega \neq i)$ normiert, dass die Merkmalsvektoren nicht von dem Sprecher stammen (sog. Gegenhypothese). Somit wird für die

Entscheidung, welcher Sprecher aktiv ist, anstelle der *Likelihood* $p(\boldsymbol{X}_{1:N}|\Omega = i)$ das Verhältnis der *Likelihoods* mit

$$\Lambda(\boldsymbol{X}_{1:N}|\Omega = i) \quad = \quad \prod_{k=1}^{N} \frac{p(\boldsymbol{x}(k)|\Omega = i)}{p(\boldsymbol{x}(k)|\Omega \neq i)} \tag{4.42}$$

betrachtet. Die Hypothese $\widehat{\Omega}$ für das wahrscheinlichste Sprechermodell ist dann durch das Sprechermodell gegeben, das die Summe der logarithmierten *Likelihood*-Verhältnisse maximiert:

$$\widehat{\Omega} \quad = \quad \underset{i}{\operatorname{argmax}} \left\{ \sum_{k=1}^{N} \log \left(\frac{p(\boldsymbol{x}(k)|\Omega = i)}{p(\boldsymbol{x}(k)|\Omega \neq i)} \right) \right\}. \tag{4.43}$$

Die Bildung des Logarithmus wird zur Verbesserung der numerischen Stabilität verwendet und hat dabei keinen Einfluss auf die argmax-Operation.

Das Modell für die Gegenhypothese, auch universelles Hintergrundmodell (engl. *Universal Background Model*, *UBM*) genannt, kann entweder aus den Aufnahmen eines unabhängigen Satzes von Sprechern oder aus der Datenmenge aller zu trainierenden Sprecher geschätzt werden [RQD00]. In dieser Arbeit wird der zweite Ansatz gewählt, da hierbei auch mit kleineren Datenmengen Sprechermodelle gut trainiert werden können.

Das universelle Hintergrundmodell ($\Omega = \Omega_{UBM}$) setzt sich aus der Kombination der geschlechtsspezifischen Hintergrundmodelle für Männer ($\Omega = \Omega_{UBM}^{M}$) und Frauen ($\Omega = \Omega_{UBM}^{F}$) zusammen. Da kein a priori Wissen über das Geschlecht der anwesenden Sprecher vorhanden ist, wird eine Gleichgewichtung der geschlechtsspezifischen Hintergrundmodelle mit

$$p(\boldsymbol{x}(k)|\Omega \neq i) = p(\boldsymbol{x}(k)|\Omega = \Omega_{UBM}) \qquad i = 1, \ldots, \mathcal{I} \tag{4.44}$$

$$= \frac{1}{2} p(\boldsymbol{x}(k)|\Omega = \Omega_{UBM}^{M}) + \frac{1}{2} p(\boldsymbol{x}(k)|\Omega = \Omega_{UBM}^{F}) \tag{4.45}$$

vorgenommen. Die Modellparameter werden jeweils aus den gesamten Daten der weiblichen bzw. männlichen Sprecher mittels *ML*-Parameterschätzung bestimmt [DHS01]. Dabei kann die Verwendung von Trainingsdaten aus unterschiedlichen Aufnahmesituationen und Mikrophonarten, wie z. B. Nahbereichs- und Fernfeldmikrophonen, die Robustheit der Sprecheridentifikation gegenüber Veränderungen der Aufnahmesituation verbessern.

Jede Dichtefunktion wird durch eine Gauß'sche Mischungsverteilung (*GMM*) beschrieben, deren Gewichte $c_{j,m}$, Mittelwertvektoren $\boldsymbol{\mu}_{j,m}$ und Kovarianzmatrizen $\boldsymbol{\Sigma}_{j,m}$ aus Trainingsdaten bestimmt werden. Das *GMM* des j-ten Modells (Sprechermodell oder geschlechtsspezifisches Hintergrundmodell) ist folglich als gewichtete Summe von M multivariaten Normalverteilungen mit

$$p(\boldsymbol{x}(k)|\Omega = j) = \sum_{m=1}^{M} c_{j,m} \cdot \mathcal{N}\left(\boldsymbol{x}(k); \boldsymbol{\mu}_{j,m}, \boldsymbol{\Sigma}_{j,m}\right) \qquad j = 1, \ldots, \mathcal{I}, \Omega_{UBM}^{F}, \Omega_{UBM}^{M} \tag{4.46}$$

definiert. Dabei ist das Gewicht $c_{j,m}$ die a priori Wahrscheinlichkeit der m-ten Mischungsverteilung der j-ten Klasse mit $c_{j,m} = P(Z = m|\Omega = j)$. Die Zufallsvariable $Z \in \{1, \ldots, M\}$ stehe für die Zugehörigkeit zu einer Mischungsverteilung und die Zufallsvariable $\Omega \in \{1, \ldots, \mathcal{I}, \Omega_{UBM}^{F}, \Omega_{UBM}^{M}\}$ für die Zugehörigkeit zu einer Klasse. Jedes Sprechermodell und jedes geschlechtsspezisches Hintergrundmodell besitzt somit einen eigenen Satz von Modellparametern $\boldsymbol{\Theta}_j = \{c_{j,1}, \ldots, c_{j,M}, \boldsymbol{\mu}_{j,1}, \ldots, \boldsymbol{\mu}_{j,M}, \boldsymbol{\Sigma}_{j,1}, \ldots, \boldsymbol{\Sigma}_{j,M}\}$.

Die Modellierung eines Sprechers durch ein *Hidden Markov Model* (*HMM*) bietet nach [RQD00] keinen signifikanten Vorteil gegenüber einer *GMM*-Modellierung, sofern keine Informationen über die gesprochenen Wörter vorliegen.

Die individuellen Sprechermodelle werden mittels Bayes'scher Adaption [RQD00] aus den geschlechtsspezifischen Modellen trainiert. Vorteil dieser Methode ist, dass auch Modelle für Sprecher mit geringen Datenmengen trainiert werden können, da nur die Teile der Modelle angepasst werden, die auch beobachtet worden sind. Liegen für einen Sprecher nur wenige Beobachtungen vor, so entspricht sein Modell zu einem großen Teil dem geschlechtsspezifischen Hintergrundmodell. Dies bedeutet aber auch, dass die Hintergrundmodelle eine hohe Ähnlichkeit mit den zu trainierenden Sprechern haben müssen. Weibliche Sprecher werden folglich ausgehend von einem weiblichen Hintergrundmodell trainiert und männliche Sprecher mit dem männlichen Hintergrundmodell. Die Schätzung der Modellparameter der Sprechermodelle erfolgt durch eine Bayes'sche Adaption der geschlechtsspezifischen Hintergrundmodelle.

Die Bayes'sche Adaption berechnet auf Basis des Hintergrundmodells zunächst die Wahrscheinlichkeit, dass der Merkmalsvektor $\boldsymbol{x}(k)$ zur m-ten Mischungsverteilung gehört:

$$p(Z = m|\boldsymbol{x}(k), \Omega = \Omega^*_{UBM}) = \frac{p(\boldsymbol{x}(k)|Z = m, \Omega = \Omega^*_{UBM}) c_{\Omega^*_{UBM},m}}{\sum_{j=1}^{M} p(\boldsymbol{x}(k)|Z = j, \Omega = \Omega^*_{UBM}) c_{\Omega^*_{UBM},j}}. \tag{4.47}$$

Dabei sei Z die Zufallsvariable der Zugehörigkeit zu einer Mischungsverteilung und Ω^*_{UBM} das geschlechtsspezifische Hintergrundmodell, welches entsprechend dem Sprecher zu Ω^M_{UBM} oder Ω^F_{UBM} gewählt wird. Anschließend werden die sprecherspezifische Modellparameter $\widetilde{\boldsymbol{\Theta}}_i$ mit

$$\widetilde{c}_{i,m} = \frac{1}{N} \sum_{k=1}^{N} p(Z = m|\boldsymbol{x}(k), \Omega = \Omega^*_{UBM}) \tag{4.48}$$

$$\widetilde{\boldsymbol{\mu}}_{i,m} = \frac{1}{N\widetilde{c}_{i,m}} \sum_{k=1}^{N} p(Z = m|\boldsymbol{x}(k), \Omega = \Omega^*_{UBM}) \cdot \boldsymbol{x}(k) \tag{4.49}$$

$$\widetilde{\boldsymbol{\Sigma}}_{i,m} = \frac{1}{N\widetilde{c}_{i,m}} \sum_{k=1}^{N} p(Z = m|\boldsymbol{x}(k), \Omega = \Omega^*_{UBM})(\boldsymbol{x}(k) - \boldsymbol{\mu}_i)(\boldsymbol{x}(k) - \boldsymbol{\mu}_i)^T \tag{4.50}$$

geschätzt, welche in Kombination mit den Modellparametern des gewählten Hintergrundmodells $\boldsymbol{\Theta}_{\Omega^*_{UBM}}$ das neue Sprechermodell $\boldsymbol{\Theta}_i$ bilden:

$$c_{i,m} = \epsilon_i \cdot \widetilde{c}_{i,m} + (1 - \epsilon_i) \cdot c_{\Omega^*_{UBM},m} \tag{4.51}$$

$$\boldsymbol{\mu}_{i,m} = \epsilon_i \cdot \widetilde{\boldsymbol{\mu}}_{i,m} + (1 - \epsilon_i) \cdot \boldsymbol{\mu}_{\Omega^*_{UBM},m} \tag{4.52}$$

$$\boldsymbol{\Sigma}_{i,m} = \epsilon_i \cdot \widetilde{\boldsymbol{\Sigma}}_{i,m} + (1 - \epsilon_i) \cdot \boldsymbol{\Sigma}_{\Omega^*_{UBM},m}. \tag{4.53}$$

Der Adaptionskoeffizient ϵ_i, der die Gewichtung der sprecherspezifischen Modellparameter $\widetilde{\boldsymbol{\Theta}}_i$ gegenüber den Parametern der Hintergrundmodelle $\boldsymbol{\Theta}_{\Omega^*_{UBM}}$ einstellt, wird mit

$$\epsilon_i = \frac{N \cdot \widetilde{c}_{i,m}}{N \cdot \widetilde{c}_{i,m} + r} \tag{4.54}$$

berechnet. Der Relevanzfaktor r aus Gl. 4.54 steuert hierbei den Einfluss des Hintergrundmodells, wobei für den Fall $r = 0$ die Relevanz des Hintergrundmodells zu Null gesetzt wird und die Bayes'sche Adaption in die *ML*-Parameterschätzung des *EM*-Algorithmus übergeht.

Theoretisch ist es möglich, unterschiedliche Relevanzfaktoren für die Adaption von Modellparametern ($c_{i,m}, \boldsymbol{\mu}_{i,m}, \boldsymbol{\Sigma}_{i,m}$) zu nutzen. Jedoch haben experimentelle Untersuchungen keine signifikanten Vorteile gezeigt, und daher werden die nachfolgenden Experimente jeweils mit einem für alle Parameter gültigen Relevanzfaktor durchgeführt.

Da der Einsatz in der vernetzten Hausumgebung den Zweck hat, den Benutzer nahezu in Echtzeit zu erkennen, um ihm bei seinen täglichen Arbeiten zu unterstützen, muss bei dem Verfahren zur Sprecheridentifikation der Aspekt der echtzeitfähigen Verarbeitung von Datenströmen betrachtet werden. Die Sprecheridentifikation als Systemkomponente trägt nicht zur Latenz des Systems bei, da lediglich für jeden Merkmalsvektor die *Likelihood* der Sprecher nach Gl. 4.43 berechnet werden muss. Dies führt nicht zu einer Verzögerung, jedoch zu einer hohen Rechenlast, falls eine große Personengruppe trainiert ist. Eine Option zur Verringerung der Rechenlast ist die Reduktion der Anzahl der berechneten Exponentialfunktionen, indem nur die Verteilungen der Gauß'schen Mischungsverteilung der Sprecher berechnet werden, bei denen die *Likelihood* des Hintergrundmodells einen minimalen Wert überschreitet.

An dieser Stelle wird nicht auf die Detektion von Personen eingegangen, die nicht in der Gruppe der bekannten Sprecher enthalten sind. Da das System im vernetzten Haus zur Unterstützung der Hausbewohner verwendet werden soll, ist die Annahme gerechtfertigt, dass alle Personen im Haushalt bekannt sind und dass deren Anzahl nicht sonderlich groß ist. Ein Ansatzpunkt für eine solche Detektion ist die Einführung eines Grenzwertes für die Summe der *Likelihood*-Verhältnisse in Gl. 4.43. Überschreitet keine der Sprecherhypothesen einen festgesetzten Schwellwert, so wird angenommen, dass der Sprecher nicht aus der Gruppe der bekannten Sprecher stammt. Dieser Ansatz ermöglicht die Erkennung von unbekannten Sprechern und reduziert die Anzahl der falsch klassifizierten Personen, jedoch zu Lasten einer neuen Fehlerart, der fälschlich zurückgewiesenen Sprecher.

4.3.2 Gemeinsame Sprecherwechseldetektion und Identifikation

In den vorherigen Kapiteln wurde beschrieben, wie zunächst eine Sprecherwechseldetektion und anschließend eine Sprecheridentifikation durchgeführt werden kann. Dieses sequentielle Vorgehen hat den Nachteil, dass die zunächst in der Segmentierung getroffenen „frühen" Entscheidungen nur auf einem Teil der vorhandenen Informationen beruhen. Denn die Sprecheridentität ist zum Zeitpunkt der Sprecherwechseldetektion noch nicht bekannt. Daher wurde die Idee entwickelt, die Identifikation und die Segmentierung parallel durchzuführen. Somit kann das Treffen von vorläufigen Entscheidungen vermieden und eine endgültige Entscheidung unter Verwendung aller Wissensquellen getroffen werden, so dass alle vorhandenen Informationen mit in die finale Entscheidung einfließen. Für die detaillierte Beschreibung des Ansatzes wird die Defintion des *Hidden Markov Models* benötigt, welche entsprechend [Rab89] im Folgenden gegeben wird.

Hidden Markov Model

Ein *Hidden Markov Model* ist ein stochastisches Modell für ein System, welches durch eine diskrete Markov-Kette erster Ordnung beschreibbar ist. Das Modell besteht aus einer Menge von $\mathcal{I}$ Zuständen, von denen einer der aktuelle Zustand ist, in dem sich das System befindet. In gleichmäßigen Zeitabständen wechselt das System von einem Zustand in einen anderen, wobei der Folgezustand auch der vorherige Zustand sein kann (vgl. Abb. 4.9). Diese Zustandsübergänge werden probabilistisch durch die Transitionswahrscheinlichkeiten

$$a_{ij} = P(\Omega(k) = j | \Omega(k-1) = i) \qquad 1 \leq i, j \leq \mathcal{I} \tag{4.55}$$

beschrieben, wobei $\Omega(k)$ der aktuelle Zustand des Systems zum Zeitpunkt k und $\Omega(k-1)$ der vorherige Zustand des System sein soll. Die Wahrscheinlichkeit, dass sich das System zum Startzeitpunkt im Zustand i befindet, ist mit

$$\pi_i = P(\Omega(0) = i) \qquad 1 \leq i \leq \mathcal{I} \tag{4.56}$$

gegeben. Der aktuelle Zustand des Systems ist nicht direkt beobachtbar (engl. *hidden*), jedoch emittiert das System zu regelmäßigen Zeitpunkten k die beobachtbaren Merkmalsvektoren $\boldsymbol{x}(k)$. Des Weiteren werden die Verteilungsdichtefunktionen, welche die Emissionswahrscheinlichkeiten der Zustände beschreiben, als bekannt vorausgesetzt. Somit sind die Emissionswahrscheinlichkeiten der Zustände mit

$$b_i(\boldsymbol{x}(k)) = p(\boldsymbol{x}(k) | \Omega = i) \qquad 1 \leq i \leq \mathcal{I} \tag{4.57}$$

bekannt. Das System ist vollständig durch die Wahrscheinlichkeiten aus Gl. 4.55, Gl. 4.56 und Gl. 4.57 beschrieben, wobei diese Wahrscheinlichkeiten häufig in vektorieller Schreibweise zusammengefasst werden. Die Transitionswahrscheinlichkeiten bilden dabei die Transistionsmatrix

$$\boldsymbol{A} = (a_{ij}) \qquad 1 \leq i, j \leq \mathcal{I}. \tag{4.58}$$

Ferner werden die Verteilungsdichtefunktionen der Emissionswahrscheinlichkeiten in $\boldsymbol{B}$ und die Anfangswahrscheinlichkeiten der Zustände in dem Vektor $\boldsymbol{\pi}$ zusammengefasst. Das Modell des *HMM* kann folglich kurz mit $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ angegeben werden.

Sprecherprotokollierung mittels eines *Hidden Markov Models*

Kern der Sprecherprotokollierung ist ein *Hidden Markov Model* zur Modellierung der Sprecher, deren Zustandsübergänge abhängig von Informationen über Sprecherwechsel und damit zeitvariant sind. Um der Anforderung nach einer geringen Latenz nachzukommen, wird ein Viterbi-Dekodierer mit vorzeitiger Ausgabe der Erkennungsergebnisse (ein sog. *Partial Traceback*) verwendet, der die optimale Abfolge der Zustände im *HMM*, gegeben die Beobachtungen, bestimmt.

Jeder der $\mathcal{I}$ Sprecher wird durch einen Zustand in diesem *Hidden Markov Model* repräsentiert. Zusätzlich wird ein Zustand $\mathcal{I}+1$ für Stille eingefügt, um Sprachpausen zu modellieren. Abbildung 4.9 zeigt ein Beispiel für $\mathcal{I} = 3$ Sprecher. Die Emissionswahrscheinlichkeiten der Zustände sind durch die *Likelihoods* der Sprecheridentifikation gegeben. Informationen über mögliche Sprecherwechsel fließen in die Transitionswahrscheinlichkeiten des

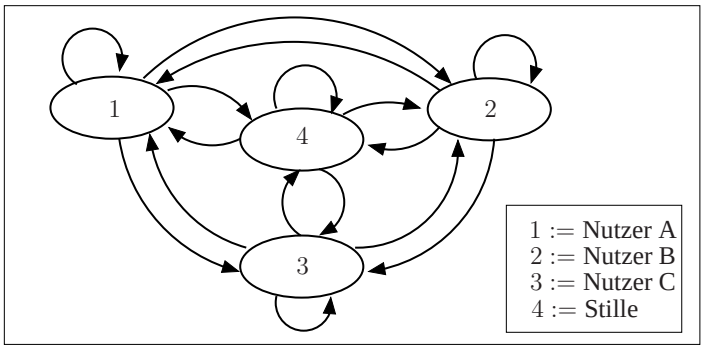


Abbildung 4.9: *Hidden Markov Model* zur Modellierung einer Sprechergruppe

HMM ein. Zustandsübergänge, die einen Sprecherwechsel anzeigen, erhalten eine erhöhte Wahrscheinlichkeit, falls Informationen über einen möglichen Sprecherwechsel vorliegen. Gleichzeitig werden die Wahrscheinlichkeiten der Zustandsübergänge reduziert, die wieder in den aktuellen Zustand führen. Ist ein Sprecherwechsel eher unwahrscheinlich, so erhalten die Zustandsübergänge, die einen Sprecherwechsel anzeigen, niedrigere Wahrscheinlichkeiten und die restlichen Zustandsübergänge höhere Wahrscheinlichkeiten. Somit entsteht eine zeitveränderliche Transitionsmatrix, welche den aktuellen Wissensstand über Sprecherwechsel repräsentiert.

Informationsquellen

Die Schätzung der Transitionswahrscheinlichkeiten soll auf Informationen über Sprecherwechselhypothesen basieren. Hierzu können die akustische Positionsschätzung und die berechneten ΔBIC-Werte verwendet werden. In Abb. 4.10 ist eine Übersicht der Systemkom-

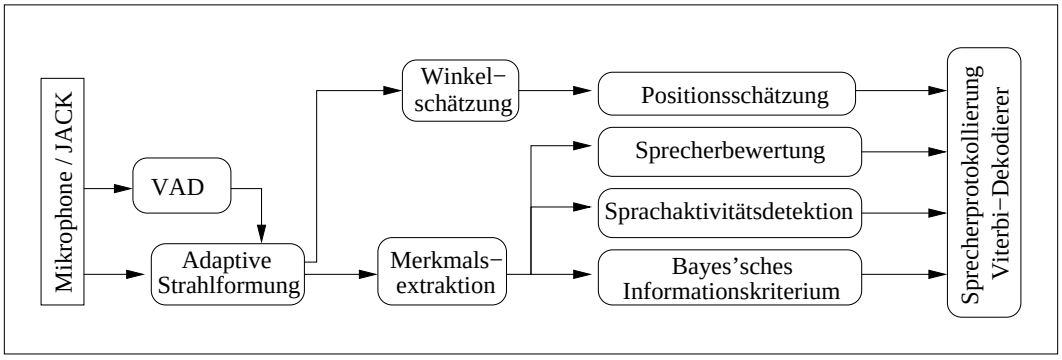


Abbildung 4.10: Systemkomponenten der Sprecherprotokollierung

ponenten für die Sprecherprotokollierung gegeben. Das Modul der Sprecherprotokollierung implementiert einen Viterbi-Dekodierer, der die berechneten Werte des Bayes'schen Informationskriteriums (ΔBIC-Werte) und die Werte der Positionsschätzung verwendet, um die Transitionsmatrix des *HMM* zu schätzen. Für die Emissionswahrscheinlichkeiten des *HMM* werden im Viterbi-Dekodierer die Werte der Sprachaktivitätsdetektion und die *Likelihoods* der Sprecheridentifikation kombiniert, welche im Modul „Sprecherbewertung" berechnet werden. Hierzu wird jede Informationsquelle, soweit noch nicht geschehen, probabilistisch modelliert.

Das in Kap. 4.3.1 vorgestellte Verfahren zur Detektion von Sprecherwechseln berechnet fortlaufend ΔBIC-Werte aus den eingehenden Merkmalsvektoren. Die Berechung der metrischen Entscheidungsregel zur Sprecherwechseldetektion ist mit einer zusätzlichen zeitlichen Latenz behaftet, da signifikante lokale Maxima detektiert werden müssen. Folglich wird zur probabilistischen Modellierung von Informationen über Sprecherwechsel, statt der metrischen Entscheidungsregel, die Varianz der ΔBIC-Werte verwendet. Diese mit $x^{bic}(k)$ bezeichnete Größe kann mit

$$\mu^{bic}(k) = \alpha \cdot \mu^{bic}(k-1) + (1-\alpha) \cdot \Delta BIC(k) \tag{4.59}$$

$$x^{bic}(k) = \beta \cdot x^{bic}(k-1) + (1-\beta) \cdot \left[\Delta BIC(k) - \mu^{bic}(k)\right]^2 \tag{4.60}$$

geschätzt werden. Vorteilhaft bei diesem Ansatz ist die Vermeidung von Latenzen durch die rekursive Schätzung der Varianz. Für die Modellierung werden die Parameter der Normalverteilungen $p(x^{bic}(k)|c(k)=0)$ und $p(x^{bic}(k)|c(k)=1)$ aus Trainingsdaten geschätzt. Hierbei ist $c(k)$ eine binäre Zufallsvariable, welche angibt, ob ein Sprecherwechsel vorliegt ($c(k)=1$) oder nicht ($c(k)=0$).

Der *FSB*, als adaptiver Strahlformer, adaptiert blind auf den stärksten Sprecher und ermöglicht durch die Korrelation der Filterimpulsantworten die Schätzung des Einfallswinkels des Sprachsignals (vgl. Kap. 4.2.2). Für den Fall, dass mehr als eine Mikrophongruppe zur Verfügung steht, können die Winkelschätzungen zu einer Position $\boldsymbol{P}(k)$ in kartesischen Koordinaten kombiniert werden (vgl. Kap. 4.2.3). Als Indiz für mögliche Sprecherwechsel wird die Varianz $x^{pos}(k)$ der Position berechnet, welche entweder auf Winkelschätzungen oder zweidimensionalen Positonsschätzungen beruht. Erneut wird zur Vermeidung von Latenzen eine rekursive Schätzung verwendet:

$$\mu^{pos}(k) = \alpha \cdot \mu^{pos}(k-1) + (1-\alpha) \cdot \parallel \boldsymbol{P}(k) - \boldsymbol{P}(k-1) \parallel_2 \tag{4.61}$$

$$x^{pos}(k) = \beta \cdot x^{pos}(k-1) + (1-\beta) \cdot \left[\boldsymbol{P}(k) - \mu^{pos}(k)\right]^2 . \tag{4.62}$$

Entsprechend des Ansatzes zur Modellierung der ΔBIC-Werte wurden aus Trainingsdaten die Parameter der Normalverteilungen $p(x^{pos}(k)|c(k)=0)$ und $p(x^{pos}(k)|c(k)=1)$ geschätzt.

Informationen über die mögliche Identität des Sprechers werden durch die Sprecherbewertung ermittelt. Für jeden akustischen Merkmalsvektor $\boldsymbol{x}^{sid}(k)$ wird das *Likelihood*-Verhältnis der einzelnen Sprechermodelle nach Gl. 4.42 als Emissionswahrscheinlichkeit der zu den Sprechern gehörenden *HMM*-Zustände berechnet.

Eine weitere Informationsquelle ist die Sprachaktivitätsdetektion. Hierzu wird das Verfahren aus dem *Extended Advanced Front-end Feature Extraction* (*XAFE*) des *ETSI* [ETS02] verwendet. Die Steuerung der Adaption des Strahlformers erfolgt jedoch mit einer energiebasierten Sprachaktivitätsdetektionen (engl. *Voice Activity Detection*, *VAD*) nach [RS04]. Beide Sprachaktivitätsdetektionen liefern einen Indikator $P(S|\boldsymbol{x}^{sid})$ für Sprache, dessen Wert zwischen 0 (Keine Sprache) und 1 (Sprache) liegt.

Emissionswahrscheinlichkeiten

Die Emissionswahrscheinlichkeiten jedes Sprechers sind gegeben durch die *Likelihood*-Verhältnisse aus Gl. 4.42, deren zugrunde liegende Dichtefunktionen auf Sprachdaten ohne

Sprachpausen für die Sprecheridentifikation trainiert werden. Jedoch treten in dem Datenstrom der Sprecherprotokollierung Zeitabschnitte ohne Sprache auf, so dass das *Likelihood*-Verhältnis mit der Wahrscheinlichkeit, dass der vorliegende Block Sprache enthält, multipliziert werden muss. Somit folgt für die Emissionswahrscheinlichkeit des Sprecherzustandes $\Omega(k) = j$ zum Zeitpunkt k:

$$\begin{aligned} b_j(\boldsymbol{x}^{sid}(k)) &= p'(\boldsymbol{x}^{sid}(k)|\Omega = j) \\ &= \begin{cases} \Lambda(\boldsymbol{x}^{sid}(k)|\Omega = j) \cdot P(S|\boldsymbol{x}^{sid}(k)) & \text{für } j = 1, \ldots, \mathcal{I} \\ \Lambda(\boldsymbol{x}^{sid}(k)|\Omega = j) \cdot (1 - P(S|\boldsymbol{x}^{sid})(k)) & \text{für } j = \mathcal{I} + 1 \end{cases} . \end{aligned} \tag{4.63}$$

Für die Emissionswahrscheinlichkeit des Zustandes Stille wird der Mittelwert der *Likelihood*-Verhältnisse verwendet:

$$\Lambda(\boldsymbol{x}^{sid}(k)|\Omega = \mathcal{I} + 1) = \frac{1}{\mathcal{I}} \sum_{j=1}^{\mathcal{I}} \Lambda(\boldsymbol{x}^{sid}(k)|\Omega = j). \tag{4.64}$$

Transitionswahrscheinlichkeiten

Die Grundidee des Verfahrens ist es, die Wahrscheinlichkeit eines Zustandsübergangs abhängig von den Informationen über die Positionsänderung eines Sprechers und der Varianz der ΔBIC-Werte zu machen. Unter Verwendung der binären Zufallsvariable $c(k)$ und den zuvor vorgestellten probabilistischen Modellierungen der Sprecherwechselinformationen folgt für die Transitionswahrscheinlichkeiten, dass sie proportional zu $P(c(k)|x^{bic}(k), x^{pos}(k))$ gewählt werden. Es wird ferner die Annahme getroffen, dass $x^{bic}(k)$ und $x^{pos}(k)$ statistisch unabhängig sind, so dass gilt:

$$P(c(k)|x^{pos}(k), x^{bic}(k)) = \frac{p(x^{pos}(k), x^{bic}(k)|c(k))P(c(k))}{p(x^{pos}(k), x^{bic}(k))} \tag{4.65}$$

$$= \frac{p(x^{pos}(k)|c(k))P(c(k))}{p(x^{pos}(k))} \frac{p(x^{bic}(k)|c(k))P(c(k))}{p(x^{bic}(k))} \frac{1}{P(c(k))}. \tag{4.66}$$

Unter der Annahme einer gleichförmigen Verteilung von $P(c(k))$ folgt:

$$P(c(k)|x^{pos}(k), x^{bic}(k)) = \frac{p(x^{pos}(k)|c(k))}{\sum_{c'} p(x^{pos}(k)|c(k) = c')} \frac{p(x^{bic}(k)|c(k))}{\sum_{c'} p(x^{bic}(k)|c(k) = c')} \frac{1}{P(c(k))}. \tag{4.67}$$

Die zeitveränderlichen Übergangswahrscheinlichkeiten zwischen den *HMM*-Zuständen werden definiert zu:

$$a_{ij}(k) := P\left(\Omega(k) = j|\Omega(k-1) = i\right) \tag{4.68}$$

$$= \frac{\tilde{a}_{ij}(k)}{\sum_j \tilde{a}_{ij}(k)} \tag{4.69}$$

mit

$$\tilde{a}_{ij}(k) = \begin{cases} P(c(k) = 0|x^{pos}(k), x^{bic}(k)) & i = j, j \neq \mathcal{I} + 1 \\ P(c(k) = 1|x^{pos}(k), x^{bic}(k)) & i \neq j, j \neq \mathcal{I} + 1 \\ P(c(k) = 0|x^{bic}(k)) & i = j = \mathcal{I} + 1 \\ P(c(k) = 1|x^{bic}(k)) & i \neq j, j = \mathcal{I} + 1 \end{cases} \text{für} \quad . \tag{4.70}$$

Der Zustand Stille benötigt, wie aus Gl. 4.70 ersichtlich ist, eine spezielle Anpassung, da für den Fall von Stille offensichtlich keine Positionsschätzung vorliegen kann. Jedoch wird der Übergang von einem Sprecher zu einer Sprachpause und umgekehrt als Sprecherwechsel durch den ΔBIC-Wert angezeigt.

Die Sprecheridentifikation aus Kap. 4.3.1 bietet die Möglichkeit, eine Bestimmung des Geschlechts des aktuellen Sprechers durchzuführen, indem die *Likelihoods* der geschlechtsspezifischen Hintergrundmodelle ermittelt werden. Die Bestimmung des Sprechergeschlechts erwies sich in Experimenten als sehr zuverlässig, jedoch führt die Verwendung dieser Information zur Berechnung der Transitionswahrscheinlichkeiten nur zu geringfügig besseren Ergebnissen. Ein Grund dürfte in der Tatsache liegen, dass Verwechselungen zwischen männlichen und weiblichen Sprechermodellen nur selten auftreten.

Viterbi-Dekodierer

Entfaltet man das Zustandsmodell aus Abb. 4.9 über die Zeit, so entsteht ein Trellisdiagramm (vgl. Abb. 4.11). Ein Viterbi-Dekodierer bestimmt dann den bestbewertesten Pfad durch das Trellis, d. h. die Zustandssequenz $\widehat{\boldsymbol{\Omega}}_{1:N} = [\widehat{\Omega}(1), \ldots, \widehat{\Omega}(N)]$ mit

$$\widehat{\boldsymbol{\Omega}}_{1:N} = \underset{\boldsymbol{\Omega}_{1:N}}{\operatorname{argmax}} \Big\{ \sum_{k=1}^{N} \Big[\log p'(\boldsymbol{x}^{sid}(k)|\Omega) + \kappa \log P(\Omega(k)|\Omega(k-1)) \Big] \Big\}. \tag{4.71}$$

Aus der Literatur ist bekannt, dass Bedingungen hinsichtlich der minimal erlaubten Zeit zwischen Sprecherwechseln und heuristische Ansätze zur Glättung benötigt werden, um das exzessive Wechseln von Zuständen zu vermeiden [TR06]. Dies kann gerechtfertigt werden durch die Annahme, dass selbst eine kurze Sprachäußerung aus mehreren Merkmalsvektoren besteht, die im Abstand von $10\,\mathrm{ms}$ aus dem Sprachsignal berechnet werden. In dem hier vorgestellten Ansatz werden durch den Faktor κ in Gl. 4.71 die Emissionswahrscheinlichkeiten gegenüber den Transitionswahrscheinlichkeiten stärker gewichtet, was zu einer Verminderung der Zustandswechsel führt.

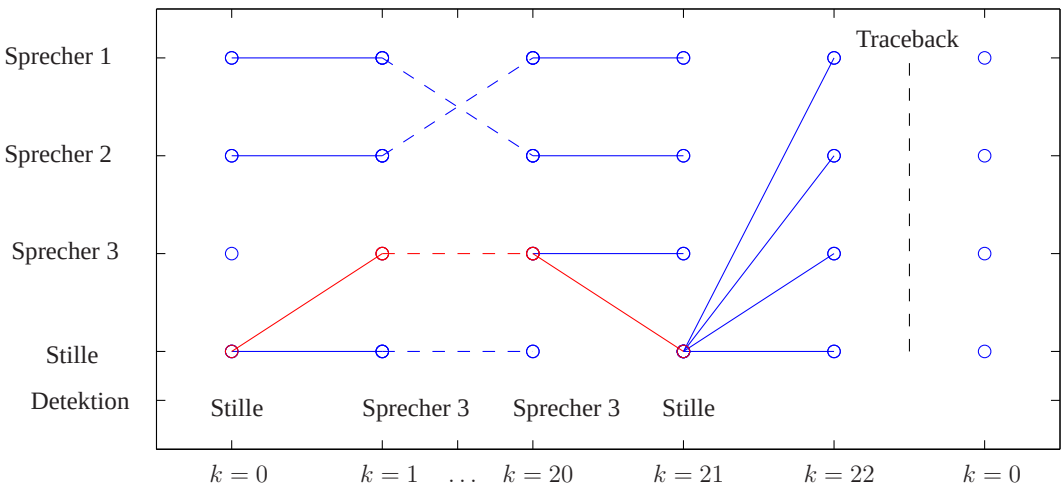


Abbildung 4.11: Beispiel eines Trellisdiagramms und der Ausgabe des Viterbi-Dekodierers

Um den zeitlichen Anforderungen des Systems gerecht zu werden, wird zu jedem Zeitpunkt ein *Partial Traceback* gestartet. Hierbei wird ausgehend von jedem Zustand der Pfad zurückverfolgt, der in dem Zustand endete. Der Teil der Pfade, welcher für alle Zustände gleich ist, bestimmt den eindeutigen Zustandsverlauf in der Vergangenheit. In Abb. 4.11 ist ein Beispiel für das *Partial Traceback* gegeben. Zum Zeitpunkt $k = 22$ wird für die vier Zustände der jeweilige Pfad über die vorangegangenen Zustände bestimmt. Beginnend mit dem Zeitpunkt $k = 21$ ergibt sich für alle Zustände ein eindeutiger Pfad (vgl. Abb. 4.11, roter Pfad). Folglich kann der rot markierte Pfad ausgegeben werden.

Die Anzahl der Zeitschritte, die man in die Vergangenheit gehen muss, bis der Pfad eindeutig ist, ist zufällig. Daher wird eine maximale Latenz τ_{max} eingeführt, ab der eine Ausgabe des Pfades erzwungen wird. Sollte kein eindeutiger Pfad existieren und gleichzeitig die maximale noch tolerierbare Latenz τ_{max} überschritten werden, so wird der am besten bewertete Pfad gewählt. Experimentelle Untersuchungen zeigen, dass in einem Großteil der Fälle der eindeutige Pfad frühzeitig vorliegt (vgl. Kap. 4.4.6).

Zunächst wurde die Information über Sprecherpositionen nur verwendet, um Sprecherwechsel zu detektieren. Man beachte, dass mit dem Ergebnis der Viterbi-Dekodierung eine Zuordnung der Positionsschätzungen zu den Sprechermodellen erfolgen kann. Dies ermöglicht für jeden Sprecher eine individuelle Nachfilterung der Positionsschätzungen, welche durch die Verwendung von Kalman- oder Partikelfiltern realisiert werden kann [WPH04].

4.3.3 Experimentelle Ergebnisse

Ein System zur Sprecherprotokollierung setzt sich aus verschiedenen Komponenten zusammen, die sich gegenseitig in ihrer Leistungsfähigkeit beeinflussen. Eine fehlerhafte Segmentierung des Datenstroms wird zwangsläufig auch zu Fehlern in der Sprecheridentifikation führen. Daher werden zunächst die Komponenten einzeln in Experimenten untersucht und anschließend der Gesamtaufbau betrachtet. Die hierfür benötigten Fehlermaße und Datenbasen werden zu Beginn erläutert.

Fehlermaße

Eine objektive Beurteilung der Segmentierung von Daten erfordert zunächst ein Fehlermaß, welches unabhängig von der Leistungsfähigkeit der nachgeschalteten Klassifikation ist. Hierfür geeignet sind die in [DW00] eingeführten Fehlermaße der *False Alarm Rate* (*FAR*) mit

$$FAR = \frac{\textit{Anzahl fehlerhafter Alarme}}{\textit{Anzahl Segmentierungspunkte} + \textit{Anzahl fehlerhafter Alarme}}\% \tag{4.72}$$

und der *Missed Detection Rate* (*MDR*) mit

$$MDR = \frac{\textit{Anzahl verpasster Detektionen}}{\textit{Anzahl Segmentierungspunkte}}\%. \tag{4.73}$$

Die Abb. 4.12 zeigt beispielhaft die Fehlerarten bei der Segmentierung. Zu den Zeitpunkten $2\,\mathrm{s}$, $7\,\mathrm{s}$ und $10\,\mathrm{s}$ findet ein Sprecherwechsel in den Aufnahmen statt. Angezeigt werden Sprecherwechsel durch die ΔBIC-Werte zu den Zeitpunkten $2{,}2\,\mathrm{s}$, $4\,\mathrm{s}$, $5{,}5\,\mathrm{s}$ und $9{,}8\,\mathrm{s}$. Grüne und rote Flächen um die Zeitpunkte der Sprecherwechsel zeigen die erlaubten Toleranzbereiche für die Detektion der Segmentierungspunkte an. Ein Segmentierungspunkt wird als

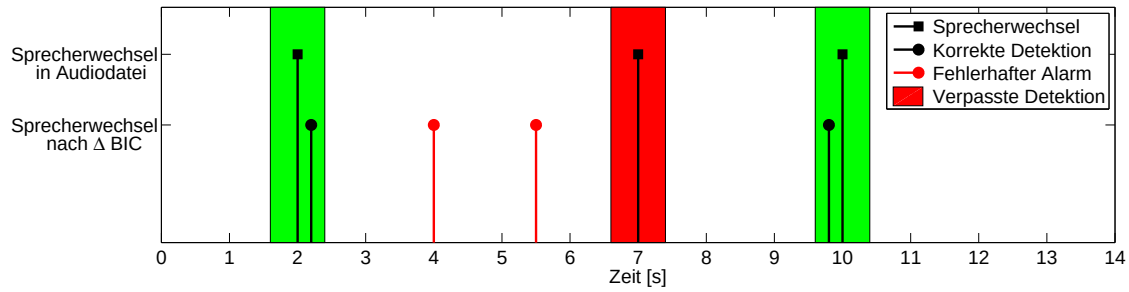


Abbildung 4.12: Fehlerarten bei der Segmentierung von Audiodaten

verpasst eingestuft, falls in einem Bereich von $\pm 0{,}4\,\mathrm{s}$ um den Segmentierungspunkt kein Sprecherwechsel durch das System angezeigt wird (vgl. Abb. 4.12, Zeitpunkt: $7\,\mathrm{s}$). Fehlerhafte Alarme sind alle vom System gemeldeten Sprecherwechsel in deren zeitlicher Umgebung ($\pm 0{,}4\,\mathrm{s}$) keine Sprecherwechsel (vgl. Abb. 4.12, Zeitpunkte: $4\,\mathrm{s}$, $5{,}5\,\mathrm{s}$) vorliegen. Der Vergleich zwischen zwei Verfahren zur Segmentierung anhand einer *Receiver Operating Characteristic* (*ROC*) kann durch die *Equal Error Rate* (*EER*) erfolgen, welche durch den Punkt auf der *ROC*, an der die *FAR* und die *MDR* übereinstimmen, definiert ist.

Ein Fehlermaß für die Beurteilung der Klassifikationsleistung durch eine der Segmentierung nachgeschalteten Sprecheridentifikation ist die *Diarization Error Rate* (*DER*) mit

$$DER = \frac{\textit{Anzahl der einem Sprecher fehlerhaft zugeordneten Merkmalsvektoren}}{\textit{Anzahl Merkmalsvektoren}} \%, \quad (4.74)$$

welche durch *NIST* [NIS08a] definiert wurde. Sie ist ein Maß für die Leistungsfähigkeit des Segmentierungs- und Identifikationsprozesses, der zusammengefasst als Sprecherprotokollierung bezeichnet wird.

Datenbasis Sprecherprotokollierung

Das zuvor beschriebene Verfahren zur gemeinsamen Sprecherwechseldetektion und Sprecheridentifikation stellt höhere Anforderungen an eine Datenbasis als einige klassische Ansätze zur Sprecherprotokollierung. Die Datenbasis des *DARPA EARS Rich Transcription Evaluation Projects* [NIS08b] kann zum Beispiel nicht verwendet werden, da bei den Aufnahmen keine Mikrophongruppen verwendet wurden, welche eine Positionsschätzung des Sprechers erlauben würden. Die Datenbasis des *CHIL* Projektes bietet theoretisch mit den verwendeten Mikrophongruppen die Möglichkeit eine Positionsschätzung durchzuführen [OSBC06]. Jedoch sind die Aufnahmen aus den Seminaren ungeeignet, da zu einem großen Teil nur ein Sprecher aktiv ist und insgesamt nur eine geringe Anzahl von Sprecherwechseln vorhanden ist. Daher wurde für die experimentellen Untersuchungen eine eigene Datenbasis erstellt, um gezielt die Komponente des Systems zu untersuchen.

In Abb. 4.13 ist der Aufbau zur Erstellung einer Datenbasis skizziert. Sie umfasst insgesamt $1{,}5$ Stunden gelesene Texte von 5 Frauen und 5 Männern. Dabei wurden die Sprecher sowohl durch eine Mikrophongruppe in $2{,}8\,\mathrm{m}$ Abstand als auch durch Nahbereichsmikrophone aufgenommen. Zwischen den Sprechern befand sich eine schalldämpfende Wand, so dass die Nahbereichsmikrophone nur einen geringen Anteil der Sprache des entfernten Sprechers aufnehmen konnten. In einem Nachbearbeitungsschritt wurden die Nahbereichsaufnahmen einer adaptiven Filterung unterzogen, um den entfernten Sprecher zusätzlich zu

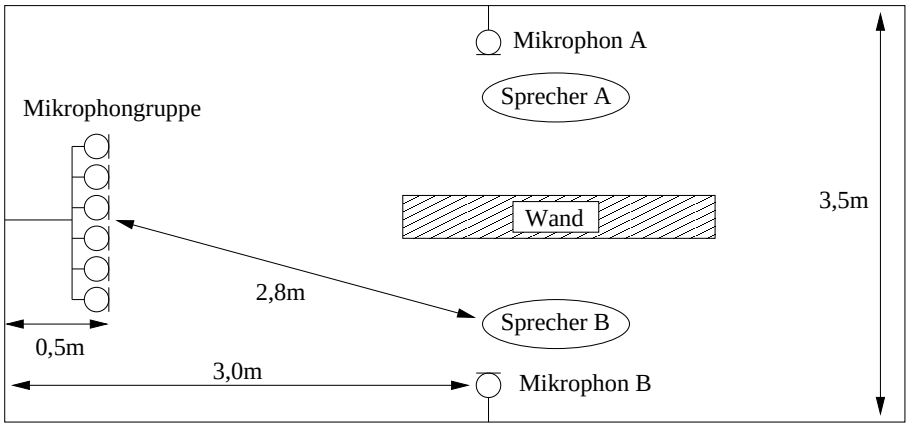


Abbildung 4.13: Versuchsaufbau zur Erstellung einer Datenbasis zur Sprecherwechseldetektion

unterdrücken. Basierend auf den bearbeiteten Nahbereichsaufnahmen war eine zuverlässige automatische Detektion des aktiven Sprechers und somit eine Annotation der Datenbasis möglich.

Die Texte wurden durch Sprecher abwechselnd abschnittsweise gelesen, wobei die Länge der Passagen vorgegeben wurde. Anschließend wurde die Datenbasis in drei Gruppen entsprechend der mittleren Passagenlängen eingeteilt. Dies waren schnelle Sprecherwechsel ($< 2\,\mathrm{s}$), mittlere Sprecherwechsel ($3 - 4\,\mathrm{s}$) und langsame Sprecherwechsel ($> 4\,\mathrm{s}$), die ohne längere Sprechpausen durchgeführt wurden.

Datenbasis Sprecheridentifikation

Die „*CHIL Campaign 2004 - Speaker Identification and Verification*“ des *CHIL* Projektes stellt eine Datenbasis für die Evaluierung von Systemen zur Sprecheridentifikation bereit [SSM05]. Sie besteht aus annotierten Seminaraufnahmen von 11 Sprechern, die parallel jeweils mit einem entfernten (engl. *Distant Talking Microphone*, *DTM*) und einem nahen Mikrophon (engl. *Close Talking Microphone*, *CTM*) aufgenommen wurden. Der Vergleich mit den veröffentlichten Ergebnisse der Evaluierung in [Mos05] und [ZLB+05] ermöglichen einen Einordnung des in dieser Arbeit beschriebenen Systems zur Sprecheridentifikation.

Die Daten der Datenbasis sind mit $16\,\mathrm{Bit}$ pro Abtastwert bei einer Abtastrate von $16\,\mathrm{kHz}$ gespeichert. In den Aufnahmen sind Hintergrundgeräusche aus den Seminaren, wie z. B. der Lüfter eines Projektors, vorhanden. Eine Segmentierung der Daten in homogene Abschnitte definierter Länge

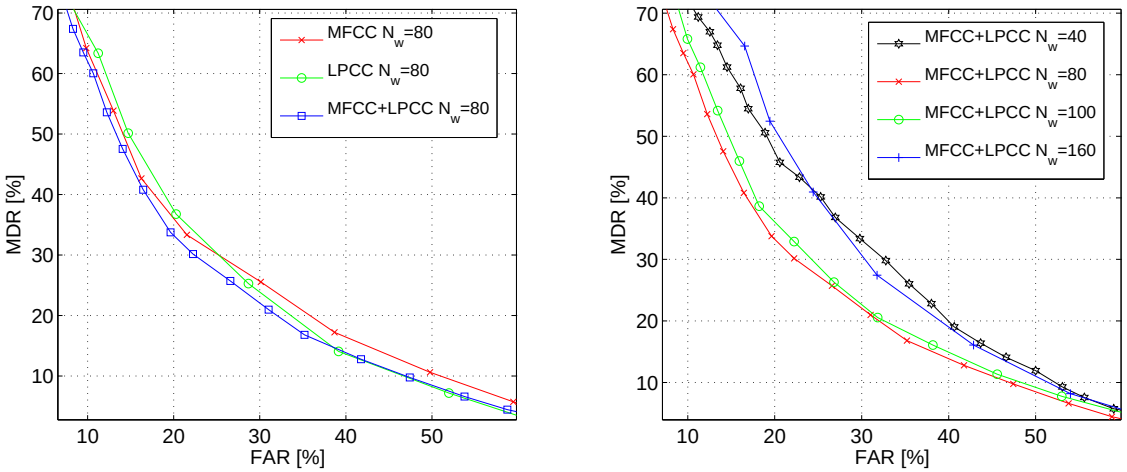
- Training ($33\,\mathrm{min}$): $30\,\mathrm{s}$, $60\,\mathrm{s}$
- Test ($11\,\mathrm{h}$): $1\,\mathrm{s}$, $3\,\mathrm{s}$, $5\,\mathrm{s}$, $10\,\mathrm{s}$, $30\,\mathrm{s}$, $60\,\mathrm{s}$

und eine Sortierung nach Fern- und Nahbereichsaufnahmen wurde durch *ELDA* [ELD08] vorgenommen.

Experimente zur Segmentierung

Die Ergebnisse in diesem Unterkapitel fassen die Experimente im Bereich der Segmentierung von Sprachdaten durch ΔBIC-Werte zusammen. Zunächst wird ein Vergleich der Seg-

mentierungsleistung für verschiedene Merkmalsvektoren und Fenstergrößen durchgeführt. Aus Abb. 4.14 (a) ist ersichtlich, dass die *Mel-Frequency Cepstral Coefficients* (*MFCC*) und die *Linear Prediction Cepstral Coefficients* (*LPCC*) vergleichbare Ergebnisse für die Segmentierung liefern. Die Kombination der beiden Merkmalsvektoren verbessert die Ergebnisse leicht, jedoch führt dieser Ansatz zu einer erheblichen Erhöhung der Systemlast und wird daher nicht weiter verfolgt. Der Vergleich unterschiedlicher Fenstergrößen in Abb. 4.14 (b)



(a) Vergleich der Merkmalsvektoren (b) Variation der Fenstergröße

Abbildung 4.14: Experimente mit Nahbereichsmikrophonen zur Merkmalsvektorwahl und Fenstergröße

zeigt eine optimale Fenstergröße im Sinne der kleinsten *EER* von ca. 80 Merkmalsvektoren für die verwendete Datenbasis. Dies entspricht einer Latenz durch die Segmentierung von 40 Merkmalsvektoren ($320\,\mathrm{ms}$).

In Abb. 4.15 ist der Vergleich der Segmentierungsergebnisse zwischen Fernfeldmikrophonen (*DTM*) und Nahbereichsmikrophonen (*CTM*) dargestellt. Der experimentelle Aufbau ist in Abb. 4.13 (S. 41) dargestellt und bestand aus einer linear angeordneten Mikrophongruppe aus 6 Fernfeldmikrophonen im Abstand von $0{,}05\,\mathrm{m}$ mit einer Distanz von ca. $2{,}8\,\mathrm{m}$ zu den Sprechern.

Die aus der Distanz zwischen Sprechern und Mikrophonen resultierende Verschlechterung der Signalqualität durch Echos und Rauschen führt zu einer Erhöhung der *EER* um ca. $7{,}0$ Prozentpunkte gegenüber den Ergebnissen der Nahbereichsmikrophone (vgl. „*CTM*, Wiener“ und „*DTM* 1 Kanal“). Die Verwendung eines Wiener-Filters („*DTM* 1 Kanal, Wiener“) oder einer akustischen Strahlformung („*DTM* 6 Kanal, *FSB*“) verbessert die *EER* gegenüber den einkanaligen Ergebnissen und erreicht fast die Ergebnisse mit Nahbereichsmikrophonen. Jedoch erst die Einbeziehung von Positionsdaten (vgl. Kap. 4.3.1) ermöglicht eine signifikante Reduktion der *EER* auf ca. $13{,}8\,\%$ („*DTM* 6 Kanal, Winkel“). Dieser Ansatz führt eine Nachfilterung der angezeigten Sprecherwechsel anhand der Positionsschätzungen durch und übertrifft auf diese Weise deutlich die Ergebnisse der Nahbereichsmikrophone.

Experimente zur Sprecheridentifikation

Die Tabellen 4.2 und 4.3 fassen die Ergebnisse der vorgestellten Sprecheridentifikation für die *CHIL* Datenbasis zusammen. Sie ermöglichen den Vergleich der Klassifikationsraten für

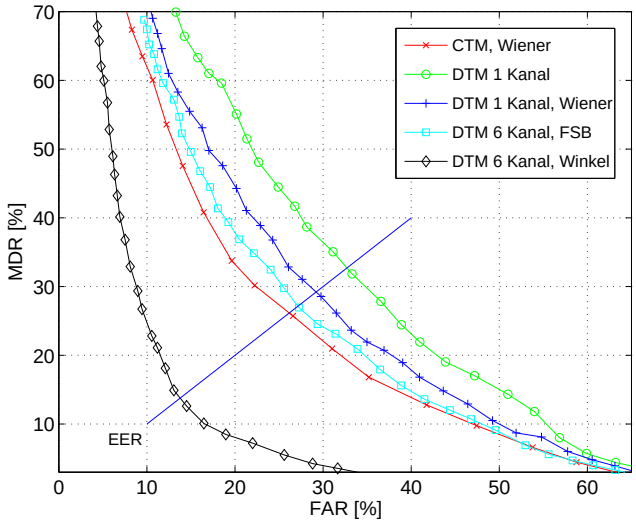


Abbildung 4.15: Vergleich der Segmentierungsergebnisse von Fernfeldmikrophonen (*DTM*) und Nahbereichsmikrophonen (*CTM*)

Nahbereichsaufnahmen und Aufnahmen aus größeren Distanzen für unterschiedliche Trainingsdatensätze und Datenmengen. Angemerkt sei dabei, dass die entfernten Aufnahmen nicht einer akustischen Strahlformung unterzogen werden können, da es sich um einkanalige Aufnahmen handelt.

Training \ Test	Klassifikationsrate (*CTM*) [%]				
	1 s	5 s	10 s	30 s	60 s
CTM 30 s	67,88	93,27	96,92	100,00	100,00
CTM 60 s	69,43	93,36	97,48	100,00	100,00
DTM 30 s	62,42	88,45	94,27	98,27	98,18
DTM 60 s	61,06	86,91	92,59	96,10	98,18
CTM 90 s & *DTM* 90 s	66,35	91,76	97,37	100,00	100,00

Tabelle 4.2: *CHIL* Datenbasis: Identifikation von Sprechern mit Nahbereichsmikrophonen (*CTM*)

In Tab. 4.2 sind die Klassifikationsraten für Nahbereichsaufnahmen für ein Training mit wahlweise entfernten oder lokalen Mikrophondaten aufgeführt. Zum Vergleich sind in Tab. 4.3 die Klassifikationsraten für entfernte Mikrophondaten angegeben. Diese Aufnahmen sind für die beabsichtigte Anwendung aussagekräftiger als die Nahbereichsaufnahmen, da im Rahmen dieser Arbeit innerhalb der akustischen Szenenanalyse nur mit entfernten Mikrophongruppen gearbeitet wird.

Die Steigerung der Trainingsdatenmenge von 30 s auf 60 s reduziert die mittlere Fehlerrate bei gleichen Trainings- und Testbedingungen. Bei unterschiedlichen Trainings- und Testbedingungen sind die Ergebnisse nicht einheitlich. Eine Vergrößerung der Trainingsmenge (*DTM*) für die Klassifikation der Nahbereichsaufnahmen (*CTM*) verschlechtert die Ergebnisse geringfügig. Im Gegensatz dazu führt eine Vergrößerung der Trainingsmenge (*CTM*) zu einer signifikanten Verbesserung der Klassifikationsraten von entfernten Mikrophonsignalen (*DTM*). Die jeweils letzte Zeile der Tabellen 4.2 und 4.3 zeigt die Ergebnisse für ein *Multi-Condition*-Training, bei dem die gesamten Nah- und Fernbereichsdaten zu einem Trai-

Training \ Test	Klassifikationsrate (*DTM*) [%]				
	1 s	5 s	10 s	30 s	60 s
CTM 30 s	48,09	81,09	87,65	91,82	90,91
CTM 60 s	49,00	87,54	96,47	100,00	100,00
DTM 30 s	46,73	86,36	95,29	100,00	100,00
DTM 60 s	47,45	88,12	95,29	99,09	100,00
CTM 90 s & *DTM* 90 s	50,18	87,34	96,6	100,00	100,00

Tabelle 4.3: *CHIL* Datenbasis: Identifikation von Sprechern mit Fernfeldmikrophonen (*DTM*)

ningsdatensatz zusammengefasst werden. Diese Kombination ermöglicht gute Erkennungsergebnisse für beide Testdatensätze, da sie sowohl die Charakteristiken der Nahbereichsmikrophone als auch der Fernfeldmikrophone trainiert.

Nachdem die Systemkomponenten der Segmentierung und der Sprecheridentifikation einzeln validiert wurden, wird als nächstes die Fusion von Merkmalen zur Sprecheridentifikation in einigen Experimenten untersucht, bevor die Sprecherprotokollierung näher betrachtet wird.

Experimente zur Gewichtung von Merkmalen

Die Fusion von Merkmalsvektoren oder deren *Likelihoods* ermöglicht eine Reduktion der Fehlerrate bei der Sprecheridentifikation, wie in [KHF04] gezeigt wurde. Hierzu wird der Merkmalsvektor $\boldsymbol{x}^{sid}$ in die drei Komponenten

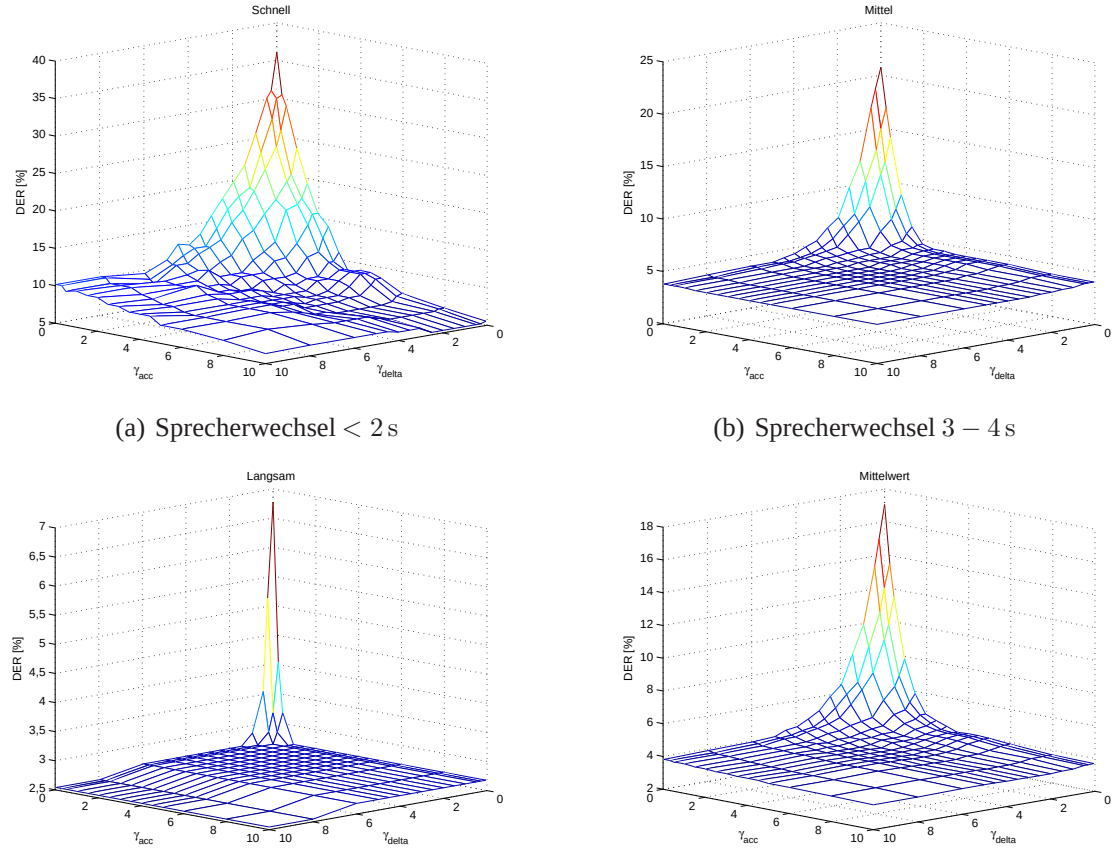
1. $\boldsymbol{x}_{M}^{sid}(k)$: *MFCC*-Merkmalsvektor und *MACV*-Wert
2. $\boldsymbol{x}_{\Delta M}^{sid}(k)$: 1. Ableitung der *MFCC*- und *MACV*-Werte
3. $\boldsymbol{x}_{\Delta\Delta M}^{sid}(k)$: 2. Ableitung der *MFCC*- und *MACV*-Werte

aufgeteilt. Diese Aufteilung ist möglich, da diagonale Kovarianzmatrizen im Verlauf des Trainings geschätzt werden. Experimentell soll eine Gewichtung der drei *Likelihood*-Werte (engl. *score level fusion*) untereinander mit

$$\begin{aligned}\log\tilde{\Lambda}\left(\boldsymbol{x}^{sid}(k)|\Omega=i\right) =& 1\cdot\log\Lambda\left(\boldsymbol{x}_{M}^{sid}(k)|\Omega=i\right)+\gamma_{delta}\cdot\log\Lambda\left(\boldsymbol{x}_{\Delta M}^{sid}(k)|\Omega=i\right)\\ &+\gamma_{acc}\cdot\log\Lambda\left(\boldsymbol{x}_{\Delta\Delta M}^{sid}(k)|\Omega=i\right)\end{aligned} \tag{4.75}$$

vorgenommen werden. Je größer die Werte γ_{delta} und γ_{acc} werden, desto weniger werden die *Likelihood*-Werte der *MFCC* berücksichtigt. Umgekehrt bedeuten die Extremwerte $\gamma_{delta} = \gamma_{acc} = 0$, dass die Ableitungen vernachlässigt werden.

In Abb. 4.16 sind die experimentellen Ergebnisse für die Sprecherwechselraten (schnell, mittel, langsam) und dem Mittelwert über alle Sprecherwechselraten angegeben. Deutlich erkennbar ist der Anstieg der Fehlerraten für alle Sprecherwechselraten bei Vernachlässigung der Ableitungen. Dieser ist umso ausgeprägter, je kleiner die durchschnittliche Segmentdauer ist. Der Mittelwert aller Segmentdauern zeigt ein schwach ausgeprägtes Minimum für die Gewichtung $\gamma_{delta} \approx 2$ und $\gamma_{acc} \approx 2$. Somit kann experimentell gezeigt werden, dass die zeitlichen Ableitungen der Merkmalsvektoren einen entscheidenden Beitrag zur Reduktion der Fehlerrate leisten.



(a) Sprecherwechsel $< 2\,\mathrm{s}$

(b) Sprecherwechsel $3 - 4\,\mathrm{s}$

(c) Sprecherwechsel $> 4\,\mathrm{s}$

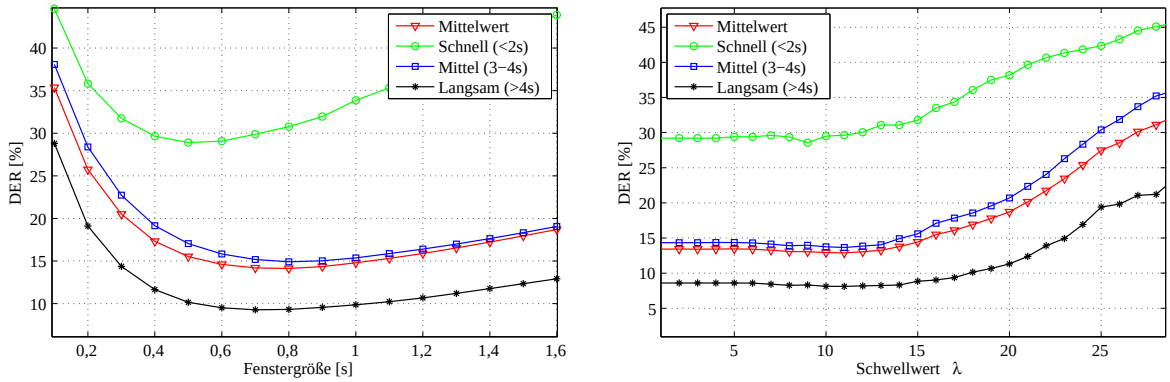
(d) Mittelwert

Abbildung 4.16: Vergleich der Fehlerraten für unterschiedliche Gewichtungen der Merkmalsvektorkomponenten

Sprecherprotokollierung

Der in Kap. 4.3.2 vorgestellte Ansatz zur Sprecherprotokollierung führt eine gleichzeitige Segmentierung und Identifikation von Sprechern in einem Datenstrom durch. Um die Leistungsfähigkeit des Ansatzes zu zeigen, werden zunächst Versuche mit zwei Standardverfahren („Gleitendes Fenster“ und „Segmentierung mit ΔBIC“) durchgeführt.

In Abb. 4.17 (a) sind die Ergebnisse für die Verwendung eines über den Datenstrom gleitenden Fensters konstanter Länge (engl. *sliding window*) gegeben. Hierbei wird ein Fenster von Merkmalsvektoren aus dem Datenstrom betrachtet und der wahrscheinlichste Sprecher ermittelt. Obwohl keine Informationen über Sprecherwechsel oder Sprecherpositionen verwendet werden, können mit diesem Verfahren brauchbare Ergebnisse erzielt werden. Deutlich zu erkennen ist, dass bei steigender Fenstergröße zunächst die Fehlerrate sinkt und jeweils abhängig von der Sprecherwechselrate anschließend wieder steigt. Es existiert kein gemeinsames Minimum für die unterschiedlichen Sprecherwechselraten, da ein größeres Fenster zwar eine sicherere Entscheidung des Sprechers ermöglicht, jedoch bei einer schnellen Abfolge der Sprecherwechsel mehrere Sprecher in einem Fenster vorhanden sein können und dadurch mehr Fehlentscheidungen entstehen. Aus diesem Grund werden im Folgenden immer die Mittelwerte der Fehlerraten (*DER*) für alle Sprachsegmentdauern als Vergleichskri-



(a) Gleitendes Fenster (b) ΔBIC-Segmentierung

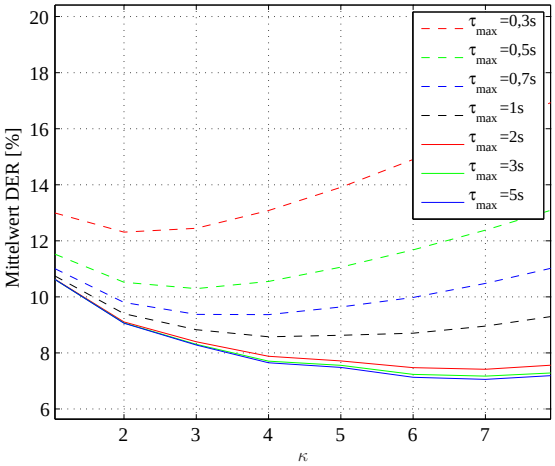
Abbildung 4.17: Ergebnisse der Sprecherprotokollierung durch ein gleitendes Fenster und eine ΔBIC-Segmentierung

terium verwendet. Die optimalen Parameter ergeben sich durch das Minimum der mittleren Fehlerrate.

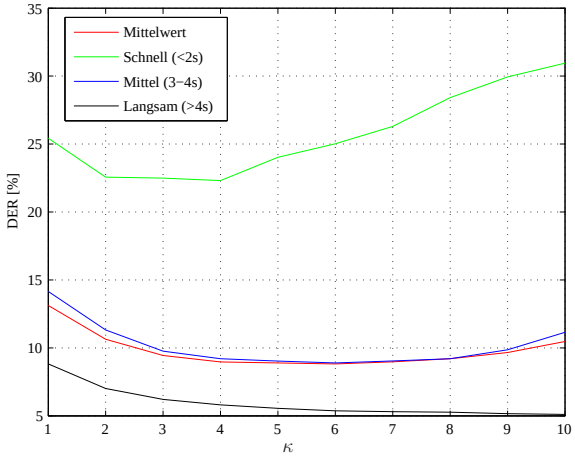
Im zweiten Verfahren werden die Informationen der Sprecherwechseldetektion aus der Berechnung der ΔBIC-Werte verwendet, um eine Segmentierung des Datenstroms durchzuführen. Anschließend werden die Segmente durch die Sprecheridentifikation einem Sprechermodell zugeordnet. Der ΔBIC-Schwellwert λ beeinflusst maßgeblich die Anzahl der gefundenen Segmentierungspunkte (vgl. Abb. 4.7, S. 29). Ein niedriger Wert von λ führt zu einer hohen Anzahl von Fehlalarmen und somit zu einer Zerstückelung von homogenen Sprachsegmenten. Diese falschen Segmentierungspunkte können durch die Sprecheridentifikation kompensiert werden, falls die Segmentgrößen nicht zu klein sind. Es ist in Abb. 4.17 (b) zu erkennen, dass mit steigendem Schwellwert λ die Fehlerrate ansteigt, da eine Vielzahl von Segmentierungspunkten nicht mehr erkannt werden.

Das vorgestellte Verfahren zur Sprecherprotokollierung verwendet einen Viterbi-Dekodierer mit einem *Partial Traceback*. Entsprechend der Gl. 4.71 (S. 38) des Viterbi-Dekodierers wird das Verfahren durch den Parameter κ zur Gewichtung der Emissionswahrscheinlichkeiten gegenüber den Transitionswahrscheinlichkeiten beeinflusst. Zusätzlich führt die Begrenzung der maximalen Latenz τ_{max} zu einem Anstieg der Fehlerrate.

In Abb. 4.18 (a) ist der Einfluss der zeitlichen Begrenzung des *Partial Traceback* auf eine maximale Latenz von $\tau_{\max}$ Sekunden bezogen auf die Konstante κ dargestellt. Es ist erkennbar, dass der Gewichtungsfaktor κ und die maximale Latenz $\tau_{\max}$ beide signifikant die Ergebnisse der Klassifikation beeinflussen und dabei voneinander abhängig sind. Aus der Abb. 4.18 (b) kann der Einfluss des Parameters κ auf die Sprecherprotokollierung abgelesen werden. Ein großer Wert des Parameters ist vorteilhaft für mittlere und lange Sprachsegmentdauern, da ein Verharren in einem Zustand unterstützt wird. Für schnelle Sprecherwechsel jedoch ist eine zu starke Gewichtung nachteilig und führt zu einer Erhöhung der Fehlerrate durch unterdrückte Sprecherwechsel. Da innerhalb der Datenbasis insgesamt mehr Daten für langsame und mittlere Sprecherwechsel als für schnelle Sprecherwechsel vorliegen, wird für den minimalen mittleren Fehler ein κ im Bereich des Optimums für mittlere Sprachsegmentdauern gewählt. Dies dürfte dem normalen Verlauf eines Gesprächs nahekommen und somit dem beabsichtigten Anwendungsbereich Rechnung tragen.



(a) Einfluss des Parameters κ und der zeitlichen Begrenzung $\tau_{\max}$ auf die mittlere Fehlerrate der Sprecherprotokollierung



(b) Einfluss des Parameters κ auf die Fehlerrate der Sprecherprotokollierung bezogen auf die Segmentdauern

Abbildung 4.18: Sprecherprotokollierung mittels Viterbi-Dekodierer unter Verwendung von Positionsdaten und ΔBIC-Werten

Verfahren \ Segmentdauer	DER [%]			
	$< 2\ s$	$3-4\ s$	$> 4\ s$	Mittelwert
Gleitendes Fenster	29,00	15,14	9,10	14,21
ΔBIC-Segmentierung	28,76	13,91	7,94	12,98
Viterbi (Position, ΔBIC, $\kappa = 1$)	22,62	11,52	6,83	10,69
Viterbi (Statisch, $\kappa = 5$)	25,53	10,05	5,72	9,66
Viterbi (Position, $\kappa = 7$)	21,66	9,32	5,69	8,95
Viterbi (ΔBIC, $\kappa = 7$)	24,03	9,48	5,35	9,08
Viterbi (Position, ΔBIC, $\kappa = 7$)	22,80	6,80	4,27	7,05
Perfekte Sprecherwechseldetektion	11,09	4,05	2,46	4,00

Tabelle 4.4: Vergleich der Verfahren zur Sprecherprotokollierung anhand der *DER*

In Tab. 4.4 sind die Ergebnisse der Sprecherprotokollierung für unterschiedliche Verfahren gegenübergestellt. Die schlechtesten Ergebnisse erzielt das Verfahren des gleitenden Fensters, da es keine Informationen über Sprecherwechsel in die Klassifikation oder Segmentierung mit einbezieht. Die Ausnutzung von Segmentierungspunkten aus der ΔBIC-Segmentierung verbessert demgegenüber die Ergebnisse. Ein Viterbi-Dekodierer mit einer geschätzten Transitionsmatrix aus Positionsdaten und ΔBIC-Werten übertrifft die reine ΔBIC-Segmentierung, jedoch führt die fehlende Glättung ($\kappa = 1$) zu Oszillationen zwischen den Zuständen, was die Ergebnisse negativ beeinflusst. Zum Vergleich ist ein Viterbi-Dekodierer mit einer statischen Transitionsmatrix und einem optimalen Gewichtungsfaktor κ untersucht worden. Dieser Ansatz liefert eine mittlere Fehlerrate von $9{,}66\,\%$, wobei jedoch die Verwendung von Positionsdaten (*DER* $8{,}95\,\%$) oder Sprecherwechselinformationen (*DER* $9{,}08\,\%$) zur Schätzung der Transitionsmatrix geringere Fehlerraten erzielen. Kombiniert man alle Informationen (Position, ΔBIC, $\kappa = 7$), so kann eine mittlere Fehlerrate von $7{,}05\,\%$ erreicht werden. Als unterste Grenze ist die Fehlerrate für eine perfekte Segmentierung angegeben, welche die Leistungsfähigkeit der Sprecheridentifikation zeigt.

4.4 Audio-visuelle Sprecherprotokollierung

Das bisher vorgestellte Verfahren zur Sprecherprotokollierung verwendet ausschließlich Informationen, welche aus akustischen Aufnahmen gewonnen wurden. Da die Sprecherprotokollierung in einem System zur ambienten Kommunikation verwendet werden soll, kann eine neue Informationsquelle in Form der Videodaten erschlossen werden. Im Folgenden soll zunächst ein Überblick über das Verfahren zur Gesichtsdetektion und Identifikation gegeben werden, bevor die Integration in den Prozess der Sprecherprotokollierung diskutiert wird.

4.4.1 System zur Gesichtsidentifikation

Aus der Literatur sind eine Reihe von Ansätzen zur Detektion und Identifikation von Gesichtern bekannt [YKA02]. Je nach Anwendungsgebiet und damit Anforderungen an die Erkennungsgenauigkeit werden unterschiedlich aufwändige Verfahren eingesetzt. Gerade die Detektion und Identifikation von Gesichtern bei schlechter Beleuchtung oder ungünstigen Aufnahmewinkeln erfordert komplexe Ansätze. Da man im Falle einer Kommunikation jedoch von einem kooperativen Benutzer ausgehen kann, soll an dieser Stelle der Standardansatz nach [VJ01] zum Auffinden von aufrechten Gesichtern in Bildern verwendet werden. Benutzer werden in diesem Zusammenhang als „kooperativ“ bezeichnet, da sie im Falle einer Kommunikation meistens den Augenkontakt zum Gesprächspartner suchen und somit in Richtung der Kamera schauen, die oberhalb des Displays angebracht ist. Die Beleuchtungssituation kann als unproblematisch angenommen werden, da ansonsten bei einer schlechten Beleuchtung das Gesicht für den entfernten Gesprächspartner nicht erkennbar wäre. Die Identifikation der detektierten Gesichter erfolgt durch die *Fisher-Faces*-Methode aus [BHK97].

4.4.2 Gesichtsdetektion

Die Anbindung der Kamera erfolgt entweder über einen USB-Anschluss, oder im Falle der in den Versuchen verwendeten Kamera über eine Ethernet-Schnittstelle. Abbildung 4.19 zeigt

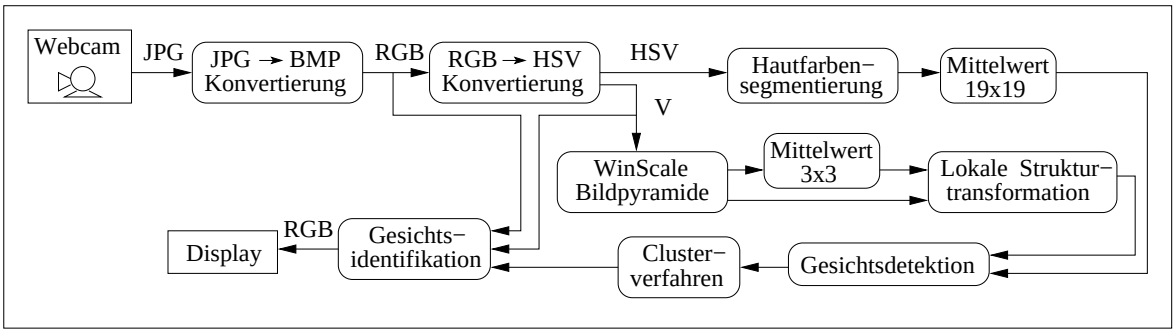


Abbildung 4.19: Blockschaltbild zur Gesichtsdetektion und Gesichtsidentifikation

die notwendigen Module zur Detektion von Gesichtern und anschließender Identifikation. Die von der Kamera gesendeten Bilder werden zunächst vom *JPG*-Format in das *BMP*-Format konvertiert. Im nächsten Schritt wird das Bild in den *HSV*-Farbraum konvertiert, da in diesem eine Hautfarbensegmentierung mit geringem Aufwand durchgeführt werden kann.

Die Hautfarbensegmentierung dient der Begrenzung des Bildausschnittes, der für die Suche nach Gesichtern im Gesichtsdetektor herangezogen wird. Parallel dazu wird das Bild in mehreren Stufen zu einer Bildpyramide skaliert und deren Teilbilder durch eine Strukturtransformation umgewandelt. Die einzelnen Module und ihre Aufgaben werden im Folgenden näher betrachtet.

Hautfarbensegmentierung

Die Hautfarbensegmentierung verwendet ein Histogramms zur Bestimmung der Wahrscheinlichkeit für Hautfarbe in einem Bildpunkt. Das entstehende zweidimensionale Bild (vgl. Abb.

(a) Kamerabild

(b) Hautfarbenwahrscheinlichkeit nach Histogramm

(c) Gemittelte Hautfarbenwahrscheinlichkeit

(d) Hautfarbengebiete nach Schwellwertentscheidung

Abbildung 4.20: Beispiel einer Hautfarbensegmentierung mit Schwellwertentscheidung

4.20 (b)) enthält zunächst durch Bildrauschen und den Schattenwurf im Gesicht nur wenige zusammenhängende Flächen, die als Haut erkannt wurden. Durch die Mittelwertbildung auf 19×19 Bildpunkten (vgl. Abb. 4.20 (c)) großen Flächen und einer Schwellwertentscheidung (vgl. Abb. 4.20 (d)) werden diese Gebiete vergrößert. Die entstehenden Gebiete definieren den Suchbereich für die Detektion von Gesichtern. Durch die Hautfarbensegmentierung ist es möglich, die Anforderungen an die Rechenleistung zu senken und gleichzeitig die Rate von Fehldetektionen zu reduzieren, da Strukturen im Hintergrund ohne Hautfarbe nicht mehr fälschlicherweise als Gesicht detektiert werden können.

Skalierung und Suche

Das Auffinden von Gesichtern unterschiedlicher Größe in Bildern kann auf zwei Arten erfolgen. Zum einen kann ein Detektor auf eine bestimmte Gesichtsgröße trainiert und das Bild in verschiedene Stufen skaliert werden, oder aber der Detektor selbst wird skaliert und das Bild beibehalten. In diesem System wird das Bild in 15 Stufen skaliert, und es wird in jeder Stufe nach Gesichtern der Größe 19×19 Bildpunkte gesucht. Ein Gesicht, welches in keiner der Skalierungsstufen des Bildes annähernd die Größe 19×19 Bildpunkte erreicht, kann nicht erkannt werden.

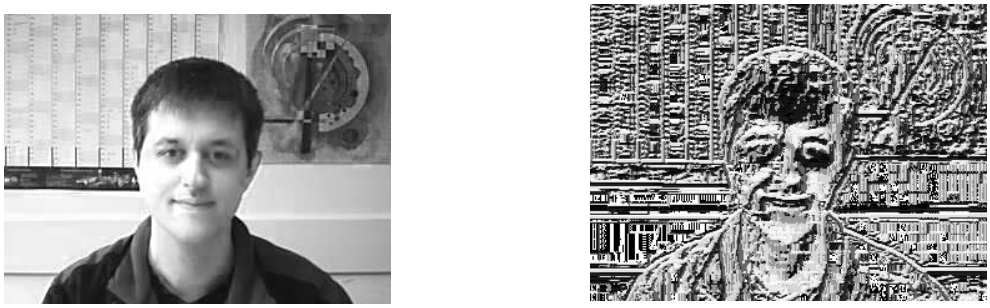
Abbildung 4.21: Beispiel einer Bildpyramide mit 8 Skalierungsstufen

Abbildung 4.21 zeigt die ersten 8 Bilder der Bildpyramide, die durch die Skalierung des Graustufenbildes (V-Komponente des Originalbildes) entstehen. Die Skalierung des Bildes erfolgt durch den in [KSLK03] vorgestellten *WinScale*-Algorithmus, der am Ausgang des Moduls die komplette Bildpyramide aller Skalierungsstufen liefert.

Für jedes skalierte Bild in der Pyramide wird eine lokale Strukturtransformation (LST) nach [FK04] durchgeführt. Die Transformation verwendet binäre 3×3 Kernel zur Kodierung der lokalen Strukturinformation. Zunächst wird der mittlere Helligkeitswert der 3×3 Umgebung eines Pixels berechnet und jedes Pixel mit diesem verglichen. Falls der Helligkeitswert des Pixels über dem Mittelwert liegt, so wird eine 1 im Kernel gesetzt ansonsten eine 0. Somit entstehen insgesamt $2^9 - 1 = 511$ unterschiedliche Kernel, deren binäre Kodierungen als Zahlen interpretiert werden.

In Abb. 4.22 (a) ist das Graustufenbild und in Abb. 4.22 (b) das zugehörige Bild der lokalen Strukturtransformation zu sehen. Deutlich erkennbar ist, dass die Transformation die Strukturen im Bild, wie z. B. Kanten und Konturen, hervorhebt und gleichzeitig die Helligkeitsunterschiede vernachlässigt.

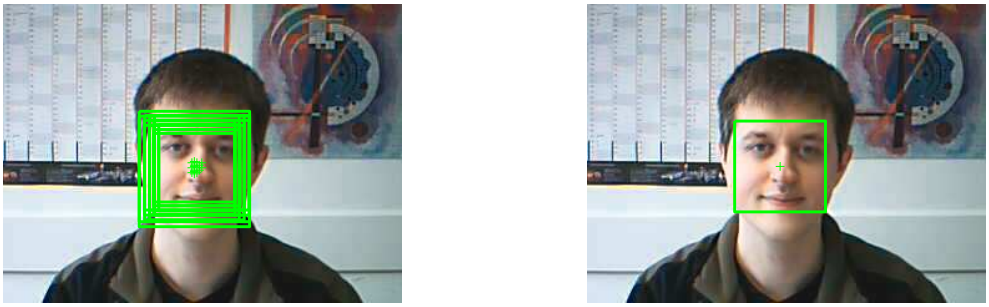
Der Gesichtsdetektor besteht, wie in [VJ01] vorgeschlagen, aus einer 4-stufigen Kaskade von Entscheidern mit zunehmender Komplexität. Dabei wird ein Analysefenster der Größe 19×19 Pixel über das Bild geschoben. Innerhalb dieses Fensters liegen $17^2 = 289$ LST Merkmale, von denen in jeder Stufe eine größer werdende Anzahl überprüft wird. Der Fokus der Detektoren liegt hierbei auf dem Verwerfen von „Nicht-Gesichtern“, so dass in den ersten Stufen der Großteil der Analysefenster verworfen werden kann und nur Fenster mit möglichen Gesichtern an die nächste, aufwändigere Stufe weitergereicht werden. Die Detektoren der Kaskade werden in Anlehnung an [KE06] mittels eines *AdaBoost*-Algorithmus [DHS01] trainiert, jedoch werden im Gegensatz zum dortigen Vorschlag nicht nur die ersten



(a) Graustufenbild (b) Ergebnis der lokalen Strukturtransformation

Abbildung 4.22: Merkmalsextraktion mittels lokaler Strukturtransformation des Graustufenbildes

drei Stufen, sondern alle vier Stufen mit dem *AdaBoost*-Algorithmus trainiert.



(a) Mehrfachdetektion eines Gesichtes (b) Detektion nach Clusterung

Abbildung 4.23: Beispiel einer Mehrfachdetektion eines Gesichtes und Ergebnis der Clusterung

Ein Gesicht wird zumeist nicht nur in einer Skalierungsstufe eines Bildes, sondern auch in der nächst höheren oder niedrigeren Skalierungsstufe gefunden. Zudem werden auch Detektionen, die nur um einige wenige Pixel verschoben sind, von der Kaskade als gefundene Gesichter ausgegeben. In Abb. 4.23 (a) wurden die detektierten Gesichter mit grünen Kästen umrandet und deren Zentren mit grünen Kreuzen markiert. In dem Beispielbild wird das Gesicht insgesamt 16 mal gefunden, und erst eine Clusterung der Detektionen liefert eine Aussage über die tatsächliche Anzahl der Gesichter im Bild. Die Clusterung wurde als Mittelwert über die Detektionen berechnet, und das Ergebnis der Clusterung ist in Abb. 4.23 (b) gegeben. Dabei werden für die Mittelung nur übereinander liegende Detektionen verwendet, so dass auch die Detektion von mehreren Gesichtern in einem Bild möglich ist. Die Information über detektierte Gesichter wird anschließend dem Modul zur Identifikation übergeben, so dass eine Zuordnung zu den bekannten Gesichtern erfolgen kann.

4.4.3 Gesichtsidentifikation

Die Gesichtsidentifikation verwendet die Detektionen aus dem vorherigen Modul, um die zu untersuchenden Bereiche des Bildes zu extrahieren und unter Verwendung der *Fisher-Faces*-

Methode aus [BHK97] zu identifizieren. Die Detektion eines Gesichtes kann zuverlässig auf einer Größe von 19×19 Pixeln erfolgen, jedoch ist dies für eine Identifikation der Person nicht ausreichend. Experimente haben gezeigt, dass für eine Identifikation das Gesicht eine Mindestgröße von 60×60 Pixeln haben sollte. Da die Gesichtsdetektion einen sehr knappen Ausschnitt des Gesichtes markiert, der oben mit den Augen und unten mit dem Mund abschließt, ist für eine Identifikation eine Erweiterung der ermittelten Gesichtsgrenzen notwendig. Dabei werden die zuvor in grün markierten Bereiche (vgl. Abb. 4.23 (b)) in jede Richtung um ca. $20\,\%$ gestreckt und der entstehende Ausschnitt so interpoliert, dass eine Auflösung von 60×60 Pixeln erzielt wird. Sollte für eine Identifikation eines Gesichtes nicht die erforderliche Menge an Pixeln zur Verfügung stehen, weil zum Beispiel die Detektion in einer der kleinsten Stufen der Bildpyramide erfolgt ist, so wird die Detektion als unbekannte Person vermerkt.

Es wird im Folgenden angenommen, dass ein Gesicht im Bild detektiert wird, das aus der Gruppe der $\mathcal{I}$ bekannten Benutzer stammt. Die Identifikation des 60×60 Pixel großen Gesichtes erfolgt in zwei Schritten. Im ersten Schritt wird der durch die Detektion definierte Bereich zeilenweise aus dem Graustufenbild ausgelesen und als Vektor $\mathbf{\Gamma}(k)$ mit 3600 Dimensionen interpretiert. Auf diesem wird mit Hilfe einer Transformationsmatrix $\boldsymbol{P}$, die auf Trainingsdaten mit einer Hauptachsentransformation (engl. *Principal Component Analysis*, *PCA*) geschätzt wurde, eine Dimensionsreduktion durchgeführt. Dies kann interpretiert werden als Reduktion der vorhandenen Bildinformationen auf die für ein Gesicht relevanten Informationen. Im zweiten Schritt wird eine Transformationsmatrix $\boldsymbol{L}$ angewendet, die durch eine lineare Diskriminanzanalyse (LDA) auf den annotierten Gesichtern der Benutzer geschätzt wurde. Diese reduziert die Dimension des Vektors auf $\mathcal{I} - 1$ Dimensionen, also auf die Anzahl der bekannten Benutzer minus Eins. Der dimensionsreduzierte Vektor der Detektion ergibt sich folglich zu:

$$\boldsymbol{x}^{vid}(k) = \boldsymbol{L}^T \cdot (\boldsymbol{P}^T \cdot (\mathbf{\Gamma}(k) - \boldsymbol{m}_{PCA}) - \boldsymbol{m}_{LDA}) \tag{4.76}$$

Hierbei bezeichnen $\boldsymbol{m}_{PCA}$ und $\boldsymbol{m}_{LDA}$ die Mittelwertvektoren der Trainingsdaten vor der *PCA* bzw. der LDA.

Das Problem der Sprecherprotokollierung wird, wie zuvor beschrieben, durch einen stochastischen Ansatz gelöst, wobei die Sequenz der Merkmalsvektoren als Realisierung eines Zufallsprozesses interpretiert wird. Dies wird entsprechend für die visuellen Merkmalsvektoren umgesetzt, indem die Dichtefunktionen $p(\boldsymbol{x}^{vid}(k)|\Omega = i)$, $i = 1, \ldots, \mathcal{I}$, bestehend aus jeweils einer Normalverteilung, aus Trainingsdaten geschätzt werden. Die Klassifikationsrate des Systems kann durch die Verknüpfung von aufeinander folgenden Beobachtungen, welche aus dem gleichen Kamerawinkel stammen, verbessert werden. Hierfür werden die a posteriori Wahrscheinlichkeiten eines Gesichtes des letzten Zeitschritts als a priori Wahrscheinlichkeiten des aktuellen Zeitschritts verwendet. Dabei bezeichnet $\boldsymbol{x}^{vid}_{\nu:k} = [\boldsymbol{x}^{vid}(\nu), \ldots, \boldsymbol{x}^{vid}(k)]$ die Merkmalsvektoren von Zeitschritt $(k - \nu + 1)$ bis zum Zeitschritt k. Unter der Annahme von unabhängigen und identisch verteilten Beobachtungen folgt für die a posteriori Wahrscheinlichkeiten:

$$P(\Omega = i|\boldsymbol{x}^{vid}_{\nu:k}) = \frac{p(\boldsymbol{x}^{vid}(k)|\Omega = i)P(\Omega = i|\boldsymbol{x}^{vid}_{\nu:k-1})}{\sum_j p(\boldsymbol{x}^{vid}(k)|\Omega = j)P(\Omega = j|\boldsymbol{x}^{vid}_{\nu:k-1})}. \tag{4.77}$$

Die Rekursion startet zum Zeitpunkt ν, an dem zum ersten Mal ein Gesicht an einer bestimmten Position detektiert wird. Startwerte für die Rekursion sind die a priori Wahrschein-

lichkeiten $P(\Omega = i)$, die auf $1/\mathcal{I}$ gesetzt werden. Nach einer erfolgten Identifikation werden die a posteriori Wahrscheinlichkeiten als a priori Wahrscheinlichkeiten für die Identifikation von Gesichtern im nächsten Bild verwendet. Dafür wird das Bild in Kacheln eingeteilt und es werden für jede Kachel, die vom Gesicht überdeckt wird, die Werte der a posteriori Wahrscheinlichkeiten der Klassen abgespeichert. Somit profitiert die Gesichtsidentifikation von den vorherigen Beobachtungen. Sollte innerhalb einer Kachel keine Detektion vorliegen, so werden die gespeicherten Wahrscheinlichkeiten schrittweise auf die Initialisierungswerte zurückgeführt.

Die Zusammenführung der akustischen und visuellen Beobachtungen bedingt, dass die Beobachtungen von einem Benutzer stammen und nicht durch unterschiedliche Benutzer hervorgerufen werden. Sollte die Kamera einen Benutzer fokussieren und die Gesichtsidentifikation ihn identifizieren, so wäre es für die Sprecherprotokollierung von Nachteil, wenn dieser Benutzer nicht der aktuelle Sprecher ist. Dieses Problem kann durch den Einsatz einer schwenkbaren Kamera gelöst werden, in dem die Kamera immer auf den aktuellen Sprecher fokussiert wird.

4.4.4 Kamerasteuerung und Systemintegration

Die Steuerung der Kamera erfolgt unter Berücksichtigung der Positionsschätzungen der akustischen Szenenanalyse und den detektierten Gesichtern des zuvor vorgestellten Systems zur Gesichtsidentifikation. In Abb. 4.24 ist das Blockschaltbild zur Kamerasteuerung und audio-visuellen Sprecherprotokollierung gegeben. Das Videosystem, welches im oberen Teil dargestellt ist, beinhaltet neben dem System zur Detektion und Identifikation von Gesichtern zwei weitere Module. Das Modul *SHM* verwaltet einen gemeinsamen Speicherbereich (engl. *Shared Memory*, *SHM*) und ist verantwortlich für den Datenaustausch mit dem Audiosystem. Das Modul *PTZ Control* steuert die Kamera über eine *TCP*/*IP*-Schnittstelle und ist somit verantwortlich für die Ausrichtung der Kamera. Hierzu fordert das Modul regelmäßig die Informationen über die Fokussierung der Kamera an und berechnet die Differenz zu den durch die akustische Positionsschätzung vorgegebenen Werten. Ist die Differenz zwischen der akustischen Positionsschätzung und dem aktuellen Kamerablickwinkel so groß, dass der Sprecher außerhalb des Bildes liegt, so wird die Fokussierung der Kamera auf den Sprecher durchgeführt. Zusätzlich verwendet das Modul die Positions- und Größeninformationen von detektierten Gesichtern im Bild, um die Fokussierung auf die Personen zu optimieren. Der untere Teil der Abb. 4.24 zeigt das Audiosystem zur Sprecherprotokollierung und Sprecherlokalisation, wie es für den in Abb. 4.25 gezeigten experimentellen Aufbau verwendet wird.

Das Audiosystem verwendet drei der vier Winkelschätzungen der adaptiven Strahlformung zur Positionsschätzung mittels Schnittpunktanalyse. Der vierte Winkel ist ein Neigungswinkel, welcher ausschließlich für die Ausrichtung der Kamera verwendet wird. Das Modul „Koordinatentransformation“ berechnet, basierend auf den Positionsdaten der Kamera und der geschätzten Sprecherposition, die Schwenk- und Neigewinkel sowie den Zoomfaktor der Kamera zur Fokussierung des aktuellen Sprechers. Diese Daten werden über das Modul „*SHM*“ an die Kamerasteuerung weitergeleitet. Des Weiteren wird die Positionsschätzung im Rahmen der Sprecherprotokollierung entsprechend Kap. 4.3.2 zur Schätzung der Transitionsmatrix verwendet. Neben den Informationen der Sprecherbewertung, der Sprachaktivitätsdetektion und des Bayes’schen Informationskriteriums werden nun auch die Informationen der Gesichtsidentifikation in der Sprecherprotokollierung berücksichtigt.

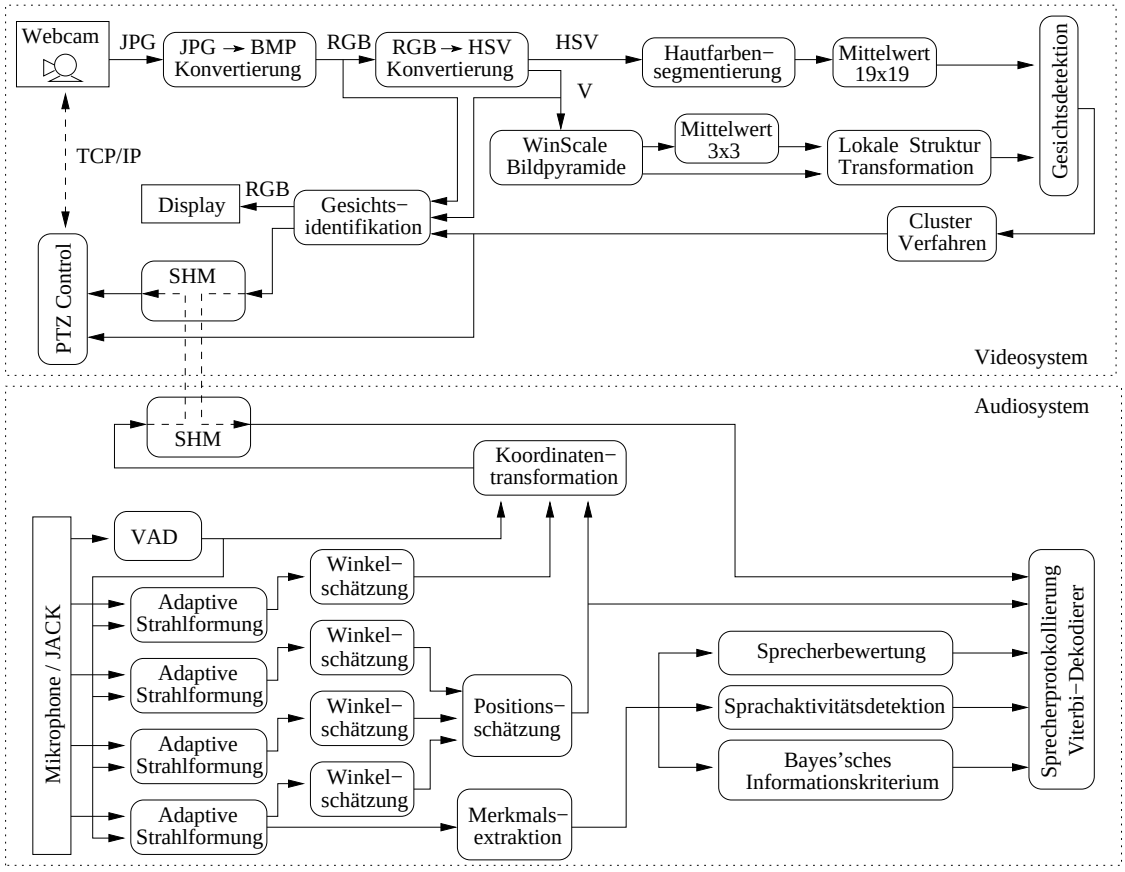


Abbildung 4.24: Blockschaltbild der Kombination von Kamerasteuerung und audio-visueller Sprecherprotokollierung

Synchronisation und Datenaustausch

Das Audiosystem arbeitet bei einer Abtastrate von $16\,\mathrm{kHz}$ und einer Blockgröße von 128 Abtastwerten mit einer konstanten Rate von $8\,\mathrm{ms}$ pro Block. Im Gegensatz dazu liefert die Kamera einen nicht kontinuierlichen Datenstrom von maximal 15 Bildern pro Sekunde, dessen Rate durch die Qualität des Netzwerks beeinflusst wird. Zusätzlich kann bedingt durch die nicht konstante Rechenlast der Gesichtsidentifikation ein sporadisches Verwerfen von Bildern durchgeführt werden, um die Belastung zu verringern. Da sowohl das Audio- als auch das Videosystem mit unterschiedlichen Datenraten arbeiten, muss eine Synchronisation erfolgen. Der hier verwendete Ansatz verzichtet auf die Annotation von Daten mit Zeitstempeln, um eine Synchronisation mittels Verzögerungen zu realisieren, zu Gunsten des Ansatzes, dass jeweils die aktuellen Daten in einem gemeinsamen Speicherbereich abgelegt werden. Diese Daten werden von dem jeweils anderen System so lange genutzt, bis sie durch aktuellere Daten überschrieben werden.

Experimenteller Aufbau

Der experimentelle Aufbau zur audio-visuellen Sprecherprotokollierung beinhaltet neben den drei Mikrophongruppen zur Lokalisierung des Sprechers (Gruppe_1-Gruppe_3) auch eine schwenkbare Kamera und einen Monitor. Mit jeder der drei Mikrophongruppen wird ein

Winkel $\overline{\alpha}_i$ in Richtung des Sprechers nach Gl. 4.21 geschätzt. Hieraus ergeben sich die drei Schnittpunkte $\boldsymbol{\chi}_{12}, \boldsymbol{\chi}_{23}$ und $\boldsymbol{\chi}_{13}$ deren Schwerpunkt als Positionsschätzung verwendet wird (vgl. Gl. 4.23, S. 22). Die Mikrophongruppe unterhalb der Kamera besitzt einen T-förmigen Aufbau, der die Schätzung eines Neigungswinkel β ermöglicht. Da die Kamera in den drei Koordinaten Drehwinkel, Neigungswinkel und Zoomstufe arbeitet, muss die Position des Sprechers von den kartesischen Koordinaten in einen Drehwinkel und eine Zoomstufe umgerechnet werden. Dies wird in dem Modul zur Koordinatentransformation im Audiosystem durchgeführt.

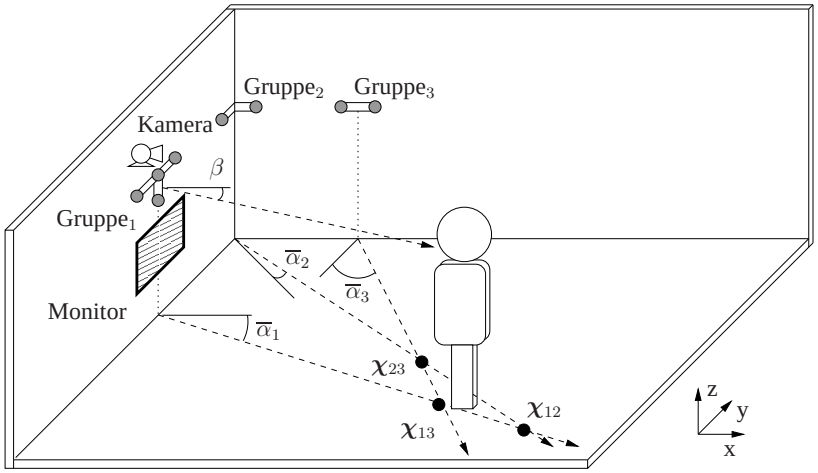


Abbildung 4.25: Experimenteller Aufbau zur ambienten Kommunikation und audio-visueller Sprecherprotokollierung

4.4.5 Integration der visuellen Information

Der in Kap. 4.3.2 vorgestellte Ansatz zur Sprecherprotokollierung verwendet ein *HMM*, dessen Emissionswahrscheinlichkeiten durch die *Likelihoods* der akustischen Merkmalsvektoren gegeben sind. An dieser Stelle wird die Berechnung der Emissionswahrscheinlichkeiten erweitert, so dass sowohl die *Likelihoods* der akustischen als auch der visuellen Merkmalsvektoren berücksichtigt werden. Die Emissionswahrscheinlichkeiten der *HMM*-Zustände sind nach Gl. 4.63 (S. 37) mit

$$b_j(\boldsymbol{x}^{sid}(k)) = p'(\boldsymbol{x}^{sid}(k)|\Omega = j) \tag{4.78}$$

gegeben. Unter der Annahme, dass die akustischen Merkmalsvektoren $\boldsymbol{x}^{sid}(k)$ und die visuellen Merkmalsvektoren $\boldsymbol{x}^{vid}_{\nu:k}$ statistisch unabhängig sind, werden die Emissionswahrscheinlichkeiten neu definiert zu:

$$\begin{aligned} b_j(\boldsymbol{x}^{sid}(k), \boldsymbol{x}^{vid}_{\nu:k}) &:= p(\boldsymbol{x}^{sid}(k), \boldsymbol{x}^{vid}_{\nu:k})|\Omega = j) && (4.79)\\ &= p'(\boldsymbol{x}^{sid}(k)|\Omega = j) \cdot p(\boldsymbol{x}^{vid}_{\nu:k}|\Omega = j) \\ &= p'(\boldsymbol{x}^{sid}(k)|\Omega = j) \cdot P(\Omega = j|\boldsymbol{x}^{vid}_{\nu:k}) \frac{p(\boldsymbol{x}^{vid}_{\nu:k})}{P(\Omega = j)}. && (4.80) \end{aligned}$$

Die Transitionswahrscheinlichkeiten des *HMM* werden wie zuvor über die Sprecherwechselinformationen der Positionsschätzung und den Δ*BIC*-Werten nach Gl. 4.70 (S. 37) geschätzt.

Die optimale Abfolge der Zustände gegeben die Beobachtungen wird durch einen Viterbi-Dekodierer bestimmt. Somit ist es gelungen, die Informationen aus dem Videosystem in das System der akustischen Sprecherprotokollierung zu integrieren, so dass ein System zur audio-visuellen Sprecherprotokollierung entsteht.

4.4.6 Experimentelle Ergebnisse

Das System der audio-visuellen Sprecherprotokollierung beinhaltet im Vergleich zu der akustischen Sprecherprotokollierung die dynamische Komponente der Kamera. Ein Test auf einer statischen Datenbasis ist somit nicht möglich, weil die aktuelle Schätzung der Position eines Sprechers auf den akustischen und visuellen Daten beruht, welche mit der Kamera und den Mikrophonen aufgenommen werden. Diese führen ihrerseits zu einer Anpassung des Kamerablickwinkels und folglich zu einer Änderung der Beobachtungen. Das System beeinflusst sich also während der Laufzeit selbst und kann nur im laufenden Betrieb getestet werden. Hierzu werden zwei typische Nutzungsszenarien ausgewählt und mit einer Gruppe von trainierten Sprechern untersucht. Die ersten Tests werden mit Einzelnutzern durchgeführt, die sich für den größten Teil der Aufnahmen an einem festen Ort des Raumes aufhalten. Vorteil dieses Szenarios ist es, dass die Kamera einen Großteil der Zeit eine gute Fokussierung auf das Gesicht besitzt. Das zweite Szenario betrachtet eine Konferenzsituation, bei der sich zwei Personen im Raum befinden und abwechselnd sprechen. Hierbei muss die Kamera die Fokussierung zwischen den Sprechern wechseln, wodurch vermehrt Phasen ausbleibender Gesichtsdetektionen entstehen.

Die so entstandene Menge von mehr als zwei Stunden Aufnahmen bietet zwar nicht die Möglichkeit, nachträglich Einfluss auf die Position oder Ausrichtung der Kamera zu nehmen, jedoch können bestimmte Aspekte der Sprecherprotokollierung untersucht werden. Zunächst wird der zeitliche Ablauf der Kamerasteuerung anhand eines Beispiels erläutert. Anschließend werden die Verzögerung des Systems und der Einfluss der zeitlichen Begrenzung näher betrachtet. Zum Abschluss der Experimente werden die Ergebnisse der audio-visuellen Sprecherprotokollierung diskutiert.

Kamerasteuerung

Zunächst soll ein Beispiel für das zeitliche Verhalten der Kamerasteuerung bei einem Sprecherwechsel gegeben werden. In Abb. 4.26 ist im unteren Teilbild die Positionsschätzung der akustischen Szenenanalyse in kartesischen Koordinaten gegeben. Im oberen Teilbild sind entsprechend Gl. 4.77 (S. 52) die a posteriori Wahrscheinlichkeiten der Nutzer auf Basis der Gesichtsidentifikation dargestellt. Im Zeitraum $0\,\mathrm{s}$ bis $4\,\mathrm{s}$ liefert das System wechselnde Hypothesen für die Identität des detektierten Gesichtes, sowie immer wieder Zeiträume in denen alle Modelle gleich wahrscheinlich sind und somit keine Gesichtsdetektion vorliegt. Dieses Verhalten kann verschiedene Gründe haben, wie z. B. Bewegungen des Sprechers, die Ausrichtung des Kopfes oder nicht optimale Beleuchtungsverhältnisse. Ab dem Zeitpunkt $8{,}5\,\mathrm{s}$ sind die Ergebnisse der Gesichtsidentifikation eindeutig, wie aus dem Verlauf der Kurven ersichtlich ist.

Deutlich ist die mit dem Sprecherwechsel verbundene Änderung der Position zum Zeitpunkt $3{,}9\,\mathrm{s}$ erkennbar. Zu diesem Zeitpunkt wird eine Sprecherposition außerhalb des Kamerablickwinkels detektiert und die Kamera beginnt mit dem Schwenk auf die neue Position

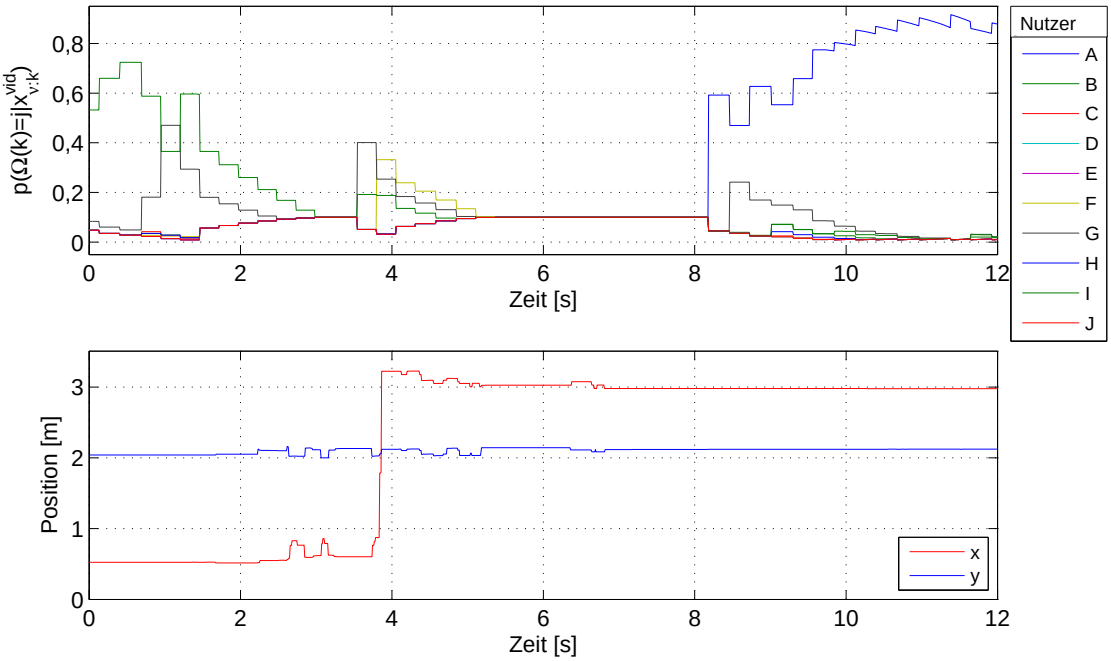


Abbildung 4.26: Vergleich zwischen den a posteriori Wahrscheinlichkeiten der Gesichtsidentifikation und der Positionsschätzung durch die akustische Szenenanalyse

und der anschließenden Fokussierung auf den Sprecher. Ab dem Zeitpunkt $8{,}2\,\mathrm{s}$ ist das Gesicht des Sprechers durch das System gefunden und identifiziert worden.

Der treppenförmige Verlauf der a posteriori Wahrscheinlichkeiten resultiert aus der im Vergleich zum Audiosignal niedrigeren Verarbeitungsrate des Videosystems. Die Abb. 4.26 wurde aus den eingehenden Daten der Sprecherprotokollierung gewonnen und enthält somit die im Takt des Audiosystems aufgezeichneten Signale. Da das Videosystem die aktuellen Daten in einer geringeren Rate als der Taktrate des Audiosystems im *SHM* ablegt, kommt es zu einer mehrfachen Nutzung der Daten durch das Audiosystem.

Systemverzögerung

Die zeitlichen Anforderungen von kontextbewussten Diensten, wie z. B. der ambienten Kommunikation, verlangen eine möglichst geringe Latenz zwischen dem Eintreten eines Ereignisses und der Benachrichtigung der Applikation durch das System. Die Sprecherprotokollierung als Kontextquelle im vernetzten Haus beinhaltet systembedingt einige Latenzen, die im Prozess der Signalverarbeitung entstehen. Folgende Verzögerungen sind im System vorhanden:

- Hardware/Software-Schnittstelle: Die Latenz beträgt bei einem echtzeitfähigen Betriebssystem bei einer Blockgröße von 128 Abtastwerten und einer Abtastfrequenz von $16\,\mathrm{kHz}$ im Minimum $8\,\mathrm{ms}$. Steht kein echtzeitfähiges Betriebssystem zur Verfügung ist eine Latenz von ca. 3 Blöcken und somit $24\,\mathrm{ms}$ realistisch.

- Positionsschätzung: Die Positionsschätzung ist frei von Latenzen, weil sie durch die

Korrelation der *FSB*-Filter berechnet wird. Jedoch benötigt die Ausrichtung der akustischen Strahlformung eine gewisse, deterministisch nicht bestimmbare Zeit, bis die korrekte Position nach Eintreten eines konvergierten Zustandes der Filter angezeigt wird. Da für die Sprecherprotokollierung weniger die korrekte Position, sondern vielmehr die Tatsache des Positionswechsels interessant ist, kann diese Latenz vernachlässigt werden.

- Sprecherwechseldetektion: Die Berechnung der ΔBIC-Werte erfordert die Betrachtung eines Zeitfensters der Größe $N_w = 80$ Merkmalsvektoren. Die Latenz beträgt folglich $N_w/2 \cdot 8\,\mathrm{ms} = 320\,\mathrm{ms}$.
- Viterbi-Dekodierer: Der Viterbi-Dekodierer besitzt eine variable Verzögerung, die durch die obere Grenze $\tau_{\max}$ zeitlich beschränkt ist.

Die variable Latenz des Viterbi-Dekodierers soll an dieser Stelle näher untersucht werden. Zunächst wird die zeitliche Begrenzung $\tau_{\max}$ weggelassen ($\tau_{\max} = \infty$), um eine Messung der tatsächlich vorliegenden Verzögerung durchführen zu können. In Abb. 4.27 sind die Er-

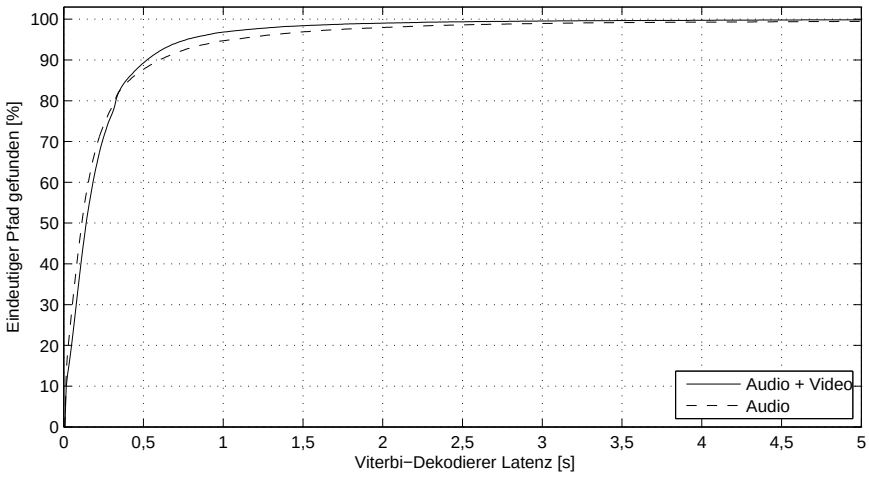


Abbildung 4.27: Experimente zur zeitlichen Verzögerung des Viterbi-Dekodierers

gebnisse des Experiments gegeben. Aufgetragen über die Latenz des Viterbi-Dekodierers (Abszisse) wird auf der Ordinate der Prozentsatz der Fälle angegeben, in denen ein eindeutiger Pfad innerhalb dieser Latenz gefunden wird. Hierbei kann festgestellt werden, dass in $90\,\%$ aller Fälle die Latenz geringer als $0{,}5\,\mathrm{s}$ ist. Die mittlere Latenz bis ein eindeutiger Pfad gefunden wird kann zu $262\,\mathrm{ms}$ für die akustische Sprecherprotokollierung und $246\,\mathrm{ms}$ für die audio-visuelle Sprecherprotokollierung bestimmt werden. Die Medianwerte liegen bei $136\,\mathrm{ms}$ (Audio) und $104\,\mathrm{ms}$ (Audio + Video). Die Verwendung der Videodaten reduziert in einem geringen Maße die Latenz des Systems, weil die zusätzliche Information die Abfolge von Zuständen eindeutiger macht.

Eine Zusammenfassung aller Verzögerungen im System der audio-visuellen Sprecherprotokollierung ergibt eine mittlere Verzögerung zwischen dem Auftreten eines Sprechers und der Registrierung dieses Sprechers durch das System von

$$\tau_{avg} = 246\,\mathrm{ms} + 320\,\mathrm{ms} + 8\,\mathrm{ms} = 574\,\mathrm{ms}. \tag{4.81}$$

Eine Vernachlässigung der Sprecherwechselinformationen, welche aus den ΔBIC-Werten berechnet werden, würde einen Großteil der Latenz zu Lasten einer etwas verschlechterten Klassifikationsrate vermeiden (vgl. Tab. 4.4, S. 47).

Latenzbegrenzung des Viterbi-Dekodierers

In der Theorie kann die Latenz des Viterbi-Dekodierers beliebig groß sein, so dass eine Begrenzung der maximalen Latenz notwendig ist. Dieser Eingriff in den Prozess der Bestimmung der optimalen Abfolge der Zustände vergrößert die Klassifikationsfehlerrate und wird in Experimenten näher untersucht. Die Abb. 4.28 zeigt den Verlauf der Klassifikationsfehler-

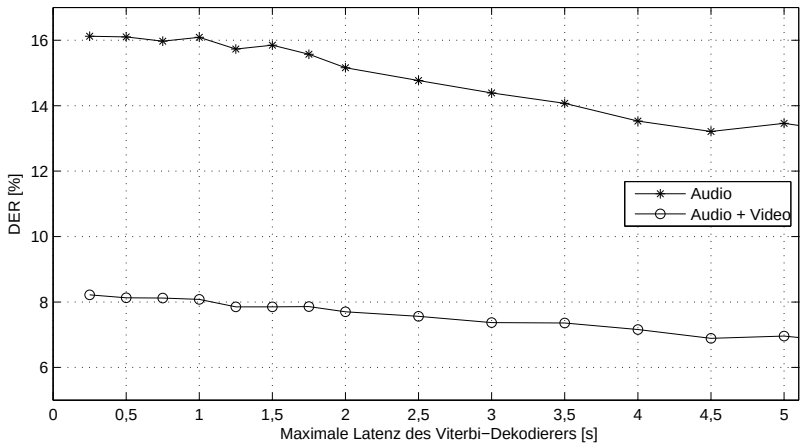


Abbildung 4.28: Abhängigkeit der Klassifikationsfehlerrate von der maximalen Latenz $\tau_{\max}$ des Viterbi-Dekodierers

rate (*DER*) gegenüber der maximalen Latenz des Viterbi-Dekodierers. Je geringer die zeitliche Begrenzung des Dekodierers gewählt wird, desto größer ist der Fehler der Klassifikation. Auf Grund der Experimente wird eine maximale Latenz von $2\,\mathrm{s}$ als vertretbarer Mittelweg zwischen Latenz und Fehlerrate gewählt. Der Vergleich der Kurvenverläufe zwischen akustischer („Audio“) und audio-visueller („Audio + Video“) Sprecherprotokollierung liefert zwei Ergebnisse. Zum einen ist unabhängig von der gewählten zeitlichen Begrenzung die Fehlerrate der audio-visuellen Sprecherprotokollierung immer geringer als bei der akustischen Sprecherprotokollierung. Zum anderen ist die Zunahme des Fehlers bei der audio-visuellen Sprecherprotokollierung geringer als bei der akustischen Sprecherprotokollierung.

Experimente zum Anwendungsszenario

Das beabsichtigte Anwendungsszenario der ambienten Kommunikation beschreibt eine Kommunikation zwischen einem oder mehreren Personen mit akustischen und visuellen Daten. Das System der Sprecherprotokollierung hat in dieser Umgebung im optimalen Fall zusätzliche Informationen über den aktuellen Sprecher, welche durch die Gesichtsidentifikation bereitgestellt werden. Dieser Vorteil kann zu einem Nachteil werden, falls eine fehlerhafte Gesichtsidentifikation vorliegt oder aber das identifizierte Gesicht nicht zum Sprecher gehört. Steht keine Gesichtsidentifikation zur Verfügung, so verhält sich das System wie eine rein akustische Sprecherprotokollierung.

Fall	Benutzer	Gesichter [%] detektiert	Gesichter [%] korrekt	DER [%] Audio	DER [%] Audio-Video	Zeit [min:sec]
Beispiele Einzelnutzer	A	83,55	83,99	5,13	2,96	3:07
	B	72,51	83,97	6,22	4,67	7:43
	C	94,18	74,60	16,54	11,65	3:18
	D	94,27	100,00	24,88	1,13	2:57
	E	93,70	19,51	6,58	14,41	2:47
	F	56,16	90,30	7,91	1,38	6:27
Beispiele Zwei Nutzer	A & D	75,99	82,76	24,56	7,81	3:14
	A & B	88,56	82,84	33,79	5,22	3:36
	C & D	89,03	86,48	15,45	8,23	7:38
	D & E	75,65	74,17	14,79	12,67	6:09
	A & F	52,90	89,84	34,25	9,78	3:31
	B & D	60,49	41,68	23,50	15,07	5:47
Mittelwert Einzelnutzer		84,53	84,79	7,46	3,72	61:18
Mittelwert zwei Nutzer		76,66	74,08	23,11	11,81	59:24
Mittelwert beide Fälle		80,46	79,49	15,16	7,70	120:42

Tabelle 4.5: Experimente zur audio-visuellen Sprecherprotokollierung

In Tab. 4.5 sind die Ergebnisse verschiedener Testläufe der audio-visuellen Sprecherprotokollierung dargestellt. Insgesamt wurden Aufnahmen von über $2\,\mathrm{h}$ Länge für die Experimente gemacht und ausgewertet. In der dritten Spalte ist der Prozentsatz der detektierten Gesichter und in der vierten der Prozentsatz der korrekt identifizierten Gesichter angegeben. Die fünfte Spalte gibt die Klassifikationsfehlerrate für die akustische und die sechste die Klassifikationsrate für die audio-visuelle Sprecherprotokollierung wieder. In der letzten Spalte ist die Zeitdauer des Experiments angegeben. Die ersten Zeilen der Tabelle zeigen eine Auswahl der Experimente mit Einzelnutzern und die darauf folgenden Zeilen die Experimente mit zwei Nutzern. Die Mittelwerte für die gesamten Aufnahmen beider Fälle sind in den letzten Zeilen angegeben.

Im Falle eines einzelnen Nutzers beträgt die mittlere Fehlerrate des Systems $7{,}46\,\%$ im rein akustischen Ansatz, und die Verwendung der visuellen Daten ermöglicht eine Reduktion der *DER* auf $3{,}72\,\%$. Betrachtet man die einzelnen Experimente genauer, so fallen die Nutzer „D“ im positiven und „E“ im negativen Sinne auf. Der Nutzer „D“ wird durch die Kamera in über $94{,}00\,\%$ der Zeit detektiert und dabei zu $100{,}00\,\%$ richtig identifiziert. Erwartungsgemäß verbessert sich die Klassifikationsfehlerrate von zunächst unterdurchschnittlichen $24{,}88\,\%$ auf einen sehr guten Wert von $1{,}13\,\%$. Im Gegensatz dazu wird der Nutzer „E“ häufig durch die Gesichtsidentifikation falsch klassifiziert. Obwohl sein Gesicht in $93{,}70\,\%$ der Fällen detektiert wird, kann es nur in $19{,}51\,\%$ korrekt identifiziert werden. Dies hat einen negativen Effekt auf die audio-visuelle Sprecherprotokollierung und führt zu einer Verschlechterung der Klassifikationsrate um $7{,}83\,\%$.

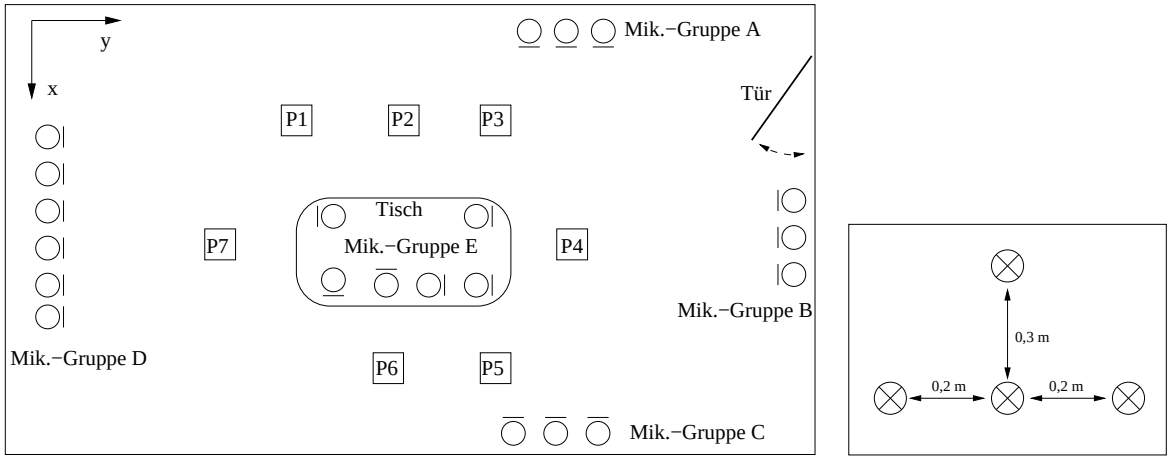
Die Beispiele für zwei Nutzer zeigen ein zu den Einzelnutzern vergleichbares Bild. Die Fehlerrate der akustischen Sprecherprotokollierung ist durch die Dialogsituation etwas höher als im Einzelnutzerfall. Die Verwendung der Videodaten führt im Mittel zu einer Verbesserung der Klassifikationsraten von $23{,}11\,\%$ auf $11{,}81\,\%$. Die Mittelung aller Daten zeigt annähernd eine Halbierung der Klassifikationsrate durch die Verwendung des audio-visuellen Ansatzes.

5 Akustische Ereignisdetektion

Die akustische Ereignisdetektion ist ein Teil der akustischen Szenenanalyse, welcher sich speziell mit der Identifikation von akustischen Ereignissen in der häuslichen Umgebung befasst. Da Mikrophone im Gegensatz zu Kameras dauerhaft den ganzen Raum erfassen können und unabhängig von der Beleuchtung sind, bieten sie die Möglichkeit, eine alternative Informationsquelle zu visuellen Verfahren zu erschließen. Die Auswahl von Ereignissen ist zunächst durch das Vorhandensein verfügbarer Daten zum Training und Testen begrenzt und orientiert sich an verfügbaren Datenbasen.

5.1 Datenbasis Ereignisdetektion

Die hier verwendete Datenbasis zur Erkennung akustischer Ereignisse wurde im Rahmen des *CHIL* Projektes erstellt und besteht aus insgesamt 3 Sitzungen [TMNS05]. Die Aufnahmen wurden in einem Konferenzraum der Größe $5{,}2\,\mathrm{m} \times 3{,}9\,\mathrm{m}$ mit weiblichen und männlichen Personen erstellt (vgl. Abb. 5.1 (a)).



(a) Aufbau des *CHIL* Konferenzraumes und Platzierung der Mikrophongruppen

(b) Aufbau der T-förmigen Mikrophongruppe

Abbildung 5.1: Experimenteller Aufbau der Datenbasis zur akustischen Ereignisdetektion

Jeder Teilnehmer musste eine vorgegebene Menge an akustischen Ereignissen an den definierten Plätzen P_1 bis P_7 erzeugen. Die Daten wurden dabei mit 3 T-förmigen Mikrophongruppen (Mik.-Gruppe A bis C) bestehend aus 4 Mikrophonen, einer linearen Mikrophongruppe mit 7 Mikrophonen (Mik.-Gruppe D) und 7 auf dem Tisch verteilten Mikrophonen (Mik.-Gruppe E) aufgenommen. Der Abstand der Mikrophone innerhalb einer Gruppe wurde zu $0{,}2\,\mathrm{m}$ bzw. $0{,}3\,\mathrm{m}$ gewählt (vgl. Abb. 5.1 (b)). Die Abtastfrequenz der Aufnahmen

betrug $44{,}1\,\mathrm{kHz}$ und wurde für die Experimente auf $16\,\mathrm{kHz}$ reduziert. In der Datenbasis sind die folgenden 14 verschiedenen akustischen Ereignisse enthalten, deren Häufigkeit in Klammern angegeben ist:

- ap (60): Applaudieren (mehrere Personen)
- cl (64): Rühren eines Löffels in einer Tasse
- cm (76): Verrücken eines Stuhls
- co (65): Husten oder Räuspern
- do (60): Öffnen einer Tür
- ds (61): Schließen einer Tür
- kj (65): Ablegen oder Aufnehmen eines Schlüsselbundes
- kn (50): Klopfen an einer Tür oder auf einem Tisch
- kt (66): Tippen auf einer Tastatur
- la (64): Lachen
- pr (116): Klingeln eines Mobiltelefons
- pw (84): Papierrascheln
- st (73): Schritte
- un (126): Unbekannt

5.2 Experimente zur Modellierung

Die Identifikation von akustischen Ereignissen ist im Vergleich zur Sprecheridentifikation ein neueres Thema in der Forschung. Zunächst soll daher eine geeignete Modellierung der Ereignisse auf Basis der in der akustischen Szenenanalyse verwendeten Merkmalsvektoren gefunden werden. Dieser Ansatz bietet den Vorteil, dass sowohl für die Ereignisdetektion als auch für die Sprecheridentifikation die gleichen Merkmale verwendet werden und in einem gemeinsamen System die aufwendige Neuberechnung von alternativen Merkmalen entfällt. Die Ergebnisse der *CHIL* Projektevaluation der Ereignisdetektion können in [TMZ+07] und [BP08] nachgelesen werden.

Für die Experimente wird ein Drittel der Daten zum Training (Sitzung 1 auf DVD 1) und zwei Drittel zum Testen (Sitzung 2 auf DVD 2 und Sitzung 3 auf DVD 3) verwendet. Als Ausgangspunkt für die Modellbildung werden zwei Ansätze näher untersucht. Zum einen werden Modelle bestehend aus Gauß'schen Mischungsverteilungen mit einer unterschiedlichen Anzahl von Verteilungen auf den Trainingsdaten geschätzt (*GMM*-Ansatz). Zum anderen wird eine Gruppierung der Ereignisse anhand der Konfusionsmatrix der Erkennungsergebnisse in zwei Gruppen vorgenommen. Für diese Gruppen werden, entsprechend dem Ansatz zur Sprecheridentifikation, Hintergrundmodelle geschätzt und auf jedes Ereignis einzeln adaptiert (*UBM*-Ansatz). Beide Verfahren nutzen Gauß'sche Mischungsverteilungen zur Modellierung der akustischen Ereignisse, jedoch wird im Folgenden zur leichteren Unterscheidung entweder vom *GMM*-Ansatz oder *UBM*-Ansatz gesprochen.

5.2.1 Modellierung mit Gauß'schen Mischungsverteilungen

Die Modellierung der Ereignisse durch Gauß'sche Mischungsverteilungen erfordert die Festlegung der Modellkomplexität durch die Wahl der Verteilungsanzahl. Mit steigender Vertei-

lungsanzahl können zwar die Ereignisse theoretisch besser modelliert werden, jedoch nimmt die benötigte Rechenleistung zu. Zudem ist die Menge an Trainingsdaten begrenzt und eine zu große Modellkomplexität wird, wie aus der Spracherkennung bekannt, durch stagnierende bzw. verringerte Klassifikationsergebnisse erkennbar sein. Zunächst soll dieser Aspekt der Modellierung experimentell untersucht werden.

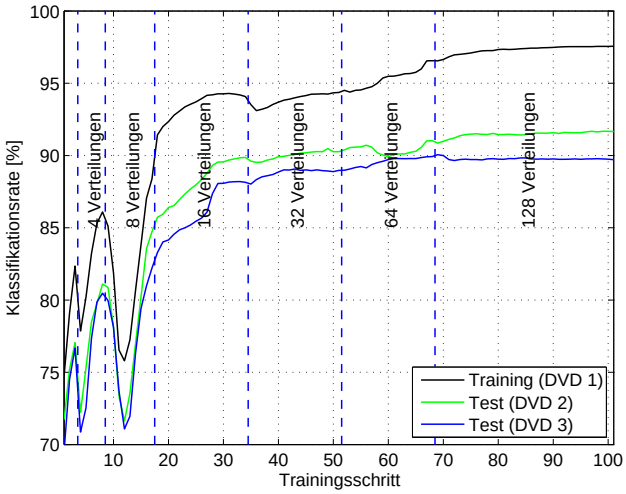


Abbildung 5.2: Vergleich der Klassifikationsraten des *GMM*-Ansatzes

In Abb. 5.2 sind die Klassifikationsraten über die Trainingsschritte angegeben. Hierbei werden 42-dimensionale Merkmalsvektoren verwendet, welche aus den *MFCC*- und *MACV*-Werten, deren ersten zeitlichen Ableitungen und deren zweiten zeitlichen Ableitungen bestehen. Zu den durch senkrechte blaue Linien gekennzeichneten Trainingsschritten wird eine Aufspaltung der Verteilungen (engl. *density splitting*) durchgeführt, so dass eine Verdoppelung der Verteilungsanzahl erzielt wird. Dabei werden, wie in der automatischen Spracherkennung üblich, die Verteilungen mit den größten Gewichten in zwei oder mehrere Verteilungen aufgeteilt. Es ist erkennbar, dass jeweils nach der Aufspaltung der Verteilungen eine Phase der Modellanpassung erfolgt, in welcher die Klassifikationsraten zunächst abnehmen und anschließend steigen.

Ein Vergleich der Klassifikationsraten auf den Trainingsdaten (DVD 1) und den Testdaten (DVD 2, DVD 3) zeigt, dass bei der Erhöhung der Modellkomplexität von 64 auf 128 Verteilungen die Klassifikationsrate der Trainingsdaten verbessert wird. Jedoch stagniert die Klassifikationsrate auf den Testdaten. Infolgedessen wird kein weiteres Aufspalten der Verteilungen mehr vorgenommen, um eine Überanpassung (engl. *overfitting* [DHS01]) der Modelle an die Trainingsdaten zu vermeiden. Im Vergleich zwischen Trainings- und Testdaten ist erkennbar, dass die Ergebnisse der beiden Testdaten (DVD 2, DVD 3) nahe aneinander liegen und gegenüber den Trainingsdaten ca. $5\,\%$ schlechter klassifiziert werden.

In Abb. 5.3 sind, aufgeschlüsselt nach den Ereignissen, die Klassifikationsraten auf den Testdaten angegeben. Die Ereignisse Schritte („*st*“) und Papier („*pw*“) erzielen die schlechtesten Ergebnisse, was auf die geringe Energie der akustischen Ereignisse im Vergleich zu den anderen Ereignissen zurückzuführen ist. Das Modell Unbekannt („*un*“) bildet ein Sammelmodell für alle unbekannten akustischen Ereignisse in den Aufnahmen der Datenbasis. Zusätzlich ist der Mittelwert („*avg*“) der Klassifikationsrate über alle Ereignisse angegeben.

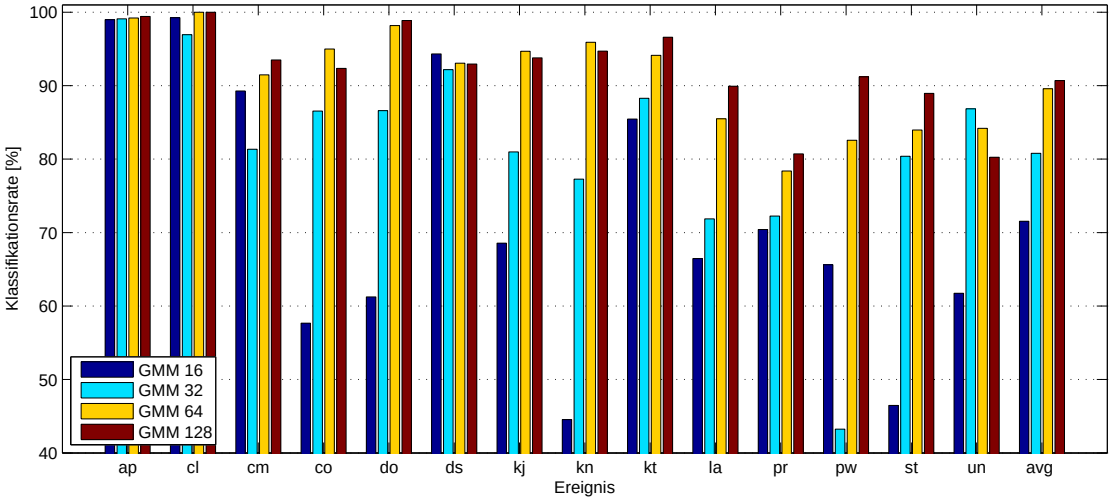


Abbildung 5.3: Vergleich der Klassifikationsraten des *GMM*-Ansatzes bezogen auf die einzelnen Ereignisse auf Testdaten (DVD 2, DVD 3)

5.2.2 Modellierung mit universellen Hintergrundmodellen

Die Modellierung der Ereignisse mit Hilfe von universellen Hintergrundmodellen ist ein Ansatzpunkt, um die geringe Anzahl an Trainingsbeispielen in der Datenbasis zu kompensieren. Bei der Modellierung durch universelle Hintergrundmodelle werden die zu trainierenden Klassen in Gruppen eingeteilt, so dass Ereignisse mit vergleichbaren akustischen Eigenschaften in einer Gruppe sind. Diese bei der Sprecheridentifikation natürlich gegebene Einteilung in zwei Gruppen (männliche und weibliche Sprecher) ist bei der akustischen Ereignisdetektion nicht gegeben.

Die Einteilung der Ereignisse in Gruppen erfolgt in zwei Schritten. Zunächst werden die Ereignisse anhand des akustischen Eindrucks in die zwei Gruppen

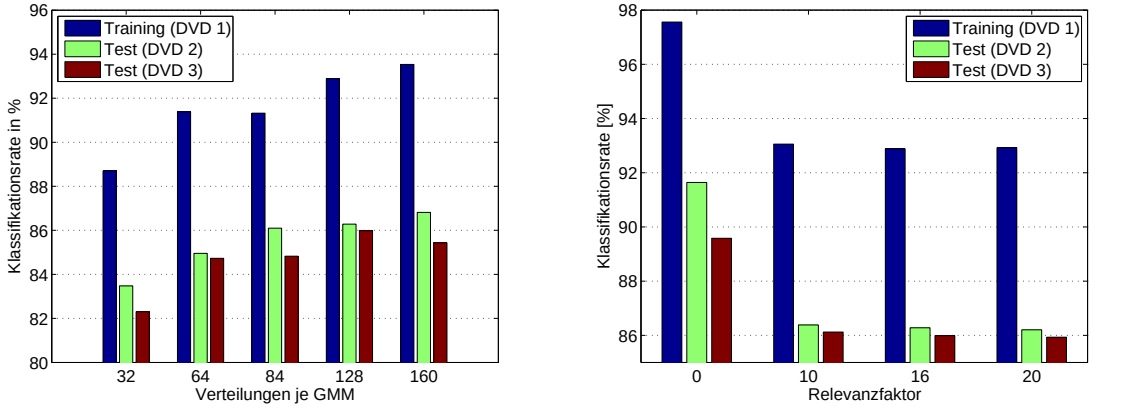
- Gruppe 1: do, ds, kn, kt, st
- Gruppe 2: ap, cl, cm, co, kj, la, pr, pw, un

eingeteilt, wobei die erste Gruppe klopfende und schlagende Ereignisse umfasst und die zweite Gruppe die übrigen Ereignisse modelliert. Grundgedanke dabei ist, die Anzahl der Hintergrundmodelle gering zu halten, dabei jedoch Gruppen mit ähnlichen akustischen Eigenschaften zu erzeugen. Dies ist notwendig, da bei der Bayes'schen Adaption Teile der Hintergrundmodelle mit in die neuen Modelle der Ereignisse eingehen. Eine große Abweichung der Hintergrundmodelle von den zu erzeugenden Modellen wäre somit nachteilhaft und ist vergleichbar mit der Adaption eines weiblichen Hintergrundmodells auf einen männlichen Sprecher.

Erste experimentelle Versuche mit 64 Verteilungen zeigten, dass entgegen der Annahme, dass die meisten Fehler durch Verwechselungen innerhalb einer Gruppe auftreten würden, einige Ereignisse häufig Modellen der anderen Gruppe zugeordnet wurden. Folglich wurden die Ereignisse mit Hilfe der Konfusionsmatrix neu geordnet, so dass die zwei Gruppen

- Gruppe 1: do, ds, kn, kt, la, pr, pw, st
- Gruppe 2: ap, cl, cm, co, kj, un

gebildet wurden.



(a) Variation der GMM-Komplexität ($r = 16$) (b) Einfluss des Relevanzfaktors (128 Verteilungen)

Abbildung 5.4: Experimente zur Modellbildung durch den *UBM*-Ansatz

Die Abb. 5.4 (a) zeigt die Klassifikationsraten in Abhängigkeit von der Modellkomplexität für die Modellierung durch universelle Hintergrundmodelle. Bis zu einer Anzahl von ca. 128 Verteilungen steigt die Klassifikationsrate mit zunehmender Modellkomplexität. Oberhalb von 128 Verteilungen pro Ereignis kann keine signifikante Verbesserung durch die Verwendung von mehr Verteilungen erzielt werden. In Abb. 5.4 (b) ist der Einfluss des Relevanzfaktors, welcher bei der Adaption der Modelle vom Hintergrundmodell verwendet wird, auf die Klassifikationsrate dargestellt. Ein geringer Relevanzfaktor bedeutet, dass dem Hintergrundmodell eine geringere Relevanz als den vorhandenen Trainingsdaten zugeordnet wird (vgl. Gl. 4.47-Gl. 4.53, S. 32). Es ist zu erkennen, dass die Klassifikationsrate mit steigendem Relevanzfaktor ($r = 10, 16, 20$) abnimmt und somit die Modellierung durch den Ansatz der Hintergrundmodelle grundsätzlich in Frage gestellt werden muss. Um den Unterschied zu verdeutlichen, sind die Werte für den *GMM*-Ansatz, welcher mit einem Relevanzfaktor 0 gleichzusetzen ist, ebenfalls eingetragen.

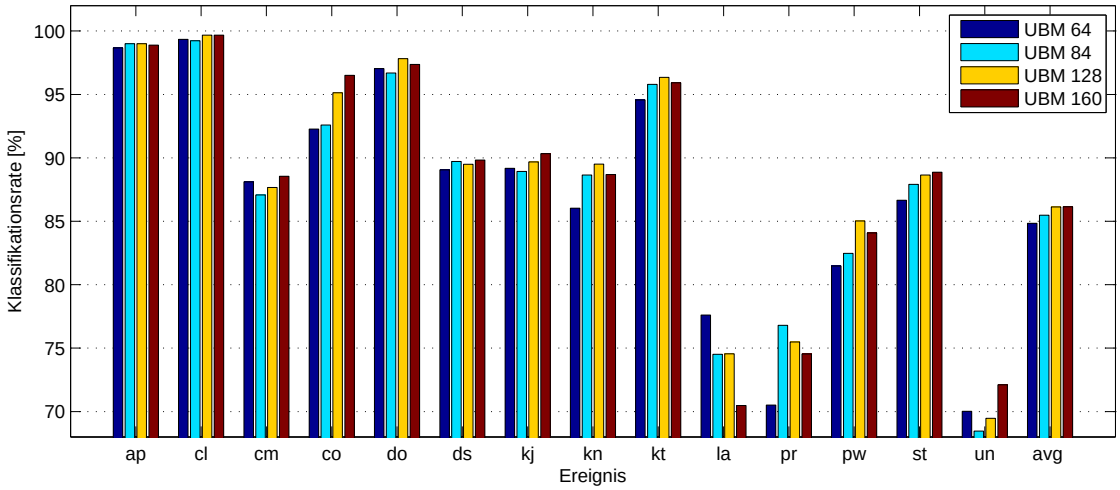


Abbildung 5.5: Vergleich der Klassifikationsraten des *UBM*-Ansatzes mit Relevanzfaktor $r = 16$ bezogen auf die einzelnen Ereignisse auf Testdaten (DVD 2, DVD 3)

In Abb. 5.5 sind die Klassifikationsraten für den *UBM*-Ansatz dargestellt. Eine Beobachtung aus den Experimenten ist, dass für einen Teil der akustischen Ereignisse, wie z. B. Schritte („*st*“), eine steigende Anzahl der Mischungsverteilungen eine bessere Erkennungsleistung ermöglicht, während bei anderen Ereignissen, wie z. B. Lachen („*la*“), eine größere Anzahl der Mischungsverteilungen den entgegengesetzten Effekt hat. Der direkte Vergleich der Modellierungsarten in Abb. 5.6 zeigt die unterschiedlichen Vorteile der Verfahren. Die *GMM*-Modellierung erzielt mit einer mittleren Klassifikationsrate von $90{,}7\,\%$ bessere Ergebnisse als der *UBM*-Ansatz mit $86{,}3\,\%$, auch wenn für einzelne akustische Ereignisse der *UBM*-Ansatz besser ist.

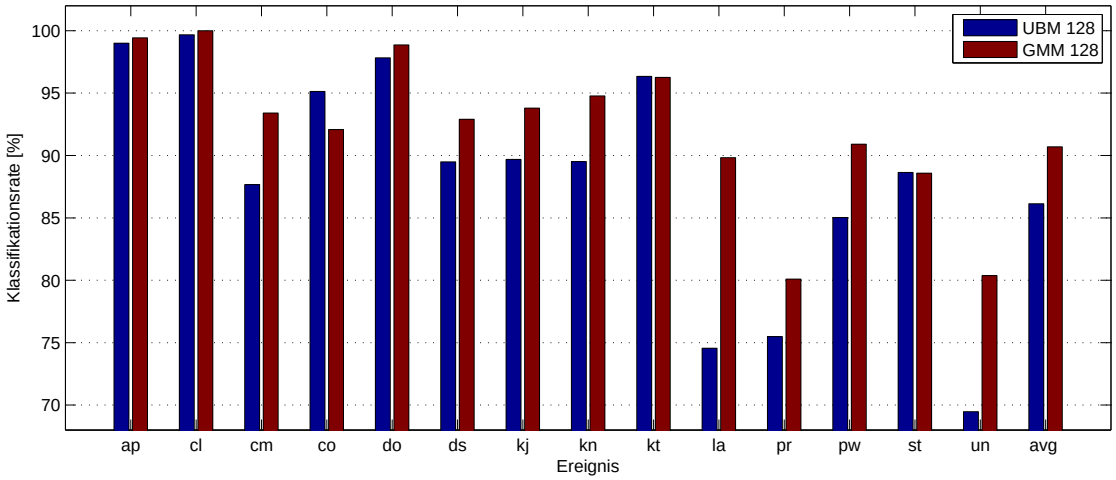


Abbildung 5.6: Vergleich der Klassifikationsraten des *UBM*- und des *GMM*-Ansatzes auf Testdaten (DVD 2, DVD 3)

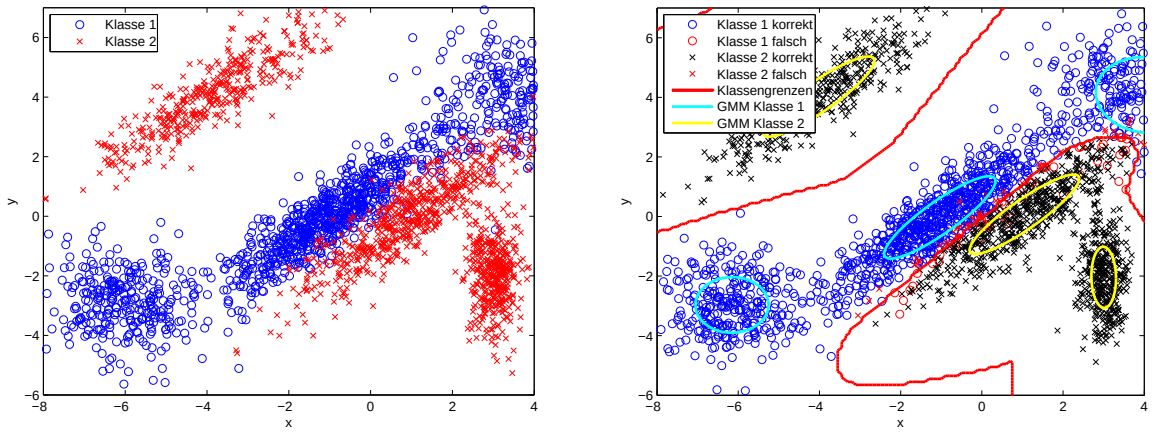
5.3 Diskriminative Lernverfahren

Statistische Klassifikationsverfahren sind in der Literatur weit verbreitet. Hierbei werden Merkmale als Zufallsvariablen mit zugehörigen klassenbedingten Verteilungen beschrieben, die häufig Gauß'sche Mischungsverteilungen verwenden. Zur Schätzung der Modellparameter gibt es unterschiedliche Ansätze. Am weitesten verbreitet ist die „*Maximum Likelihood*“-Parameterschätzung (*ML*-Parameterschätzung), bei dem die Modellparameter so bestimmt werden, dass die *Likelihoods* der Trainingsdaten maximiert werden.

Wenn die vorgegebene Form der klassenbedingten Verteilungen korrekt und die Trainingsdatenmenge sehr groß ist, dann können mit der *ML*-Parameterschätzung die den Daten zugrundeliegenden Verteilungen korrekt geschätzt werden. Durch die Anwendung der Bayes'schen Entscheidungsregel erzielt man in diesem Fall die minimale Fehlerrate. In der Praxis sind diese Annahmen jedoch meist nicht erfüllt. Dann wird mit der *ML*-Parameterschätzung die eigentlich interessierende Größe, die Klassifikationsrate, nicht mehr unbedingt optimiert [LYL07].

Diskriminative Lernverfahren greifen diesen Punkt auf und versuchen, durch das Einbeziehen aller Klassen im Trainingsprozess eine minimale Fehlerrate zu erzielen. Dabei können auch Näherungen und Einschränkungen in den Modellen, wie z. B. diagonale Kovarianzmatrizen, mit diskriminativen Ansätzen besser behandelt werden als bei der klassischen

ML-Parameterschätzung [NCM91]. In der akustischen Ereignisdetektion werden diagonale Kovarianzmatrizen in den Modellen verwendet. Folglich könnten diskriminative Lernverfahren zu einer Verbesserung der Klassifikationsraten führen.



(a) Beispieldaten des 2-Klassenproblems

(b) Klassengrenzen Bayes'sche Entscheidungsregel

Abbildung 5.7: Beispieldaten eines 2-Klassenproblems und zugehörige Klassengrenzen nach der Bayes'schen Entscheidungsregel (vollständig besetzte Kovarianzmatrizen)

Im Folgenden wird das diskriminative Lernverfahren der „*Maximum Mutual Information*"-Parameterschätzung vorgestellt. Anschließend werden die experimentellen Ergebnisse auf der Datenbasis zur akustischen Ereignisdetektion diskutiert. Da die Darstellung von Daten mit mehr als zwei Dimensionen in Graphen nicht möglich ist, wird zur Veranschaulichung ein 2-Klassenproblem in zwei Dimensionen (vgl. Abb. 5.7 (a)) betrachtet. Den Daten liegen Gauß'sche Mischungsverteilungen zugrunde, die je Klasse aus drei multivariaten Normalverteilungen mit vollständig besetzten Kovarianzmatrizen bestehen. In Abb. 5.7 (b) sind die idealen Klassengrenzen nach der Bayes'schen Entscheidungsregel eingezeichnet. Die einzelnen Mischungsverteilungen werden hierbei durch Ellipsen angedeutet.

Zur Simulation von Modellierungsfehlern wird die Annahme getroffen, dass die zu schätzenden Kovarianzmatrizen eine diagonale Form haben. Folglich werden durch die Parameterschätzung Kovarianzmatrizen der Form

$$\widehat{\boldsymbol{\Sigma}}_{i,m} = \begin{pmatrix} \sigma^2_{i,m,1} & 0 \\ 0 & \sigma^2_{i,m,2} \end{pmatrix} \tag{5.1}$$

ermittelt. Im Anhang A.4 (S. 121) sind für dieses Beispiel die Modellparameter und die Ergebnisse der Parameterschätzung für verschiedene Verfahren aufgeführt.

5.3.1 *MMI*-Parameterschätzung

Das Ziel der „*Maximum Mutual Information*"-Parameterschätzung (*MMI*-Parameterschätzung) ist die Maximierung der Transinformation zwischen den Merkmalsvektoren und den zugehörigen Klassen, welche durch eine Maximierung der a posteriori Wahrscheinlichkeiten der Klassen gegeben die Merkmalsvektoren erreicht werden kann [HD08]. Dies führt zu einer Maximierung der Anzahl korrekt klassifizierter Merkmalsvektoren im Training [LP96].

Im Folgenden wird eine Menge von K Klassen betrachtet, deren Modellparameter geschätzt werden sollen. Jede Klasse soll durch eine Gauß'sche Mischungsverteilung beschrieben werden, die aus einer gewichteten Summe von M multivariaten Normalverteilungen bestehen soll. Der Vektor $\boldsymbol{\Theta}$ beinhaltet die Parameter der Mischungsverteilungen aller Klassen, bestehend aus den Gewichten $c_{k,m}$, den Mittelwertvektoren $\boldsymbol{\mu}_{k,m}$ und den Kovarianzmatrizen $\boldsymbol{\Sigma}_{k,m}$. Der erste Index steht hierbei für die Klasse und der zweite Index für die betrachtete Mischungsverteilung.

Für die Parameterschätzung sind je Klasse N_k Merkmalsvektoren der Dimension D mit $\boldsymbol{X}_{k,1:N_k} = [\boldsymbol{x}_k(1), \ldots, \boldsymbol{x}_k(N_k)]$ vorhanden ($k = 1, \ldots, K$). Des Weiteren wird die Zufallsvariable der Klassenzugehörigkeit eines Merkmalsvektors $\boldsymbol{x}_k(n)$ mit Ω bezeichnet. Sie kann die diskreten Werte aus der Menge $\mathcal{O} = \{1, \ldots, K\}$ annehmen. Zusätzlich wird die Zufallsvariable Z verwendet, um die Zugehörigkeit eines Merkmalsvektors zu einer Mischungsverteilung anzuzeigen. Diese Zufallsvariable kann die diskreten Werte der Menge $\mathcal{Z} = \{1, \ldots, M\}$ annehmen.

Ein Merkmalsvektor der Klasse i wird nach der Bayes'schen Entscheidungsregel korrekt klassifiziert, falls

$$P(\Omega = i|\boldsymbol{x}_i(n); \boldsymbol{\Theta}) > P(\Omega = j|\boldsymbol{x}_i(n); \boldsymbol{\Theta}) \qquad \text{für alle } j \neq i \tag{5.2}$$

gilt, wobei $\boldsymbol{\Theta}$ die Abhängigkeit der Entscheidungsregel von den *GMM*-Modellparametern anzeigt. Soll die Anzahl korrekt klassifizierter Merkmalsvektoren maximiert werden, so müssen folglich die a posteriori Wahrscheinlichkeiten der Klassen gegeben die Merkmalsvektoren maximiert werden. Dabei ist die a posteriori Wahrscheinlichkeit der i-ten Klasse für die Merkmalsvektoren $\boldsymbol{X}_{i,1:N_i}$ mit

$$P(\Omega = i|\boldsymbol{X}_{i,1:N_i}; \boldsymbol{\Theta}) = \prod_{n=1}^{N_i} \frac{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}) \cdot P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}) \cdot P(\Omega = k)} \tag{5.3}$$

gegeben. Der Logartihmus der Gl. 5.3 wird im Folgenden als Zielfunktion bezeichnet, welche durch die *MMI*-Parameterschätzung maximiert wird.

Eine ausführliche Herleitung der Adaptionsgleichungen der *MMI*-Parameterschätzung ist im Anhang A.2 (S. 115) zu finden. Zunächst wird dabei der Gradient der Zielfunktion

$$Q_i(\boldsymbol{\Theta}) = \sum_{n=1}^{N_i} \log \left(\frac{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}) \cdot P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}) \cdot P(\Omega = k)} \right) \tag{5.4}$$

bezüglich des gesuchten Modellparameters bestimmt. Anschließend wird der Gradient der Zielfunktion zu null gesetzt, so dass die Gleichungen für die Schätzwerte der Gewichte, Mittelwerte und Kovarianzmatrizen bestimmt werden können.

Die Parameterschätzung mittels *MMI* wird in einem iterativen Verfahren durchgeführt, welches einen *EM*-Algorithmus verwendet. Zur Initialisierung des Algorithmus werden die Modellparameter der *ML*-Parameterschätzung genutzt. Im ersten Schritt, dem Erwartungswertschritt (engl. *Expectation*), wird eine Schätzung von zwei versteckten Parametern vorgenommen. Dies sind die Wahrscheinlichkeit einer Fehlklassifikation des Merkmalsvektors

und die Zugehörigkeit des Merkmalsvektors zu einer Mischungsverteilung. Die Erwartungswerte der versteckten Parameter werden anhand der aktuellen Modellparameter geschätzt. Im zweiten Schritt, dem Maximierungsschritt (engl. *Maximization*), werden die im vorherigen Schritt berechneten versteckten Parameter verwendet, um eine neue Schätzung der Modellparameter durchzuführen. Anschließend wird zur Verbesserung des Konvergenzverhaltens eine Glättung der Parameterschätzungen vorgenommen. Im Folgenden wird ein Überblick über den Algorithmus gegeben.

EM-Algorithmus zur *MMI*-Parameterschätzung

1. **Initialisierung**: Setze den Iterationszähler $\nu = 0$ und initialisiere die Parameter $\boldsymbol{\Theta}_i(\nu)$ der i-ten Klasse mit den Modellparametern der *ML*-Parameterschätzung für diese Klasse.

2. **Erwartungswertschritt**: Berechne für jeden Merkmalsvektor $\boldsymbol{x}_i(n), n = 1, \ldots, N_i$ die Wahrscheinlichkeit der Fehlklassifikation durch die aktuellen Modelle mit

$$\psi_i(n) = \left(1 - \frac{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i(\nu)) \cdot P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k(\nu)) \cdot P(\Omega = k)} \right) \tag{5.5}$$

   und für jede Mischungsverteilung $j = 1, \ldots, M$ die Wahrscheinlichkeit, dass der Merkmalsvektor zu dieser Mischungsverteilung gehört mit

$$\gamma_{i,j}(n) = \left(\frac{p(\boldsymbol{x}_i(n)|\Omega = i, Z = j; \boldsymbol{\Theta}_i(\nu)) \cdot P(Z = j|\Omega = i)}{\sum_{m=1}^{M} p(\boldsymbol{x}_i(n)|\Omega = i, Z = m; \boldsymbol{\Theta}_i(\nu)) \cdot P(Z = m|\Omega = i)} \right). \tag{5.6}$$

3. **Maximierungsschritt**: Schätzung der Modellparameter $\widehat{\boldsymbol{\Theta}}_i$ unter Verwendung der im vorherigen Schritt berechneten Erwartungswerte mit

   - Gewichte

$$\widehat{c}_{i,j} = \frac{\sum_{n=1}^{N_i} \psi_i(n) \cdot \gamma_{i,j}(n)}{\sum_{n=1}^{N_i} \psi_i(n)} \tag{5.7}$$

   - Mittelwerte

$$\widehat{\boldsymbol{\mu}}_{i,j} = \frac{\sum_{n=1}^{N_i} [\psi_i(n) \cdot \gamma_{i,j}(n) \cdot \boldsymbol{x}_i(n)]}{\sum_{n=1}^{N_i} \psi_i(n) \cdot \gamma_{i,j}(n)} \tag{5.8}$$

- Kovarianzmatrizen

$$\widehat{\boldsymbol{\Sigma}}_{i,j} = \frac{\sum\limits_{n=1}^{N_i} \left[\psi_i(n) \cdot \gamma_{i,j}(n) \left(\boldsymbol{x}_i(n) - \boldsymbol{\mu}_{i,j} \right) \left(\boldsymbol{x}_i(n) - \boldsymbol{\mu}_{i,j} \right)^T \right]}{\sum\limits_{n=1}^{N_i} \psi_i(n) \cdot \gamma_{i,j}(n)} \tag{5.9}$$

4. **Glättung**: Berechnung der neuen Modellparameter als Kombination aus den aktuellen Modellparametern $\boldsymbol{\Theta}_i(\nu)$ und den neu geschätzten Modellparametern $\widehat{\boldsymbol{\Theta}}_i$ des Maximierungsschrittes mit

$$\boldsymbol{\Theta}_i(\nu+1) = \alpha \cdot \boldsymbol{\Theta}_i(\nu) + (1-\alpha) \cdot \widehat{\boldsymbol{\Theta}}_i \qquad \text{für} \qquad \alpha \in [0,1]. \tag{5.10}$$

   Erhöhe Iterationsindex $\nu = \nu+1$ und gehe zu „Schritt 2“ **oder** Abbruch nach erreichen der gewünschten Iterationsanzahl.

Diskussion

Die Schätzungen der Mischungsparameter $\widehat{\boldsymbol{\Theta}}_i$ nach Gl. 5.7 (Mischungsgewichte), Gl. 5.8 (Mittelwertvektoren) und Gl. 5.9 (Kovarianzmatrizen) erfolgt iterativ, wobei für die Berechnung der neuen Schätzwerte $\widehat{\boldsymbol{\Theta}}_i$ die vorherigen Schätzwerte $\boldsymbol{\Theta}_i(\nu)$ aus der letzten Iteration verwendet werden. Hierbei kann es zu einem schwingenden Verhalten der Schätzungen kommen, das durch den Glättungsschritt (vgl. *EM*-Algorithmus 4. Schritt) gedämpft wird. Alternativ kann in die Optimierung eine Nebenbedingung eingeführt werden, welche die Distanz zwischen neuen und alten Schätzwerten der Parameter begrenzt [LLJ+08].

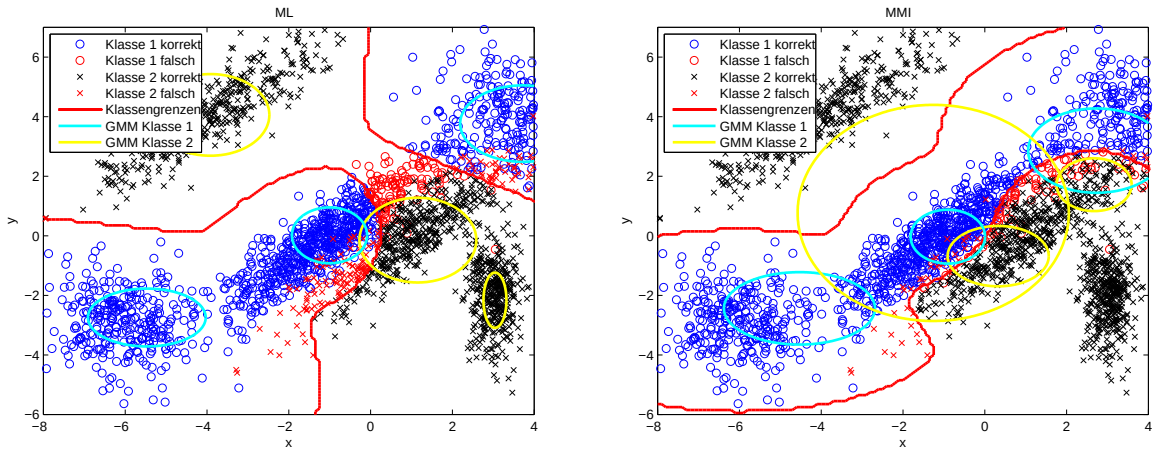
Eine Gegenüberstellung der Gleichungen zur Schätzung der Mischungsparameter entsprechend der *ML*-Parameterschätzung (vgl. Gl. 4.47, S. 32) und dem *MMI*-Verfahren zeigt eine hohe Ähnlichkeit der Ansätze. Das *MMI*-Verfahren verwendet im Vergleich zum *ML*-Verfahren den zusätzlichen Gewichtsfaktor

$$\psi_i(n) = \left(1 - \frac{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i(\nu)) P(\Omega = i)}{\sum\limits_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k(\nu)) P(\Omega = k)} \right) \tag{5.11}$$

für die Schätzung der neuen Modellparameter, wodurch eine Gewichtung der Merkmalsvektoren anhand der Wahrscheinlichkeit der Fehlklassifikation vorgenommen wird. Falls ein Merkmalsvektor $\boldsymbol{x}_i(n)$ durch die aktuellen Modellparameter mit einer hohen Wahrscheinlichkeit falsch klassifiziert wird, so wird $p(\Omega = i|\boldsymbol{x}_i(n); \boldsymbol{\Theta}(\nu))$ einen kleinen Wert annehmen und der Gewichtsfaktor $\psi_i(n)$ strebt gegen den Wert Eins. Umgekehrt wird ein zuverlässig richtig klassifizierter Trainingsvektor einen Gewichtsfaktor von annähernd Null besitzen ($\psi_i(n) \rightarrow 0$). Folglich berücksichtigt das *MMI*-Verfahren während der Schätzung der neuen Modellparameter die vermutlich falsch klassifizierten Trainingsvektoren stärker als die vermutlich richtig klassifizierten Vektoren.

Mit Hilfe des Gewichtsfaktors $\psi_i(n)$ kann das zuvor erwähnte schwingende Verhalten der Modellparameterschätzung während der Iterationen erklärt werden. Angenommen, die Menge A von Merkmalsvektoren wird zunächst zuverlässig korrekt klassifiziert und die gleich

große Menge B derselben Klasse wird falsch klassifiziert, so wird für die Schätzung der Mittelwertvektoren die Gruppe B im Verhältnis zur Gruppe A stärker verwendet. Es kommt zu einer Verschiebung der Mittelwerte in Richtung der Menge B und folglich zu einer Änderung der Klassengrenzen, was im nächsten Iterationsschritt dazu führen kann, dass nun die Vektoren der Menge A anstatt der Menge B falsch klassifiziert werden. Es wird somit im nächsten Schritt eine Gegenbewegung in Richtung der Menge A entstehen, welche bei fehlender Dämpfung zu einem schwingenden Verhalten führt.



(a) *ML*-Parameterschätzung (b) *MMI*-Parameterschätzung

Abbildung 5.8: Vergleich der Klassengrenzen von Modellen nach einer *ML*- bzw. *MMI*-Parameterschätzung (diagonale Kovarianzmatrizen)

In Abb. 5.8 sind die Ergebnisse der Klassifikation der durch *ML*- bzw. *MMI*-Parameterschätzung gewonnenen Modelle dargestellt. Die *ML*-Parameterschätzung optimiert die Modellparameter der einzelnen Klassen so, dass die *Likelihood* der Trainingsdaten maximiert wird. Die so entstehenden Klassengrenzen sind nicht optimal für die Separation der Trainingsdaten, wie die in rot eingezeichneten Klassengrenzen in Abb. 5.8 (a) verdeutlichen. Jede Klasse wird durch drei Mischungsverteilungen modelliert, welche als Ellipsen angedeutet sind. Die Hauptachsen der Ellipsen sind dabei proportional zur Standardabweichung der Verteilungen in der jeweiligen Raumrichtung. Die *MMI*-Parameterschätzung hat das Ziel, die Transinformation zu maximieren, wodurch die Modellierung der Klassen nebensächlich wird. Dies ist deutlich aus Abb. 5.8 (b) zu entnehmen, da z. B. die Daten der Klasse 2 bei $[x, y] = [3, -2]$ durch keine Mischungsverteilung mehr direkt modelliert werden. Vielmehr werden diese Daten automatisch durch die gebildeten Klassengrenzen korrekt klassifiziert. Die Fehlerrate bei der Klassifizierung sinkt von $10{,}6\,\%$ bei der *ML*-Parameterschätzung auf $5{,}4\,\%$ bei der *MMI*-Parameterschätzung.

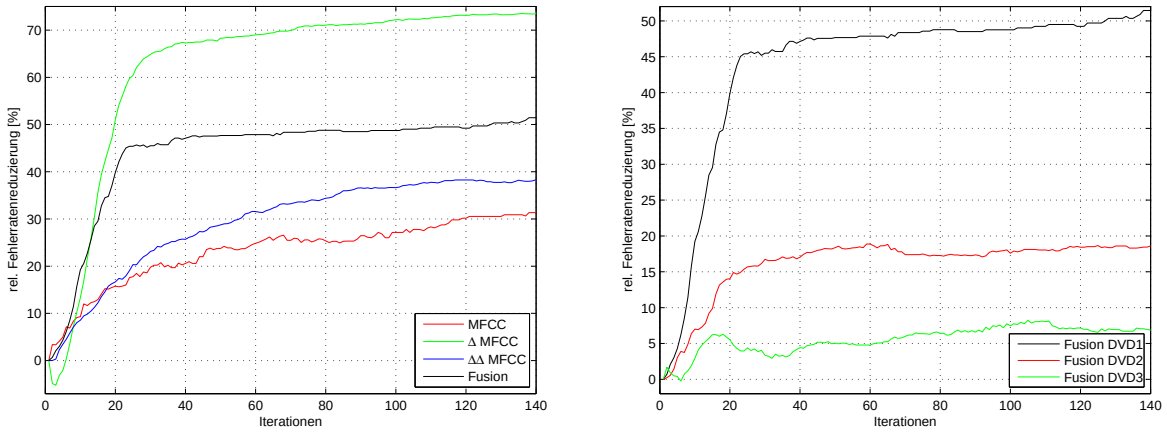
5.3.2 Experimentelle Ergebnisse

In den Experimenten wird untersucht, ob die Klassifikationsrate durch das diskriminative Lernverfahren bei einer gleichbleibenden Komplexität der Modelle verbessert werden kann. Als Referenz wird das beste Modell aus der *ML*-Parameterschätzung verwendet. Die Beobachtung der Fehlerratenänderung bei der Klassifikation der Trainingsdaten während der Parameterschätzung mit *MMI* wird jeweils einen Hinweis auf die möglichen Verbesserungen

durch das diskriminative Lernverfahren liefern. Ein Test der Modelle auf den unabhängigen Testdaten zeigt anschließend, ob die Reduktion der Fehlerrate durch eine verbesserte Modellierung der Ereignisse entstanden ist oder ob eine zu starke Anpassung an die Trainingsdaten vorgenommen wurde.

MMI-Parameterschätzung

Die *MMI*-Parameterschätzung wird mit den Modellen der *ML*-Parameterschätzung initialisiert, welche aus 128 Gauß'schen Mischungsverteilungen je Klasse bestehen. Die experimentellen Ergebnisse der *MMI*-Parameterschätzung sind in Abb. 5.9 dargestellt, wobei die relative Fehlerratenreduktion sich jeweils auf die Klassifikationsergebnisse der mit *ML* geschätzten Mischungsparameter bezieht. In Abb. 5.9 (a) ist die relative Fehlerratenreduktion



(a) Trainingsdaten aufgeschlüsselt nach Merkmalen (b) Vergleich von Trainingsdaten (DVD 1) und Testdaten (DVD 2, DVD 3)

Abbildung 5.9: Fehlerratenreduktion durch die *MMI*-Parameterschätzung von Modellen

auf den Trainingsdaten über die Iterationsschritte aufgetragen. Die höchste Reduktion der Fehlerrate wird mit $73{,}52\,\%$ für die Δ*MFCC*-Merkmale erzielt. Danach folgen die Werte der $\Delta\Delta$*MFCC* mit $38{,}08\,\%$ und der *MFCC* mit $31{,}35\,\%$. In den ersten 25 Iterationen wird der größte Teil der Verbesserungen erreicht, wie aus dem Verlauf der Kurve für die Fusion der Merkmalsvektoren (vgl. Abb. 5.9 (a), „Fusion") entnommen werden kann, jedoch steigen die Kurven selbst für Iterationen oberhalb von 120 noch leicht an. Durch die *MMI*-Parameterschätzung ist es also möglich, die Fehlerrate auf den Trainingsdaten nochmals um die Hälfte gegenüber der *ML*-Parameterschätzung zu senken.

Die auf den Trainingsdaten erreichten Fehlerratenreduktionen sind nicht im gleichen Umfang auf den Testdaten zu erwarten, da ein Teil der Verbesserungen durch eine Überanpassung der Modelle auf die Trainingsdaten entsteht. Speziell die hohe Anzahl der Iterationen lässt die Vermutung aufkommen, dass eine Überanpassung der Modelle vorliegen könnte. In Abb. 5.9 (b) sind daher die Ergebnisse der fusionierten Merkmale ($MFCC + \Delta MFCC + \Delta\Delta MFCC$) für die Trainings- und Testdaten über die Iterationen dargestellt. Erwartungsgemäß fallen die Fehlerratenreduktionen auf den Testdaten der DVD 2 und DVD 3 geringer aus als auf den Trainingsdaten der DVD 1. Jedoch sind für die Daten der zweiten Sitzung eine

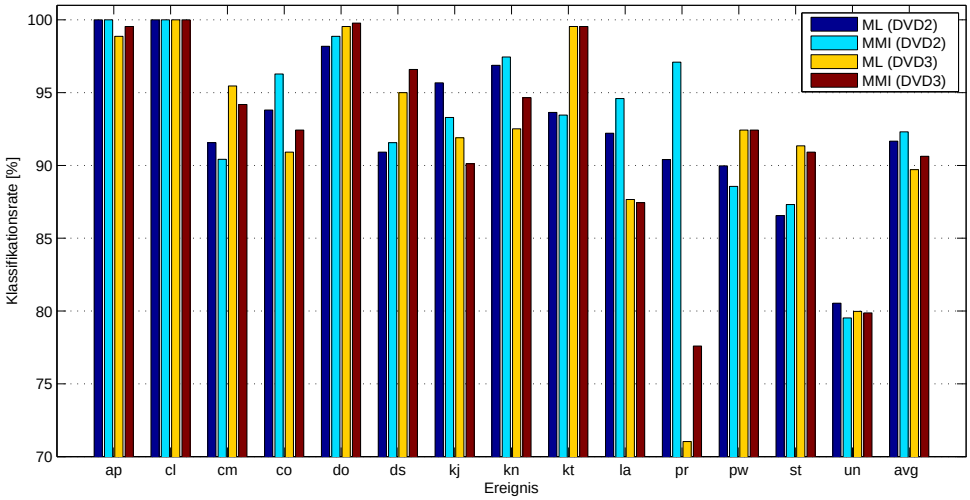


Abbildung 5.10: Vergleich der Klassifikationsraten für Modelle aus der *ML*- und *MMI*-Parameterschätzung auf Testdaten (DVD 2, DVD 3)

relative Fehlerratenreduktion von $18{,}86\,\%$ und für die Daten der dritten Sitzung von $8{,}12\,\%$ zu verzeichnen.

In Abb. 5.10 sind die Klassifikationsraten der Testdaten der zweiten und dritten Sitzung als Vergleich zwischen *ML*- und *MMI*-Parameterschätzung dargestellt. Es zeigt sich hierbei ein nicht einheitliches Bild für die Klassen, da einige besser und einige schlechter erkannt werden. Insgesamt jedoch verbessert sich die mittlere Klassifikationsrate („*avg*“) auf beiden Testdatensätzen.

5.4 Quellenauswahl und Fusion

Die Lokalisation von Sprechern und akustischen Ereignissen erfordert eine gewisse Menge an verteilten Mikrophonen in einem Raum. Dadurch ergibt sich die Möglichkeit, auch für die Identifikation eine Auswahl oder Fusion der verfügbaren Mikrophonsignale vorzunehmen. In dieser Arbeit wird die Fusion nach der Modellbewertung näher betrachtet. Abbildung 5.11

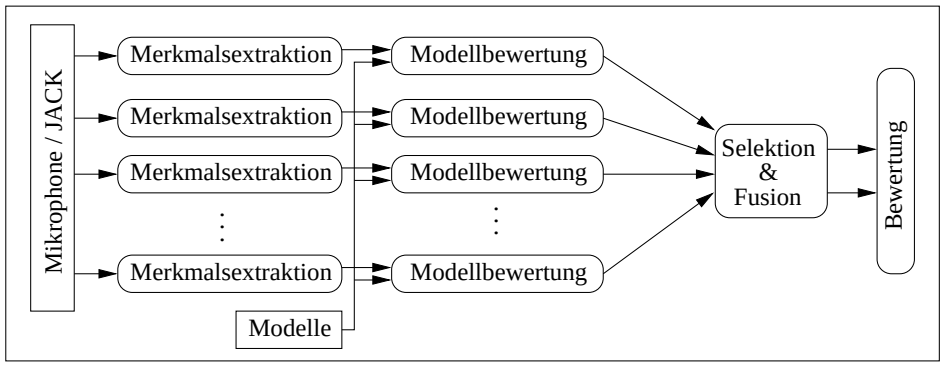


Abbildung 5.11: Fusion und Selektion von *Likelihood*-Werten bei der Ereignisdetektion

zeigt diesen Ansatzpunkt, welcher eine Fusion oder Selektion auf Grundlage der *Likelihood*-Werte im System der akustischen Ereignisdetektion vornimmt. Grundsätzlich wird zunächst

eine Entstörung und Merkmalsextraktion für alle verfügbaren Audiosignale der Mikrophone vorgenommen. Anschließend werden die *Likelihoods* der Merkmalsvektoren mit den vorab trainierten Modellen berechnet. Für die finale Entscheidung, welches Ereignis vorliegt, wird eine Fusion oder eine Selektion der *Likelihoods* oder auch eine Kombination aus beidem vorgenommen.

5.4.1 Ansätze zur Fusion von Modellbewertungen

Die Datenbasis zur Ereignisdetektion beinhaltet Aufnahmen von 22 unabhängigen Mikrophonen, welche in 5 Gruppen angeordnet sind. Da die meisten Ereignisse nur eine geringe zeitliche Dauer aufweisen (z. B. Klopfen) oder keine eindeutige Position im Raum besitzen (z. B. Applaus), ist eine verlässliche Ausrichtung einer Strahlformung auf die Position eines Ereignisses schwierig oder unmöglich. Daher wird auf eine akustischen Strahlformung verzichtet, wie sie bei der Sprecherprotokollierung verwendet wird.

Die Parameterschätzung der Modelle kann prinzipiell auf zwei Weisen erfolgen. Entweder wird für jedes Mikrophon separat ein Satz von Parametern geschätzt, so dass mikrophonspezifische Modelle entstehen, oder sämtliche Daten aller Mikrophone werden zur Schätzung der Modellparameter verwendet. Letzterer Ansatz bedeutet, dass mehr Daten pro Modell zur Parameterschätzung zur Verfügung stehen, da ein Ereignis in 22 leicht variierenden Aufnahmen vorliegt. Experimente mit mikrophonspezifischen Modellen zeigten schlechtere Erkennungsergebnisse als die Verwendung eines mikrophonunabhängigen Modells. Daher wurden die weiteren Experimente mit einem Modell für alle Mikrophone durchgeführt.

Im Anhang A.3 (S. 120) befinden sich die zwei Tabellen Tab. A.1 und Tab. A.2, welche die Motivation für die folgenden Untersuchungen liefern. Beide Tabellen zeigen die Klassifikationsraten der Testdaten aufgeteilt nach den 22 Mikrophonen, so dass die Spannbreite der Klassifikationsraten zwischen den vorliegenden Mikrophonkanälen deutlich wird. Ein Beispiel ist das Ereignis Lachen, welches im Datensatz der DVD 2 vom besten Mikrophon zu $100{,}00\,\%$ (Mikrophon 20) und vom schlechtesten Mikrophon nur zu $80{,}95\,\%$ (Mikrophon 10) richtig klassifiziert wurde. Umgekehrt ist das Mikrophon 20 mit einer Klassifikationsrate von $87{,}50\,\%$ eines der schlechtesten Mikrophone für die Identifikation des Ereignisses Klopfen und das Mikrophon 10 liefert mit einer Klassifikationsrate von $100{,}00\,\%$ eine perfekte Leistung. Ein Mikrophon, welches ein Ereignis schlecht klassifiziert, kann folglich für ein anderes Ereignis optimal sein.

Die Vermutung, dass bestimmte Mikrophone durch ihre Lage vielleicht für einzelne Ereignisse optimal sind, kann durch den Vergleich der Tabellen widerlegt werden. Beispielsweise können die mit dem Mikrophon 10 aufgenommenen Ereignisse Klopfen („*kn*“) als Gegenbeispiel verwendet werden. Im Datensatz der DVD 2 wird dieses Ereignis in allen Aufnahmen des Mikrophons 10 richtig erkannt. Jedoch werden die Aufnahmen von Klopfen im Datensatz der DVD 3 von diesem Mikrophon mit am schlechtesten klassifiziert. Da die Lage der Mikrophone kein Kriterium für eine Selektion ist, werden während der Klassifikation alle Mikrophonaufnahmen gleich behandelt.

Untersucht werden drei Ansätze zur Selektion und Fusion der vorliegenden *Likelihoods*. Alle drei Verfahren sind durch die alleinige Betrachtung der *Likelihood*-Werte unabhängig von der zugrunde liegenden Methode der Modellparameterschätzung und werden sowohl mit den Modellen der *ML*- als auch der *MMI*-Parameterschätzung verwendet. Gegeben seien für jedes Mikrophonsignal m der M Mikrophonsignale eine Menge von N Merkmalsvektoren

$\boldsymbol{X}_{1:N}^{(m)}$, deren Klassenzugehörigkeit mit Ω bezeichnet wird.

Maximum-MAP-Entscheidungsregel

Die optimale Entscheidungsregel ist durch die „Maximum A Posteriori"-Entscheidungsregel (MAP-Entscheidungsregel) gegeben. Da mehr als ein Mikrophon zur Verfügung steht, kann zwar für jedes Mikrophon eine optimale Entscheidung durch die MAP-Entscheidungsregel getroffen werden, jedoch ist dann noch eine Entscheidung auf den 22 Ergebnissen zu treffen. Hierzu wurde die MAP-Entscheidungsregel um einen weiteren $\max$-Operator erweitert (Maximum-MAP), so dass das Maximum aller MAP-Werte über allen Mikrophonen verwendet wird. Die Maximum-MAP-Entscheidungsregel lautet:

$$\widehat{\Omega} \quad = \quad \underset{k,m}{\operatorname{argmax}} \left\{ P(\Omega = k | \boldsymbol{X}_{1:N}^{(m)}) \right\}. \tag{5.12}$$

Es wird also das Mikrophon ausgewählt, deren a posteriori Wahrscheinlichkeiten auf die sicherste Entscheidung hindeuten.

Mehrheitsvotum

Die zweite Entscheidungsregel verwendet ein Mehrheitsvotum über alle Kanäle, um die Entscheidung für eine Klasse zu treffen. Zunächst wird innerhalb eines jeden Kanals eine Hypothese $\widehat{\Omega}^{(m)}$ für das beobachtete Ereignis anhand der MAP-Entscheidungsregel aufgestellt. Anschließend wird die Klasse ausgewählt, welche am häufigsten als Hypothese genannt wurde. Die Entscheidungsregel des Mehrheitsvotums lautet somit:

$$\widehat{\Omega}^{(m)} \quad = \quad \underset{k}{\operatorname{argmax}} \left\{ P(\Omega = k | \boldsymbol{X}_{1:N}^{(m)}) \right\} \tag{5.13}$$

$$\widehat{\Omega}^{(m)} \quad \overset{Mehrheit}{\longrightarrow} \quad \widehat{\Omega}. \tag{5.14}$$

MAP-Produkt-Entscheidungsregel

Die Maximum-MAP-Entscheidungsregel trifft eine Auswahl aus allen Kanälen für die endgültige Entscheidung. Dabei kann ein stark gestörter Kanal mit sehr niedrigen *Likelihood*-Werten zu einer Fehlentscheidung führen, weil durch die Normierung der MAP-Entscheidungsregel die absoluten Werte der *Likelihoods* vernachlässigt werden. Diese Unzulänglichkeit wird im Mehrheitsvotum umgangen, indem die Mehrheit der Entscheidungen betrachtet wird. Hierbei gehen jedoch nur die Werte der *Likelihoods* innerhalb eines Kanals in die Entscheidung ein und nicht ein Vergleich der Werte zwischen den Kanälen. Die MAP-Produkt-Entscheidungsregel versucht diesen Aspekt zu berücksichtigen und eine Fusion der *Likelihood*-Werte aller Kanäle durchzuführen. Unter der Annahme, dass die Merkmalsvektoren der Mikrophone voneinander statistisch unabhängig sind, folgt

$$p(\boldsymbol{X}_{1:N}^{(1)}, \ldots, \boldsymbol{X}_{1:N}^{(M)} | \Omega = k) \quad = \quad \prod_{m=1}^{M} \left(p(\boldsymbol{X}_{1:N}^{(m)} | \Omega = k) \right). \tag{5.15}$$

Ferner sei das Auftreten aller Ereignisse gleich wahrscheinlich, so dass die MAP-Produkt-Entscheidungsregel definiert werden kann durch:

$$\widehat{\Omega} \quad = \quad \underset{k}{\operatorname{argmax}} \left\{ p(\boldsymbol{X}_{1:N}^{(1)}, \ldots, \boldsymbol{X}_{1:N}^{(M)} | \Omega = k) \right\}. \tag{5.16}$$

Die MAP-Produkt-Entscheidungsregel verwendet explizit die Annahme, dass die Merkmalsvektoren an den Mikrophonen statistisch unabhängig voneinander sind. Diese Annahme könnte für weit voneinander entfernte Mikrophone zutreffen, jedoch ist dies für Mikrophone einer Mikrophongruppe womöglich nicht gegeben.

5.4.2 Experimentelle Ergebnisse

Die Experimente verwenden die Datenbasis der akustischen Ereignisidentifikation des Projektes *CHIL*. Die Abb. 5.12 zeigt einen Vergleich der Klassifikationsraten auf den Testdaten

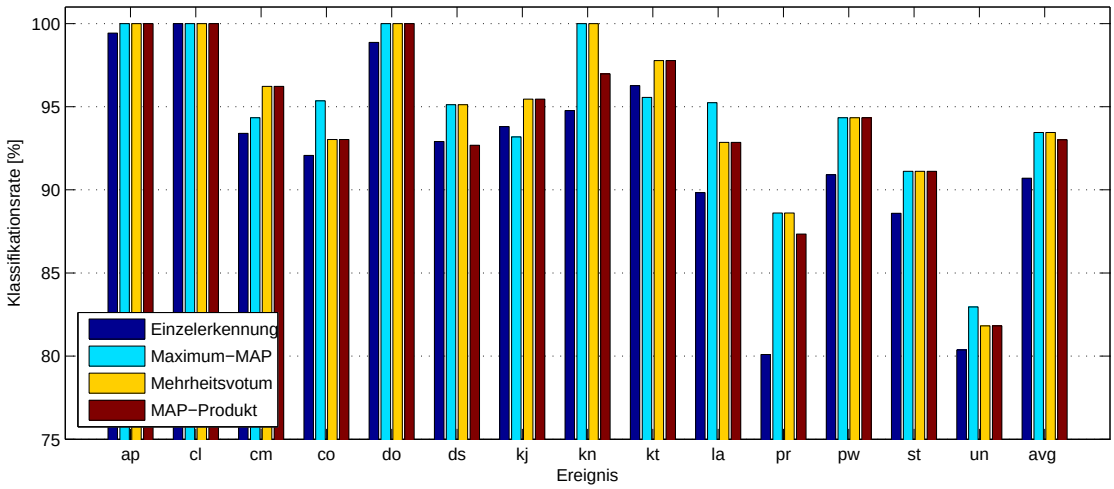


Abbildung 5.12: Vergleich von Auswahlverfahren und Kombinationsansätzen zur akustischen Ereignisidentifikation (*ML*-Parameterschätzung, 128 *GMM*, DVD 2 und DVD 3)

(DVD 2 und DVD 3) zwischen den drei Entscheidungsregeln und einer Einzelerkennung, jeweils aufgeteilt nach den Ereignissen. Dabei sei darauf hingewiesen, dass die Ergebnisse der Einzelerkennung, wie sie aus den vorherigen Kapiteln bekannt sind, jeweils die Klassifikation aller Aufnahmen eines Ereignisses beinhaltet. Die zugrunde liegenden Modelle sind Gauß'sche Mischungsverteilungen mit 128 Verteilungen. Die mittlere Klassifikationsrate („*avg*“) ist in den drei Ansätzen im Vergleich zu den Ergebnissen der Einzelerkennung verbessert worden.

In Tab. 5.1 sind die Klassifikationsraten für verschiedene Ansätze der Modellparameterschätzung gegeben. Es werden die Ergebnisse der *ML*-Parameterschätzung denen des diskriminativen Lernverfahrens durch *MMI* gegenüber gestellt. Dabei wird deutlich, dass die Verbesserung der Modelle durch das diskriminative Training durch die Fusion der *Likelihood*-Werte an Bedeutung verliert. Sowohl die *ML*- als auch die *MMI*-Parameterschätzung liefern vergleichbare Resultate nach der Fusion, wobei die Wahl des Ansatzes, d. h. ob „Maximum-MAP“, „Mehrheitsvotum“ oder „MAP-Produkt“, keinen signifikanten Unterschied macht.

Diskussion und Ausblick

Zuletzt soll das Potential zukünftiger Ansätze für die Verbesserung der Selektion und Fusion von *Likelihoods* anhand eines Experiments untersucht werden. Hierzu werden die *Like-*

Ansatz \ Daten	DVD 2	DVD 3	DVD 2 + DVD 3	Modelle
Einzelerkennung	91,64	89,58	90,70	*ML*
Maximum-MAP	94,29	92,58	93,45	
Mehrheitsvotum	94,57	92,28	93,45	
MAP-Produkt	94,00	91,99	93,01	
Einzelerkennung	93,21	90,43	91,85	*MMI*
Maximum-MAP	94,57	92,58	93,59	
Mehrheitsvotum	94,57	92,28	93,45	
MAP-Produkt	94,57	91,99	93,30	

Tabelle 5.1: Vergleich der Klassifikationsraten für unterschiedliche Trainingsverfahren

lihoods aller Mikrophone für ein Ereignis darauf untersucht, ob ein Mikrophonsignal existiert bei dem das Ereignis richtig identifiziert wird. Bei einer optimalen Wahl eines Kanals würde in diesem Fall das Ereignis richtig erkannt werden. Die Abb. 5.13 zeigt den Vergleich zwischen der Einzelerkennung, dem Mehrheitsvotum und der optimalen Wahl eines Mikrophons. Für einige Ereignisse ist bereits das Maximum der Klassifikationsraten erreicht, falls nicht die zugrunde liegenden Modelle verbessert werden. Die Klassifikationsraten einiger anderer Ereignisse, wie z. B. Papier („*pw*“), könnten jedoch beträchtlich gesteigert werden.

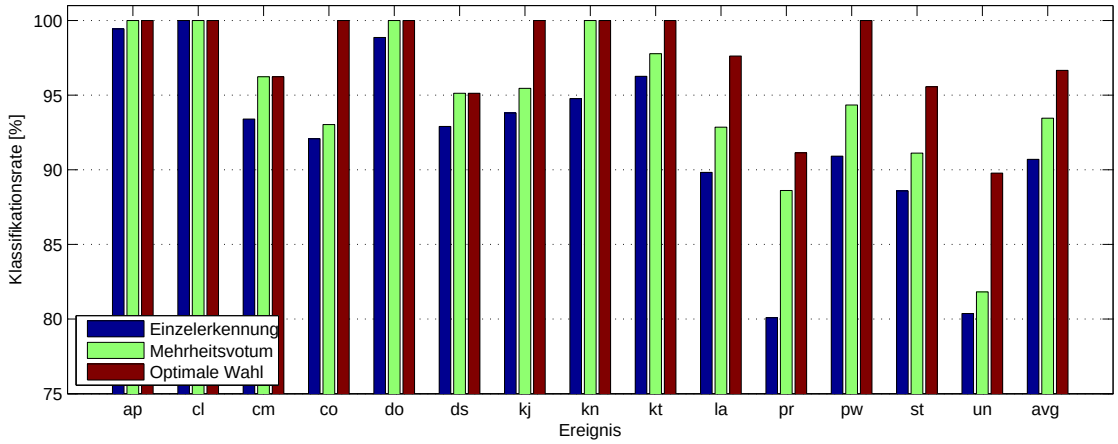


Abbildung 5.13: Vergleich der Klassifikationsraten zwischen Einzelerkennung, Mehrheitsvotum und optimaler Mikrophonwahl auf Testdaten (DVD 2, DVD 3)

6 *Middleware* und ambiente Intelligenz

Die Amigo Architektur orientiert sich an den durch die Vision der ambienten Intelligenz aufgestellten Anforderungen an eine intelligente Hausumgebung [Ami06]. Im vernetzten Haus werden Applikationen und Dienste entsprechend den Bedürfnissen der Nutzer gestartet, konfiguriert, verwendet und beendet. Zusätzlich kann die Ausstattung mit Komponenten zeitlich variieren, da diese in das Haus eingebracht oder aus dem Haus entfernt werden bzw. ihre Position im Haus ändern. Somit ist die vernetzte Hausumgebung durch eine starke Dynamik geprägt, welcher durch die gewählte Architektur Rechnung getragen wird [SBG+05].

Ein weiterer Aspekt ist die Interaktion mit vorhandener *Middleware* und Technologien zur Vernetzung. Das Amigo System verwendet einen semantischen Ansatz, um eine größtmögliche Interoperabilität zu erzielen. Hierbei wird im Amigo System die Bedeutung einer Einheit durch eine Referenz zu einem definierten Vokabular von Ausdrücken (Ontologie) gekapselt, welche ein spezielles Gebiet von Wissen repräsentieren [GMB+05].

Im Folgenden wird gezeigt, wie die Ideen des semantischen Netzes für die vernetzte Hausumgebung genutzt werden können. Anschließend wird die Interaktion zwischen den Diensten mittels *Webservice*-Schnittstellen erklärt und ein Überblick über die Amigo Architektur gegeben. Zum Abschluss wird das Amigo Kontextmanagement anhand des Beispiels der akustischen Szenenanalyse diskutiert.

6.1 Semantisches Netz

Das semantische Netz (engl. *semantic web*) ist als Weiterentwicklung des *World Wide Web* (*WWW*) entworfen worden, um die derzeitigen Unzulänglichkeiten im Umgang mit Informationen zu beheben [B+01]. Seit Erfindung des *Hypertext Transfer Protocols* (*HTTP*) im Jahre 1990 ist das *WWW* auf eine für den Menschen unüberschaubare Größe gewachsen ([ISC07]: Jul 2007, 489.774.269 *Hosts* im *Domain Name System* (*DNS*)). Dadurch ist der Nutzen für den Einzelnen eher begrenzt, obwohl die verfügbare Menge an Informationen gestiegen ist. Erst die Möglichkeit einer durch Maschinen gesteuerten Suche, Verarbeitung und Auswertung wird dem Nutzer einen spürbaren Vorteil bringen [BHL01].

Die vernetzte Hausumgebung bildet wie das *WWW* oder zukünftig das semantische Netz einen Wissensraum mit vielen heterogenen Informationsquellen. Dieses Wissen kann nur durch eine automatische Verarbeitung für den Nutzer erschlossen werden, um „intelligente Systeme“ zu realisieren. Somit ist es naheliegend, in der vernetzten Hausumgebung die Konzepte und Ideen des semantischen Netzes einzusetzen. Im Zentrum des semantischen Netzes stehen die Ontologien, die präsentiertes Wissen für Maschinen annotieren und damit erst für Maschinen verständlich machen.

6.1.1 Ontologien

Eine Ontologie stellt entsprechend [Gru93] eine „explizite formale Spezifikation einer gemeinsamen Konzeptualisierung“ dar. Grundgedanke hierbei ist die Repräsentation einer gemeinsamen Wissensbasis durch die formale Festlegung von Begriffen und deren Relationen. Eine Ontologie soll die für einen Menschen verständlichen Informationen und deren Zusammenhänge Maschinen zugänglich machen, so dass eine maschinelle Verarbeitung und Interpretation möglich wird.

In der Amigo *Middleware* wird die *Web Ontology Language* (*OWL*) [MH04] verwendet, um Ontologien für die vernetzte Hausumgebung zu erstellen. Sie basiert auf dem *Resource Description Framework* (*RDF*), welches eine *Extensible Markup Language* (*XML*) [B+08b] nutzt.

Die Amigo Ontologien sind unter [R+08] verfügbar und frei zugänglich. Sie definieren unter anderem das Vokabular zur Repräsentation von Sensoren, Geräten und Diensten. Die mit diesem Vokabular darstellbaren Kontextinformationen umfassen beispielsweise Sensormesswerte (Temperatur), vorhandene Geräte im Haus (Bildschirm, Kühlschrank), die Zustände der Geräte (Ein, Aus, Standby) und die Fähigkeiten von Diensten (Helligkeitskontrolle, Benachrichtigungsdienst), um an dieser Stelle nur eine Auswahl zu nennen.

Die Nutzung von Ontologien ist nicht begrenzt auf die von Amigo vorgegebenen Vokabulare und kann durch eigene Ontologien ergänzt werden. Somit können auch neue Zusammenhänge, die nicht in den bestehenden Ontologien berücksichtigt wurden, durch das Erstellen und Veröffentlichen einer Ontologie in das System integriert werden. Ist der Kontext einer Information hinreichend durch Ontologien beschrieben, so kann die Information in Form einer *RDF*-Beschreibung im System dargestellt werden.

6.1.2 Kontextinformation

Eine im System vorliegende Kontextinformation wird zum Zwecke der Veröffentlichung den Ontologien entsprechend beschrieben und in ein *RDF*-Modell verpackt. Dabei unterscheidet das *RDF*-Modell allgemein die drei Informationstypen Ressource, Eigenschaft und Objekt. Eine Kombination dieser drei Typen wird als *RDF*-Tripel bezeichnet und stellt eine Aussage über eine Ressource in einer definierten Domäne dar (engl. *statement*) [B+08a]. Ein *RDF*-Modell kann durch einen sprachunabhängigen *RDF*-Graphen repräsentiert werden. Ressourcen werden durch Ellipsen, Eigenschaften durch beschriftete Pfeile und Objekte als Rechtecke gekennzeichnet.

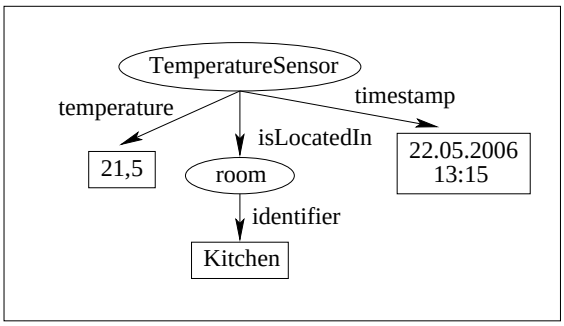


Abbildung 6.1: Beispiel eines *RDF*-Graphen zur Beschreibung einer Temperaturinformation

```
1  <?xml version="1.0"?>
2  <rdf:RDF
3  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4  xmlns:amigo:<http://amigo.org/owl/AmigoICCS.owl#>
5  xmlns:domotic:<http://amigo.org/owl/Domotics.owl#>
6  xmlns:context:<http://amigo.org/owl/ContextTransport.owl#>
7    <domotic:TemperatureSensor>
8       <context:timestamp>
9          2006-05-22T13:15:15.452+0200
10      </context:timestamp>
11      <context:isLocatedIn>
12         <context:room>
13          <context:identifier>
14             Kitchen
15          <context:identifier>
16         </context:room>
17      </context:isLocatedIn>
18      <amigo:temperature>
19         21.5
20      </amigo:temperature>
21   </domotic:TemperatureSensor>
22 </rdf:RDF>
```

Liste 6.1: *RDF*-Beschreibung einer Temperaturinformation

Die Liste 6.1 zeigt ein Beispiel für die *RDF*-Beschreibung einer Temperaturinformation in *XML*-Notation für den *RDF*-Graphen aus Abb. 6.1. Die Aussage der Kontextinformation lautet, dass ein Temperatursensor (*TemperaturSensor*), welcher sich in dem Raum (*isLocatedIn*) mit dem Bezeichner (*identifier*) *Kitchen* befindet, zum angegeben Zeitpunkt (*timestamp*) die Temperatur $21{,}5$ (*temperature*) gemessen hat. Die Zeilen 3-6 der Liste 6.1 beinhalten die Abkürzungen und Verweise auf die verwendeten Ontologien. Der Temperatursensor ist als Gerät in der Ontologie der Hausvernetzung *Domotics.owl* beschrieben. Die kontextbezogenen Zusammenhänge stammen aus der Ontologie *ContextTransport.owl*, und die Beschreibung des Temperaturwertes ist aus der Amigo Ontologie *AmigoICCS.owl* entnommen worden.

Nachdem die Grammatik und das Vokabular zur Darstellung der Informationen durch die Ontologien festgelegt sind, werden nun gemeinsame Definitionen zur Abfrage der Informationen benötigt. Applikationen, die Informationen suchen, benötigen eine definierte Abfragesprache, welche von den Kontextquellen verstanden und verarbeitet werden kann.

6.1.3 Abfragesprache für Kontextinformationen

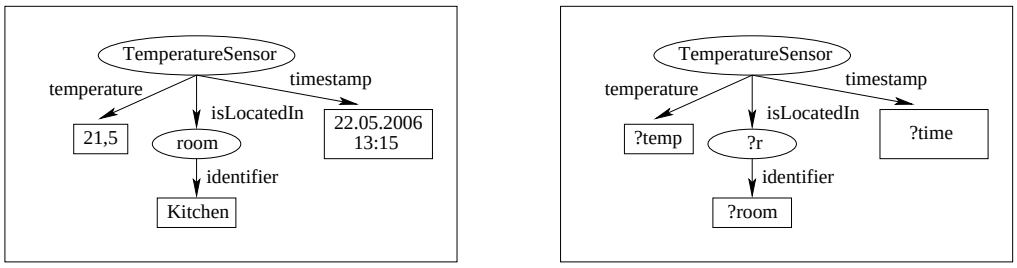
Eine maschinelle Verarbeitung von Informationen benötigt neben der Repräsentation der Daten mittels einer Ontologie auch eine definierte Abfragesprache. Die Amigo *Middleware* verwendet die *SPARQL Protocol and RDF Query Language* (*SPARQL*) [PS08], um Informationen abzufragen. Als Beispiel soll nun eine *SPARQL*-Frage für die Kontextquelle aus Abb. 6.1 vorgestellt werden.

Eine *SPARQL*-Frage gliedert sich in zwei Teile. Zunächst werden über eine Menge von Variablen die Namen der Rückgabevariablen der Objekte festgelegt (Liste 6.2: Zeile 5). Anschließend wird über ein Muster von Tripeln der kontextuelle Zusammenhang der gesuchten Informationen definiert, bei *RDF* sind dies die Ressourcen und Eigenschaften (Liste 6.2:

```
1  PREFIX domotic: <http://amigo.gforge.inria.fr/owl/Domotics.owl#>
2  PREFIX amigo: <http://amigo.gforge.inria.fr/owl/AmigoICCS.owl#>
3  PREFIX context:<http://amigo.gforge.inria.fr/owl/ContextTransport.owl#>
4  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5  SELECT ?room ?temp ?time WHERE {
6   ?id rdf:type domotic:TemperatureSensor .
7   ?id context:isLocatedIn ?r .
8   ?r context:identifier ?room .
9   ?id amigo:temperature ?temp .
10  ?id context:timestamp ?time .}
```

Liste 6.2: Beispiel einer *SPARQL*-Frage nach Temperaturinformationen

Zeile 6-10). Somit kann sowohl gezielt nach Objekten in Informationen als auch nach dem Kontext gefragt werden. Optional können Präfixe zur Verkürzung verwendet werden (Liste 6.2: Zeile 1-4). Die in Liste 6.2 gestellte Frage sucht explizit nach den Kontextinformationen von Temperatursensoren (*domotic:TemperatureSensor*) und möchte neben der Temperaturinformation (*?temp*) auch die Position des Sensors (*?room*) und den Zeitpunkt der Messung (*?time*) wissen.



(a) Beispiel eines *RDF*-Graphen zur Beschreibung einer Temperaturinformation

(b) *SPARQL*-Beispielfrage nach Temperaturinformationen

Abbildung 6.2: Vergleich zwischen Kontextinformation und Kontextabfrage

Vergleicht man die Frage aus Liste 6.2 mit der Information aus Liste 6.1 so kann festgehalten werden, dass die *SPARQL*-Frage eine Art von Sieb für Informationen definiert (vgl. Abb. 6.2). Zum einen werden definierte Ressourcen und Eigenschaften genannt, um die Menge an Kontextquellen einzuschränken. Zum anderen werden durch Platzhalter mehrere Informationen gleichzeitig abgefragt. Durch die Einschränkung der gesuchten Ressource auf Temperatursensoren aus der Heimvernetzung (*domotic:TemperatureSensor*) werden andere Temperaturinformationen, wie zum Beispiel die von Kühlschränken, ausgeschlossen.

Nachdem die Grammatik, das Vokabular, die Beschreibung und die Abfrage von Kontextinformationen beschrieben wurden, wird im Folgenden die Suche nach Kontextquellen und die Interaktion mit ihnen beschrieben. Dienste, die Informationen anbieten, müssen durch eine geeignete Technik im Netz veröffentlicht werden, so dass eine Applikation, die Informationen sucht, diese finden und abfragen kann. Diese Aufgabe eines zentralen Anlaufpunktes übernimmt ein Verzeichnisdienst.

6.1.4 Verzeichnisdienst

Die Aufgabe des Verzeichnisdienstes ist die Speicherung von Informationen über Dienste und deren Referenzen im Amigo System. Dabei verwaltet der Verzeichnisdienst eine hierar-

chisch strukturierte Datenbank von Informationen. Dienste können nach dem *Server-Client*-Prinzip auf diese Daten mittels eines festgelegten Protokolls zugreifen. Im Amigo System wird das von der *International Telecommunication Unit* (*ITU*) standardisierte *Lightweight Directory Access Protocol* (*LDAP*) [Z+06] der X.500 Architektur [ITU01] verwendet. Die Amigo *Middleware* stellt geeignete Methoden zur Suche von Diensten basierend auf *LDAP* zur Verfügung.

Hat eine Applikation einen geeigneten Dienst über den Verzeichnisdienst gefunden, so ist der nächste Schritt die Interaktion mit dem Dienst. Dies kann zum einen die Abfrage von Informationen sein (Beispiel: Temperatursensor) oder zum anderen das Auslösen von Aktionen durch den Dienst (Beispiel: Anschalten einer Lampe). Im Amigo System werden zur Interaktion *Webservice*-Schnittstellen verwendet.

6.2 *Webservice*

Die vom Amigo System im Netz bereitgestellten Dienste besitzen *Webservice*-Schnittstellen [WWW02], um Methoden für Applikationen oder Dienste bereitzustellen. Die Beschreibung der Schnittstellen kann semantisch mit der im Projekt Amigo entwickelten Sprache *Amigo-S* oder rein syntaktisch mit der *Web Services Description Language* (*WSDL*) [C+07] erfolgen. *Amigo-S* ist eine verallgemeinerte Form der *Web Ontology Language for Web Services* (*OWL-S*) [DAM06], die gegenüber der *OWL-S* um Klassen und Eigenschaften für die Unterstützung von *Quality of Service* (*QoS*) und das Kontextbewusstsein erweitert wurde [MKGI07].

Jeder Amigo Dienst wird mit dem *Uniform Resource Name* (*URN*) „urn:amigo“ im Amigo System gekennzeichnet. Eine *URN* [M+97] ist eine dauerhafte, ortsunabhängige Bezeichnung einer Ressource, die das Schema *Uniform Resource Identifier* (*URI*) vom Typ „*urn*“ [B+05a] verwendet.

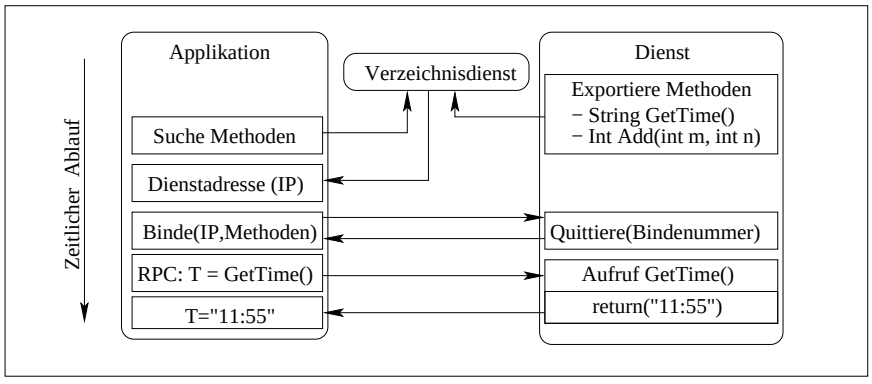


Abbildung 6.3: Interaktion zwischen Applikation und Dienst mittels *Webservices*

In Abb. 6.3 ist die zeitliche Abfolge der Kommunikation bei der Verwendung eines exportierten *Webservices* durch eine Applikation gezeigt. Zunächst exportiert der Dienst seine zwei Methoden (*GetTime* und *Add*), indem er sie beim *LDAP*-Verzeichnisdienst registriert. Eine Applikation kann diese Methoden über den Verzeichnisdienst suchen und die Adresse des Dienstes ermitteln. Anschließend bindet die Applikation die Methoden an die Adresse

und erhält als Quittung die Bindenummer vom Dienst. Durch einen *Remote Procedure Call* (*RPC*) kann nun die Applikation die Methoden des Dienstes verwenden.

6.3 Amigo Architektur

Grundsätzlich gliedert sich die Amigo Architektur in vier Schichten: Plattform, *Middleware*, intelligente Dienste und Applikationen (vgl. Abb. 6.4) [J+05]. Diese Schichten werden in den folgenden Kapiteln näher betrachtet, wobei deren Aufgaben, Funktionen und Schnittstellen spezifiziert werden.

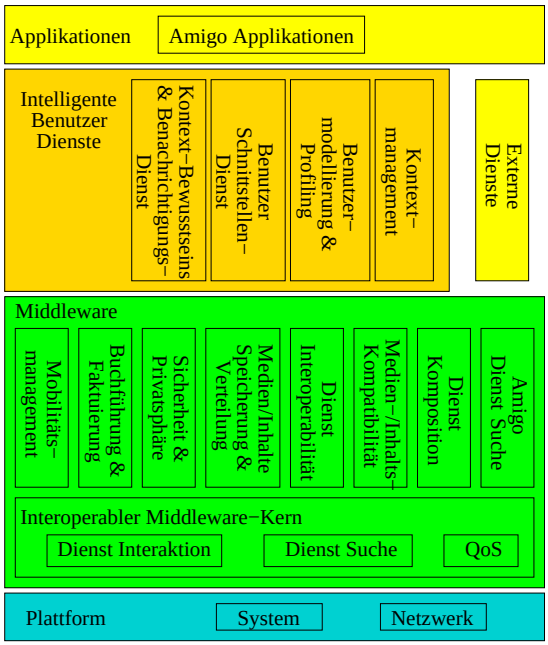


Abbildung 6.4: Spezifikation der Amigo Architektur gemäß [J+05]

6.3.1 Plattform

Die vorhandenen Plattformen in einer vernetzten Hausumgebung stellen eine heterogene Umgebung für die Verwendung von Software dar. Dabei variieren sie in den Bereichen Speicher, Rechenleistung, Betriebssystem, Benutzer- und Netzwerkschnittstellen. Das Spektrum der anvisierten Geräte, auf denen die Amigo *Middleware* eingesetzt werden soll, reicht von Haushaltsgeräten über Smartphones, Notebooks bis hin zur Unterhaltungselektronik. Diese Geräte nutzen neben den verbreiteten Betriebssystemen Windows, Linux, Windows Mobile und Symbian OS auch zum Teil hardwarespezifische Softwareumgebungen.

Eine hoher Anteil an Hardwareplattformen wird im Projekt Amigo durch die Verwendung der auf Java basierenden „*Open Services Gateway Initiative*“-Plattform (*OSGI*-Plattform) [OSG08] abgedeckt. Entwickler können zudem optional unter Windows mit dem *.net-Framework* Applikationen und Dienste erstellen. Die *OSGI*-Laufzeitumgebung eignet sich für die plattformübergreifende Entwicklung von Software, da sie auf allen Geräten mit einer *Java Virtual Machine* und ausreichenden Ressourcen ausgeführt werden kann [SS07].

Eine Applikation auf der *OSGI*-Plattform gliedert sich in Softwarepakete (engl. *Bundles*), deren Lebenszyklen durch die Zustände „Installiert, Startend, Aktiv, Stoppend, Aufgelöst und Entfernt“ festgelegt sind. Hierbei teilen sich die Applikationen auf einer *OSGI*-Plattform die vorhandenen Ressourcen und können applikationsübergreifend aktive *Bundles* und deren exportierte Klassen nutzen. Jedes *Bundle* verfügt über einen Lademechanismus für Klassen (engl. *Class Loader*), welcher den Speicherbereich der Klassen (engl. *Class Space*) verwaltet. In diesem Speicherbereich sind drei Arten von Klassen vorhanden:

- Private Klassen: Exklusiv durch das *Bundle* genutzte und bereitgestellte Klassen.
- Importierte Klassen: Klassen, die von anderen *Bundles* bereitgestellt werden.
- Exportierte Klassen: Klassen, die für andere *Bundles* bereitgestellt werden.

Zusätzlich existieren Mechanismen zum Installieren, Starten, Stoppen, Aktualisieren und Löschen der *Bundles*. Diese Verwaltungsmechanismen sind besonders im Bereich der Geräte mit eingeschränkter Benutzerschnittstelle notwendig, um eine Fernwartung zu ermöglichen.

6.3.2 Amigo *Middleware*

Oberhalb der Plattformschicht ist die Amigo *Middleware* mit ihrem interoperablen Kern angesiedelt. Eine der Schlüsseltechnologien des Amigo Systems ist die nahtlose Integration von heterogenen Strukturen im Bereich etablierter *Middleware* (z. B. *UPnP*) und Geräten in der vernetzten Hausumgebung. Diese Interoperabilität wird mit Hilfe des interoperablen *Middleware*-Kerns realisiert.

Interoperabler *Middleware*-Kern

Eine *Middleware* muss zum einen Funktionen zur Bekanntmachung und zur Suche von Diensten im *Service Discovery Protocol* (*SDP*) definieren. Zum anderen müssen Methoden zur Interaktion im *Service Interaction Protocol* festgelegt werden. Beide Protokolle sind *Middleware* spezifisch und im Allgemeinen zwischen zwei *Middleware*-Technologien nicht austauschbar.

Im Amigo System ist die Aufgabe des *Middleware*-Kerns, eine für die Dienste transparente Interoperabilität zu schaffen. Dabei vermittelt das „*SDP-Detection and Interoperability*“-Protokoll (*SDI*-Protokoll) [BI05] die Suchanfragen und Antworten, und das „*Service Interaction Interoperability*“-Protokoll (*SII*-Protokoll) ermöglicht die Interaktion [SBG+05]. Interoperabilität bedeutet in diesem Zusammenhang, dass zwei unterschiedliche *Middleware*-Technologien miteinander kommunizieren und interagieren, als ob beide die gleichen Protokolle verwenden würden.

In Abb. 6.5 ist das Beispiel aus [SBG+05] gegeben, welches die Kommunikation zwischen einem mobilen Gerät (*Personal Digital Assistent, PDA*) und einem Medienserver zeigt. Das mobile Gerät verwendet das *Service Location Protocol* (*SLP*) und die *Remote Method Invocation* (*RMI*), und der Medienserver nutzt *Universal Plug and Play* (*UPnP*) mit dem *Simple Service Discovery Protocol* (*SSDP*) und das *Simple Object Access Protocol* (*SOAP*). Dieses Beispiel wird hier vorgestellt, um die Realisierung der vielfach geforderten Interoperabilität durch die Amigo *Middleware* zu erläutern. Zunächst initiiert der Benutzer über seinen *PDA* durch eine *SLP*-Anfrage eine Suche nach Medienservern im Netz. Diese Anfrage wird

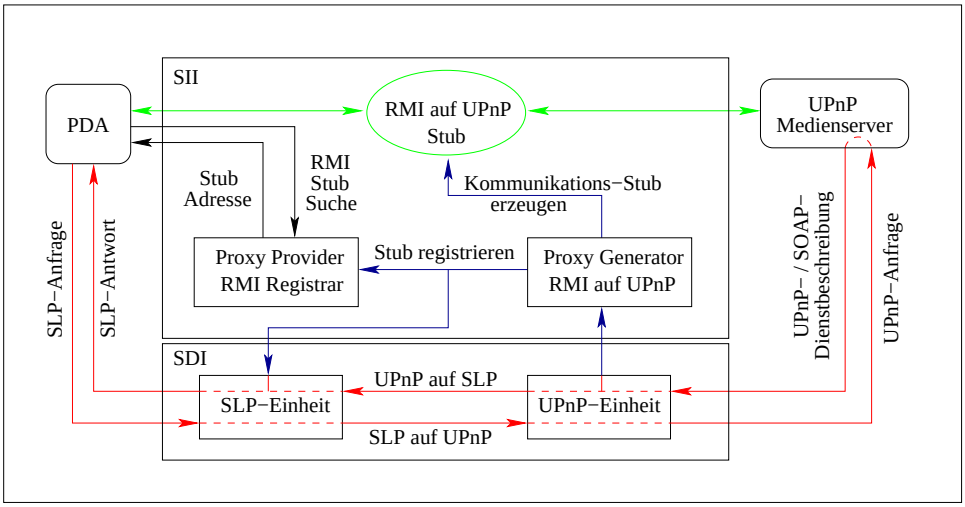


Abbildung 6.5: Amigo interoperabler *Middleware*-Kern

von der *SLP*-Einheit an die *UPnP*-Einheit weitergegeben und mittels der *SDI*-Einheit vom *SLP*-Protokoll auf das *SSDP*-Protokoll für *UPnP* übersetzt. Die *UPnP*-Einheit erhält vom Medienserver als Antwort eine Beschreibung der verfügbaren Dienste mittels des *SOAP*-Protokolls. Anschließend veranlasst die *UPnP*-Einheit den *Proxy Generator*, einen *RMI* auf *UPnP* Kommunikations-*Stub* zu erstellen und diesen sowohl beim *Proxy Provider* als auch bei der *SLP*-Einheit zu publizieren. Der *PDA* wird von der *SLP*-Einheit über die Verfügbarkeit des *RMI-Proxy* informiert. Die Adresse des *Stubs* wird vom *Proxy Provider* geliefert und der *PDA* kann transparent über den *Stub* mit dem Medienserver kommunizieren, als ob beide die gleichen *Middleware*-Technologien verwenden würden.

Middleware

Die Amigo *Middleware* ist verantwortlich für die Bereitstellung von Grundfunktionen zur Dienstsuche, Komposition und Interoperabilität. Des Weiteren sind Medien- und Inhaltsdienste für die Unterhaltungselektronik in der *Middleware* implementiert, wie z. B. die Speicherung und Verteilung von Medien. Entsprechend der Nutzerstudien aus [M$^+$05] sind Dienste zum Schutz der Sicherheit und der Privatsphäre in der *Middleware* verankert. Ein Dienst zum Mobilitätsmanagement unterstützt Nutzer bei der Verwendung mobiler Geräte.

6.3.3 Intelligente Dienste

Die intelligenten Benutzerdienste im Amigo System nutzen die Amigo *Middleware*, um Grundfunktionen für die Entwicklung von Applikationen in der vernetzten Hausumgebung bereitzustellen [J$^+$05]. Eine der Kernaufgaben ist die Verwaltung und Verarbeitung von Kontextinformationen, um Diensten automatisierte und intelligente Entscheidungen zu ermöglichen. Zusätzlich wurden Dienste implementiert, die z. B. bei der Erstellung von Benutzerschnittstellen hilfreich sind. Im Folgenden werden die wichtigsten Dienste erläutert.

Informationen über Benutzer und ihre Gewohnheiten werden durch die Benutzermodellierung bereitgestellt. Dieser Dienst erstellt eine Datenbank über Benutzer und macht diese über eine *Webservice*-Schnittstelle anderen Diensten zugänglich. Jedes Benutzermodell star-

tet mit einem Stereotypenmodell, bei dem ein minimaler Satz von Standardeigenschaften angewendet wird.

Der Kontextbewusstseins- und Benachrichtigungsdienst stellt Dienste für die automatisierte Benachrichtigung bei Eintreten eines Ereignisses oder einer Kombination von Ereignissen bereit [ECB06]. Applikationen können hierfür Regeln definieren und beim Dienst hinterlegen. Dieser überwacht die Kontextquellen im System und benachrichtigt die Applikation, sobald eine hinterlegte Regel erfüllt ist.

6.4 Kontextmanagement

Der Amigo Kontextmanagementdienst (engl. *Context Management Service, CMS*) stellt eine offene Infrastruktur für das Austauschen von Kontextinformationen bereit [RPS+07]. Hierbei werden sowohl Informationen über physikalische Sensoren, Benutzeraktivitäten oder ausgeführte Applikationen als auch deren Zustände verarbeitet und bereitgestellt. Informationen, die aus der Kombination von unterschiedlichen Quellen oder deren Abstraktion entstehen, werden dabei als Kontextinformationen bezeichnet. Eine Applikation kann diese Kontextquellen über den Kontextmanagementdienst nutzen und somit zu einer kontextbewussten Applikation werden.

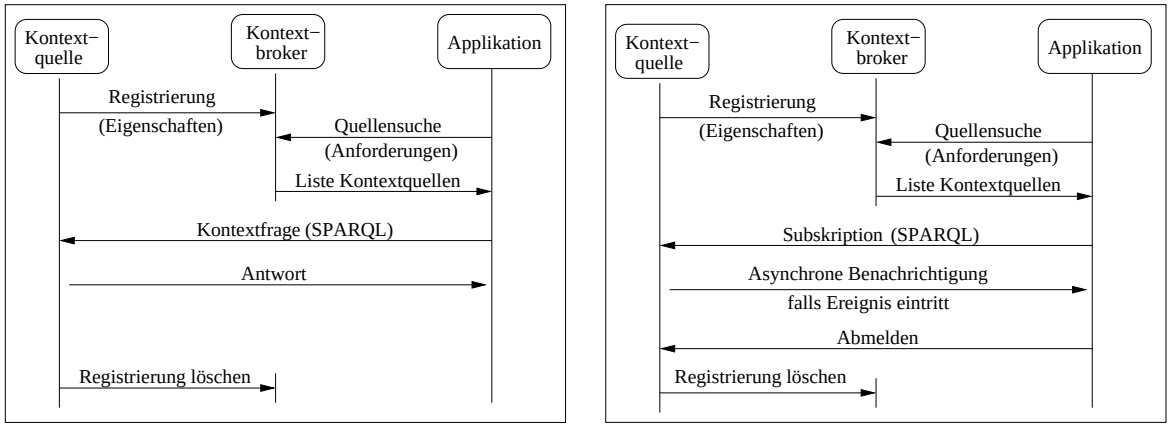
Das System zum Kontextmanagement beinhaltet drei Arten von Komponenten: Kontextquellen, Kontextnutzer und Kontextbroker. Eine Quelle stellt dabei den Nutzern Kontextinformationen zur Verfügung, wobei der Broker als zentrale Vermittlungsstelle zwischen diesen fungiert.

6.4.1 Schnittstellendefinition und Kommunikation

Das Projekt Amigo hat durch die Entwicklung der Amigo *Middleware* eine offene Lösung für die Vernetzung von Diensten in der häuslichen Umgebung geschaffen. Innerhalb dieser *Middleware* nutzen Dienste definierte Verfahren zur Dienstsuche (vgl. Kap. 6.2) und standardisierte Schnittstellen für die Kommunikation. Eine von diesen Schnittstellen ist die *IContextSource*-Schnittstelle, welche einen Satz von vier *Webservice*-Methoden für das Amigo Kontextmanagementsystem definiert. Kontextquellen und Kontextnutzer müssen diesen Satz von Methoden implementieren, um im Kontextmanagementsystem miteinander kommunizieren zu können [J+05].

Für die synchrone Kommunikation (vgl. Abb. 6.6 (a)) ist auf der Seite der Kontextquelle die *query*-Methode zu implementieren, welche als Übergabeparameter die *SPARQL*-Frage nach der Kontextinformation erwartet und als Rückgabewert die Antwort auf die *SPARQL*-Frage liefert. Die asynchrone Kommunikation (vgl. Abb. 6.6 (b)) erfordert drei Methoden. Dies sind auf der Seite der Kontextquelle die *subscribe*-Methode und die *unsubscribe*-Methode und auf der Seite des Kontextnutzers die *notify*-Methode.

Die Kommunikation zwischen Kontextquelle und Kontextnutzer kann auf zwei Arten erfolgen. In Abb. 6.6 (a) ist zunächst die synchrone Kommunikation dargestellt. Hierbei registriert sich die Kontextquelle mit einer Beschreibung ihrer Eigenschaften beim Kontextbroker und hinterlegt die Adresse zum Aufruf ihrer *Webservice*-Methoden. Eine Applikation kann zunächst den Kontextbroker durch ein *Webservice-Lookup* im Netzwerk finden und anschließend eine Quellensuche durch die Spezifikation der Anforderungen an die Quelle



(a) Synchrone Kommunikation (b) Asynchrone Kommunikation

Abbildung 6.6: Kommunikation zwischen Kontextquelle und Applikation

eingrenzen. Die Applikation stellt dann eine auf *SPARQL* basierende Kontextfrage, worauf die Kontextquelle direkt antwortet. Dieses Kommunikationsverfahren eignet sich zum direkten Abfragen von Informationen. Es ist jedoch weniger geeignet, falls die Applikation auf ein bestimmtes Ereignis reagieren soll. Ein kontinuierliches Abfragen von Kontextquellen erzeugt entweder eine hohe Last durch häufige Anfragen oder hat eine hohe Latenz bis die Änderungen bekannt werden, falls die Applikation nur selten Anfragen stellt.

Eine Beobachtung von Sensoren ohne zyklisches Abfragen der Kontextquelle kann durch die asynchrone Kommunikation erfolgen (vgl. Abb. 6.6 (b)). Die Applikation fordert wie im synchronen Fall die Liste der Kontextquellen an. Bei diesen führt sie eine Subskription mit einer *SPARQL*-Frage durch und übergibt dabei die Adresse der *Webservice*-Methode (*notify*-Methode), welche die Kontextquelle zur Benachrichtigung verwenden soll. Als Rückgabewert erhält die Applikation eine eindeutige Identifikationsnummer für die Registrierung, welche in der *unsubscribe*-Methode verwendet wird, um die Subskription rückgängig zu machen. Findet nun ein Ereignis statt, welches zur *SPARQL*-Frage der Applikation passt, so wird diese über die neuen Kontextinformationen informiert. Hierzu nutzt die Kontextquelle die *notify*-Methode der Applikation, deren Funktionsparameter auf die Antwort der *SPARQL*-Frage gesetzt wird.

6.4.2 Kontextbewusste Applikationen

Eine Applikation wird von einem Benutzer als „intelligent“ wahrgenommen, falls die von der Applikation getroffenen Entscheidungen dem Nutzer sinnvoll erscheinen. Hierzu benötigt diese Zugriff auf Kontextinformationen, so dass die Applikation den aktuellen Kontext erfassen kann. Die verfügbaren Kontextinformationen werden in der Applikation miteinander verknüpft und anhand von Entscheidungsregeln ausgewertet. Anschließend kann die Applikation eine kontextbewusste Entscheidung treffen, welche vom Nutzer als „intelligent“, im Sinne von kontextabhängig, wahrgenommen wird.

Die Idee des Amigo Systems ist, zunächst jede Art von Information durch eine Kontextquelle zu abstrahieren und diese anschließend miteinander zu verknüpfen. Dies kann durch Dienste erfolgen, die Informationen eines Typs bündeln und sie als neue Kontextquelle wieder verfügbar machen. Als Beispiel kann hier der *Location Management Service* (*LMS*) ge-

nannt werden. Dieser Dienst kombiniert die unterschiedlichen Positionsinformationen aus beispielsweise *RFID*-Systemen, akustischen Lokalisierungstechniken und anderen Quellen in einer zentralen Datenbank und stellt anschließend diese Datenbank als Kontextquelle anderen Applikationen zur Verfügung.

Ein weiterer Ansatz zum Aufbau „intelligenter Applikationen“ ist die semantische Suche nach Kontextquellen im vernetzten Haus mit Hilfe der Amigo *Middleware* und der Verknüpfung der verfügbaren Informationen in der Applikation selbst. Eine Applikation wird als kontextbewusste Applikation bezeichnet, falls ein Teil der Entscheidungen automatisiert durchgeführt wird und dabei auf Kontextinformationen beruht und nicht nur auf Eingaben eines Benutzers.

6.4.3 Akustische Szenenanalyse als Kontextquelle

Die akustische Szenenanalyse nutzt die Signale der im Haus verteilten Mikrophongruppen, um gleichzeitig Positionsschätzungen und Identifikationen von Personen und Ereignissen durchzuführen. Die hierbei generierten Kontextinformationen werden Diensten im Amigo System zur Verfügung gestellt.

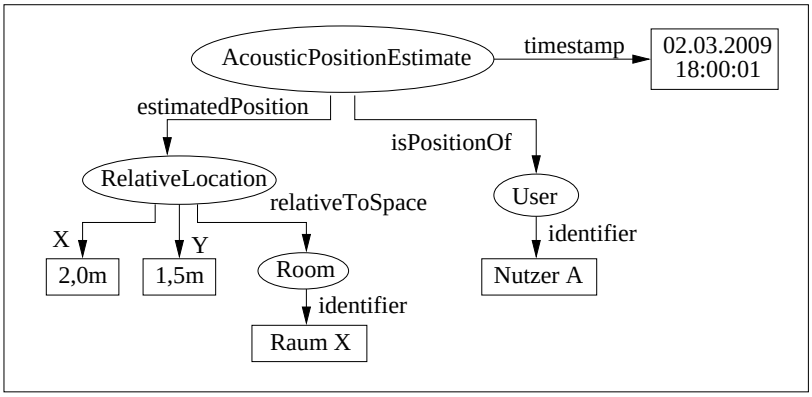


Abbildung 6.7: Beispiel einer Kontextinformation der akustischen Szenenanalyse

In Abb. 6.7 ist beispielhaft eine Kontextinformation der akustischen Szenenanalyse für eine Personenlokalisation dargestellt. Zur Vereinfachung des Graphen wurden die Präfixe der Ontologie weggelassen, welche in [R$^+$08] definiert ist. Die enthaltene Kontextinformation sagt aus, dass der Nutzer A sich zum angegebenen Zeitpunkt im Raum X an der Stelle $X = 2{,}0\,\mathrm{m}$ und $Y = 1{,}5\,\mathrm{m}$ befand.

Betrachtet man das gesamte Aufgabenspektrum der akustischen Signalverarbeitung, so muss neben der akustischen Szenenanalyse auch der Aspekt der Kommunikation berücksichtigt werden. Da die akustische Szenenanalyse nicht nur die Signale analysiert, sondern auch eine Störgeräuschunterdrückung durchführt, sollten folglich zur Rechenzeitersparnis die entstörten Signale der akustischen Szenenanalyse für die Kommunikation genutzt werden. Um Überlastungen des Systems und infolgedessen Aussetzer des Audiodatenstroms während der Kommunikation vorzubeugen, wird die Bereitstellung von Kontextinformationen aus der akustischen Szenenanalyse (ASA) durch das gesonderte *OSGI-Bundle „OSGI:ASA“* auf einer *OSGI*-Plattform durchgeführt. Dieses *Bundle* wird durch eine Interprozesskommunikation auf Basis eines *UDP*-Datenstroms mit dem Modul der Sprecherprotokollierung verbunden.

Nachdem nun die Architektur der Amigo *Middleware* und die verfügbaren Dienste vorgestellt wurden, wird im folgenden Kapitel die Realisierung der ambienten Kommunikation auf Basis des Amigo Systems dargestellt. Diese Anwendung ist ein Beispiel für einen kontextbewussten Dienst, der unabhängig von expliziten Benutzereingaben Entscheidungen trifft und somit als ein Schritt in die Richtung von ambienter Intelligenz angesehen werden kann.

7 Ambiente Kommunikation

Das Konzept der ambienten Intelligenz beschreibt das Entfernen von Geräten aus dem Umfeld der Benutzer bei gleichzeitiger Bereitstellung der zuvor durch die Geräte verfügbaren Dienste [AM04]. Überträgt man dieses Konzept auf den Bereich der Kommunikation, bedeutet dies ein Entfernen der klassischen Kommunikationsgeräte, wie z. B. des Telefons, und den Übergang von der geräteorientierten Kommunikation zur Freisprechfunktionalität. Der Benutzer muss nun nicht mehr ein Telefon zur Kommunikation aufsuchen und mit sich tragen, sondern kann jederzeit auch ohne Gerät kommunizieren [SLH08].

Ein wichtiger Aspekt der ambienten Kommunikation, welcher aus der Forderung nach einer freien Kommunikation folgt, ist die Realisierung von sog. *Follow-Me*-Fähigkeiten. Unter dem Begriff „*Follow-Me*“ wird im Rahmen dieser Arbeit die Fähigkeit des Systems beschrieben, eine Kommunikation dem Benutzer automatisch und somit kontextabhängig folgen zu lassen. Ein Benutzer kann eine Kommunikation in einem Raum starten und sich anschließend frei in seiner Wohnumgebung bewegen, während das System dafür sorgt, dass das Gespräch automatisch dem Benutzer folgt. Hierdurch treten die technischen Randbedingungen der Kommunikation in den Hintergrund, während der Benutzer seinen täglichen Arbeiten nachgeht.

Im Folgenden werden das *Seamless Audio Interface* (*SAInt*) und seine Komponenten vorgestellt, welches zur Realisierung einer ambienten Kommunikation verwendet werden kann. Nach der Vorstellung der Systemarchitektur und der Integration in die *Middleware* werden die grundlegenden Module zur Signalverarbeitung erläutert. Zum Abschluss wird die Erweiterung des Systems um Komponenten zur audio-visuellen Kommunikation diskutiert. Um eine klare Trennung zwischen dem *Middleware*-Dienst und der signalverarbeitenden Komponente vorzunehmen werden folgende Begriffe verwendet: Der „*SAInt*-Dienst“ wird für das *OSGI-Bundle* von *SAInt* verwendet, welches für die Kommunikation mit der *Middleware* verantwortlich ist. Das „*SAInt*-Modul“ bezeichnet das *Spark*-Modul[1], welches als Teil der Signalverarbeitung für das *Routing* der Audiodaten und die Echtzeitkommunikation verantwortlich ist.

7.1 Systemarchitektur und *Middleware*-Integration

Die Systemarchitektur der ambienten Kommunikation, dargestellt in Abb. 7.1, teilt sich auf in die vier Bereiche Hardware, Signalverarbeitung, Echtzeitkommunikation und kontextabhängige Steuerung. Der Begriff Hardware umfasst die verteilten Mikrophone und Laut-

[1] Das *Speech processing and recognition toolkit* (*Spark*) ist eine modulare Software des Fachgebietes Nachrichtentechnik zur digitalen Signalverarbeitung auf Computern.

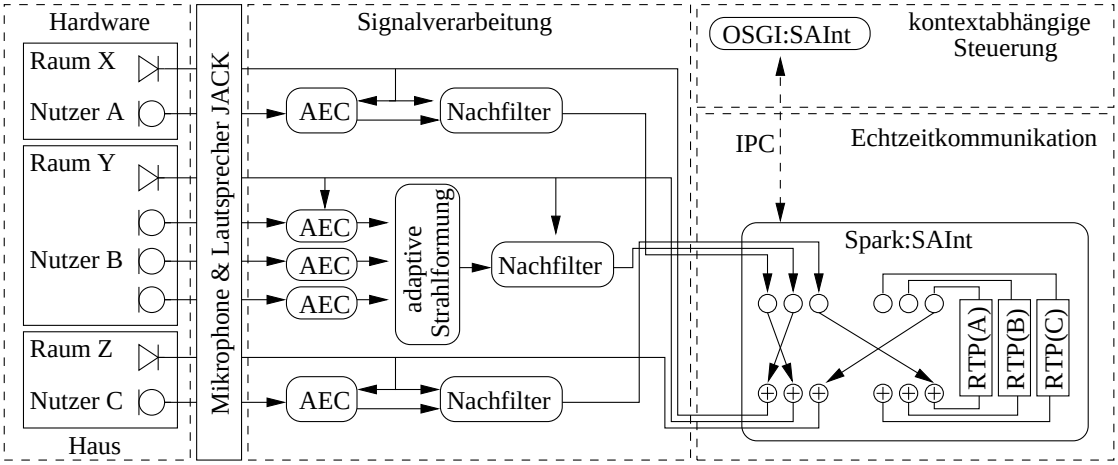


Abbildung 7.1: Blockschaltbild der Systemkomponenten der ambienten Kommunikation

sprecher im Haus, die entweder in Wänden oder Geräten integriert sind, sowie die zu deren Betrieb notwendigen Verstärker und Analog-Digital/Digital-Analog-Wandler (AD/DA-Wandler). Die Schnittstelle zwischen der Hardware und der Software wird mittels dem *Jack Audio Connection Kit* (*JACK*) [JAC08] realisiert, um eine geringe Latenz an der Schnittstelle zwischen Hardware und Software (*HW/SW*) zu erzielen.

Die Signale aus den Mikrophonen werden in der Signalverarbeitung einer Echounterdrückung und einer Störgeräuschfilterung sowie gegebenenfalls einer adaptiven Strahlformung unterzogen. Die Echounterdrückung teilt sich hierbei in die adaptive Echounterdrückung (engl. *Adaptive Echo Canceler, AEC*) und in ein Nachfilter zur Reduktion der verbliebenen Restechos auf. Innerhalb des Nachfilters wird neben der Unterdrückung der Restechos auch die Unterdrückung der Störgeräusche durchgeführt. Falls mehrkanalige Aufnahmen aus Mikrophongruppen verwendet werden, so muss vor der adaptiven Strahlformung die Unterdrückung der Echos erfolgen.

Bei der echtzeitfähigen Kommunikation können zwei Fälle unterschieden werden. Dies ist zum einen die interne Kommunikation, bei der eine Verbindung zwischen zwei Personen im selben Haus aufgebaut wird. Zum anderen ist es die externe Kommunikation zwischen einer lokalen Person und einer entfernten Person. Das *SAInt*-Modul muss im ersten Fall die Daten wie ein Router zwischen den Räumen austauschen. Für den zweiten Fall, dass ein Kommunikationspartner nicht im Haus ist, verbindet das *SAInt*-Modul die Nutzer über eine „*Internet Protocol*“-Verbindung (*IP*-Verbindung) mittels des *Real-Time Transport Protocols* (*RTP*). In Abb. 7.1 sind beispielhaft eine lokale Verbindung zwischen den Nutzern A und B sowie eine externe Verbindung des Benutzers C aus dem Raum Z dargestellt. Die Signalverarbeitung des *SAInt* ist in der Lage, mehrere Verbindungen gleichzeitig zu unterstützen. Es ist als fortlaufend aktives System konzipiert, um mögliche Verzögerungen durch Startzeiten auszuschließen. Da es zudem dauerhaft die Signalverarbeitung für alle Räume durchführt, ist die Systemauslastung konstant und nicht durch Lastspitzen geprägt.

Integration in die Amigo *Middleware*

Die für die Steuerung der Kommunikation benötigten Kontextinformationen werden aus der Amigo *Middleware* bezogen. Der *SAInt*-Dienst registriert sich hierzu bei den benötig-

ten Kontextquellen mit Hilfe des Kontextbrokers und baut eine Interprozesskommunikation (engl. *Inter Process Communication, IPC*) zum *SAInt*-Modul auf. Zusätzlich werden über diese *IPC*-Schnittstelle in der umgekehrten Richtung die gewonnenen Kontextinformationen anderen Applikationen und Diensten im Amigo System zur Verfügung gestellt.

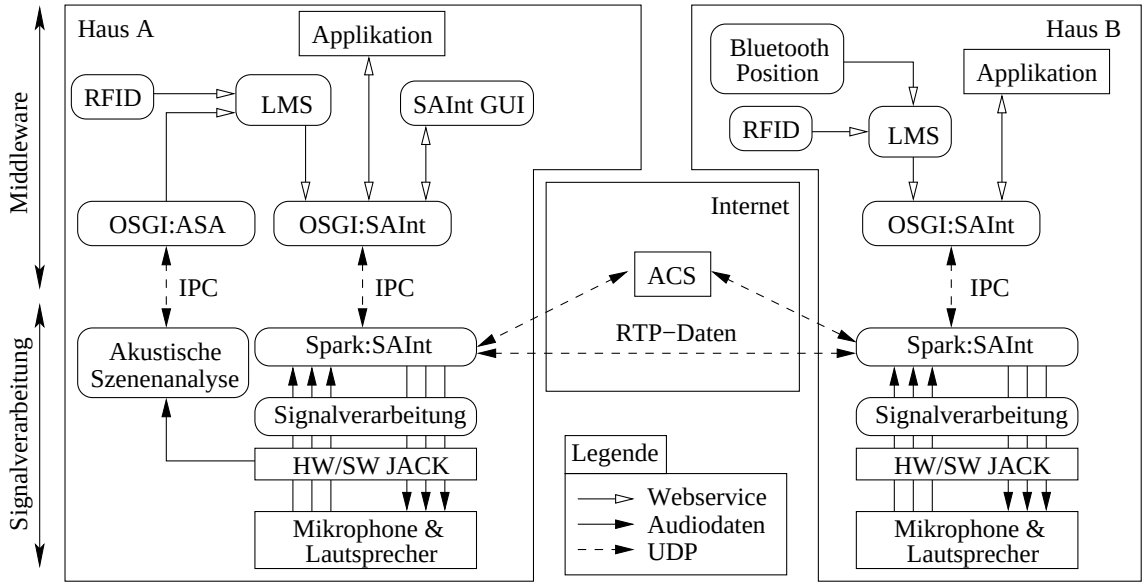


Abbildung 7.2: Blockschaltbild zur Integration von *SAInt* in die Amigo *Middleware*

In Abb. 7.2 sind die Abhängigkeiten der verschiedenen Komponenten und die Datenströme für eine Kommunikation zwischen zwei Häusern dargestellt. Wie bereits in Abb. 7.1 detaillierter gezeigt wurde, verbindet *JACK* die Hardware mit der Signalverarbeitung. An dieser Stelle werden auch die Audiodaten für die akustische Szenenanalyse entnommen, deren Kontextinformationen über die Kontextquelle *OSGI:ASA* der *Middleware* zur Verfügung gestellt werden. Der untere Teil der Abb. 7.2 zeigt die Signalverarbeitung, die durch *IPC*-Schnittstellen mit den Diensten der *Middleware* verbunden ist. Der Datenaustausch innerhalb der *Middleware* wird durch *Webservice*-Aufrufe realisiert und basiert im Falle von Kontextquellen auf der *IContextSource*-Schnittstelle (vgl. Kap. 6.4.1).

Im Haus A sind als Lokalisierungstechniken die akustische Szenenanalyse und ein *RFID*-System vorhanden. Die Daten der beiden Kontextquellen werden im *LMS* zu einer neuen Kontextquelle zusammengefasst. Diese wird durch den *SAInt*-Dienst („*OSGI*:*SAInt*“) zur Lokalisierung von Benutzern verwendet. Gleichzeitig interagieren eine Applikation und die graphische Schnittstelle von *SAInt* („*SAInt GUI*“) mit dem *SAInt*-Dienst. Im Haus B befindet sich neben einem *RFID*-System auch eine Positionsbestimmung auf Basis von Bluetooth®-Signalen.

Die Kommunikation zwischen den Häusern verwendet die entstörten Signale aus der Signalverarbeitung. Diese werden durch das *SAInt*-Modul entweder direkt im Haus oder über eine *RTP*-Verbindung weitergeleitet. Hierbei wird ein Kommunikationsdienst (engl. *Ambient Communication Service, ACS*) auf einem entfernten Server verwendet, welcher für den Sitzungsaufbau und die Behandlung der Übersetzung von Netzwerkadressen (engl. *Network Address Translation, NAT*) zuständig ist.

7.2 Signalverarbeitung

Die Aufgabe der Signalverarbeitung ist eine adaptive Filterung der Mikrophonsignale vor der Übertragung durch das Kommunikationssystem. Hierbei wird sowohl eine Echounterdrückung als auch eine Störgeräuschreduktion durchgeführt. Eine Echounterdrückung ist nötig, da die empfangenen Signale des entfernten Sprechers über die Lautsprecher wiedergegeben werden und über die Mikrophone im selben Raum aufgenommen werden. Falls keine Filterung der Signale durchgeführt wird, so kann der entfernte Sprecher sein eigenes Echo hören. Wird auf beiden Seiten eine Freisprecheinrichtung verwendet, so kann es zu einer Rückkopplung der Signale und einem Aufschwingen des Systems kommen. Die Echounterdrückung ist somit nicht nur für den subjektiven Höreindruck der Nutzer wichtig, sondern auch für die Stabilität des Übertragungssystems notwendig. Die Nachfilterung der Mikrophonsignale hinsichtlich möglicher stationärer Störungen ist optional, da es im Rahmen der ambienten Kommunikation durchaus erwünscht sein könnte, dass Hintergrundgeräusche zur Einordnung der aktuellen Aktivitäten mit übertragen werden.

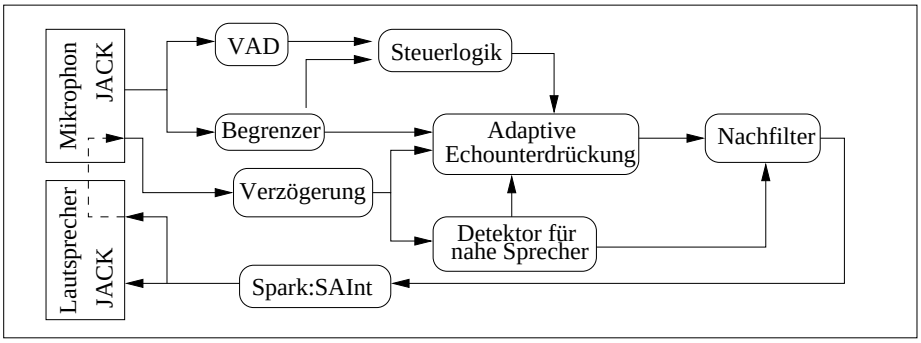


Abbildung 7.3: Blockschaltbild zur Echounterdrückung und Störgeräuschfilterung des *SAInt*

Die Abb. 7.3 zeigt das Blockschaltbild der Signalverarbeitung zur Echounterdrückung und Störgeräuschfilterung, wie es im Amigo System zur ambienten Kommunikation verwendet wird. Die Signalverarbeitung in *Spark* arbeitet nach dem Prinzip eines diskreten Ereignissystems und ist modular aufgebaut. Jedes Modul wird einmal ausgeführt, sobald an jedem Eingang des Moduls ein Datenpaket anliegt. Somit sind rekursive Strukturen, bei denen Eingänge von Modulen von deren Ausgängen abhängig sind, nicht mit *Spark* realisierbar. Die in Abb. 7.3 eingezeichnete Rückkopplung der wiedergegebenen Signale des entfernten Sprechers, welche in der Echounterdrückung benötigt wird, erfolgt über *JACK* [JAC08]. Hierzu wird ein virtueller Lautsprecher in *JACK* erzeugt und intern mit einem virtuellen Mikrophon verknüpft (gestrichelte Linie). Sollte es bei *JACK* durch eine zu hohe Rechenlast zu Paketverlusten kommen, so verliert die wiedergegebene Tonspur im virtuellen Mikrophon die gleiche Anzahl an Paketen wie die Tonspuren der aufgenommen Mikrophonsignale. Es besteht somit nicht die Gefahr, dass die beiden Tonspuren zeitlich auseinanderlaufen. Im Folgenden werden die signalverarbeitenden Module und ihre zugrunde liegenden Algorithmen erläutert.

7.2.1 Begrenzer

Der Begrenzer ist eine notwendige Komponente, um die Stabilität des Systems im Falle von lauten Störungen zu gewährleisten. Bei akustischen Ereignissen mit hohen Energien, wie z. B. einer laut rufenden Person in der Nähe eines Mikrophons oder einer zuschlagenden Tür, kann die begrenzte Dämpfung der Echounterdrückung auf der entfernten Seite kurzzeitig nicht ausreichen und es kommt zu einer aufschwingenden akustischen Rückkopplung in Form eines Pfeifens. Der Begrenzer nach [Zöl97] dämpft die Eingangssignale, deren Energie oberhalb einer festgelegten Schwelle liegt, auf den Schwellwert und beeinflusst Signale unterhalb der Schwelle nicht.

Zunächst wird der geglättete Spitzenwert $x_p(n)$ der Energie $|\boldsymbol{x}(n)|$ eines Blocks über den zeitlichen Verlauf der Signalblöcke $\boldsymbol{x}(n)$ mit

$$x_p(n) \;=\; \begin{cases} (1-\tau_A-\tau_R)x_p(n-1)+\tau_A|\boldsymbol{x}(n)| & \text{für } |\boldsymbol{x}(n)| > x_p(n-1) \\ (1-\tau_R)x_p(n-1) & \text{für } |\boldsymbol{x}(n)| \leq x_p(n-1) \end{cases} \tag{7.1}$$

bestimmt. Die Parameter τ_A für die Anstiegszeit und τ_R für die Abfallzeit beeinflussen die Stärke der Glättung und sind in informellen Experimenten im Akustiklabor zu $\tau_A = 0{,}9$ und $\tau_R = 0{,}005$ bestimmt worden. Anschließend wird der Gewichtsfaktor $\Gamma(n)$ entsprechend des Schwellwertes γ_T durch

$$\Gamma(n) \;=\; \begin{cases} \beta \cdot \Gamma(n-1) & \text{für } \log\{x_p(n)\} > \gamma_T \\ \beta \cdot \Gamma(n-1) + (1-\beta) & \text{für } \log\{x_p(n)\} \leq \gamma_T \end{cases} \tag{7.2}$$

berechnet. Die Glättungskonstante wurde experimentell zu $\beta = 0{,}9$ bestimmt. Das Ausgangssignal des Begrenzers ergibt sich aus der Multiplikation des Eingangssignalblocks $\boldsymbol{x}(n)$ mit der Dämpfung $\Gamma(n)$.

Für den Fall, dass die Bedingung $\log\{x_p(n)\} > \gamma_T$ erfüllt ist, wird der logische Ausgang des Moduls für mehrere Blöcke auf „Wahr“ gesetzt. Dies signalisiert dem nachfolgenden adaptiven Filter die künstliche Begrenzung der Eingangssignale und verhindert so eine mögliche fehlerhafte Adaption.

7.2.2 Sprachaktivitätsdetektion

Die Sprachaktivitätsdetektion (engl. *Voice Activity Detection*, *VAD*) ist eine der entscheidenden Komponenten im System, da basierend auf der Sprachaktivitätsdetektion Entscheidungen in der Strahlformung, der Echounterdrückung, der Positionsschätzung und der Sprecheridentifikation vorgenommen werden. Jedes dieser Teilaufgabengebiete hat spezielle Anforderungen an eine Sprachaktivitätsdetektion, die eine *VAD* alleine nicht erfüllen kann. Eine *VAD* kann entweder Sprache von Hintergrundgeräuschen sicher unterscheiden, was dazu führt, dass Teile der Sprache mit wenig Energie als Geräusche klassifiziert werden, oder eine *VAD* kann so eingestellt werden, dass auch Sprachanteile mit geringer Energie gefunden werden, was dazu führt, dass Störgeräusche häufiger als Sprache klassifiziert werden.

Die akustische Strahlformung soll die Richtcharakteristik der Mikrophongruppe auf einen Benutzer immer dann anpassen, sobald dieser spricht. Störgeräusche, wie z. B. Türen oder Lüfter, sollen hingegen ignoriert werden. Ein effizienter Ansatz hierzu wurde in [RS04] vorgestellt. Hierbei werden im Zeitbereich Mittelwerte der Energie berechnet und miteinander

verglichen. Übersteigt der über ein kurzes Fenster gemittelte Wert der Energie den langfristig gemittelten Wert für die Hintergrundstörung, so wird eine Entscheidung für Sprachaktivität getroffen. Dieser Ansatz liefert in Umgebungen mit geringen Störungen sowohl für die Steuerung der akustischen Strahlformung als auch für Entscheidungen für die Adaption der Filter in der Echounterdrückung gute Ergebnisse. Die Leistungsfähigkeit sinkt jedoch mit ansteigendem Pegel der Störungen, so dass in stark gestörten Umgebungen aufwändigere Ansätze, wie z. B. in [WSH07] vorgeschlagen, verwendet werden müssen.

Die Sprecherprotokollierung besitzt andere Anforderungen an die Sprachaktivitätsdetektion als die akustische Strahlformung. Entsprechend der in der Spracherkennung verwendeten Verfahren, soll eine *VAD* zur Sprecherprotokollierung möglichst zusammenhängende Segmente von Sprache erkennen und diese auch zusammenhängend kennzeichnen. Selbst Sprachanteile mit geringer Energie sollen als Sprache gekennzeichnet werden. Somit wird es nötig, einen Sicherheitsbereich um einen Bereich erkannter Sprache zu definieren, welcher auch der Sprache zugeordnet wird. Dies führt zwangsläufig zu einer Vergrößerung der Latenz der Sprachaktivitätsentscheidung in der Größenordnung des Sicherheitsbereichs vor der erkannten Sprache. Da zur Merkmalsextraktion und zur Entstörung bereits das *Advanced Frontend ETSI* nach [ETS02] verwendet wird, kann auch die dort beschriebene Erweiterung zur Sprachaktivitätsdetektion verwendet werden. Diese ist zur Verwendung mit einem Spracherkenner optimiert und erfüllt die zuvor beschriebenen Anforderungen.

7.2.3 Echounterdrückung

Die Module der Echounterdrückung benötigen zur Neuschätzung der adaptiven Filter Informationen über die Sprachaktivität der Kommunikationsteilnehmer. Hierbei kann die Entscheidung bezüglich des entfernten Sprechers durch die Verwendung einer *VAD* auf den empfangenen Signalen getroffen werden. Ein lokaler Sprecher kann ebenfalls durch eine *VAD* detektiert werden, falls der entfernte Sprecher nicht aktiv ist. Da jedoch die Möglichkeit besteht, dass auf beiden Seiten die Sprecher aktiv sind, muss eine Detektion des nahen Sprechers durchgeführt werden. Dies erfolgt im Detektor für nahe Sprecher (engl. *Near Speaker Detector, NSD*), welcher seine Entscheidung auf Grund der Mikrophonsignale, der wiedergegeben Signale und der geschätzten Raumimpulsantwort trifft.

Die Echounterdrückung schätzt durch die Adaption des *AEC*-Filters die unbekannte Übertragungsfunktion zwischen Mikrophon und Lautsprecher. Da diese Übertragungsfunktion nicht nur durch die Anordnung der Mikrophone und Lautsprecher, sondern maßgeblich durch den Raum bestimmt ist, wird die Fourier-Rücktransformierte dieser Übertragungsfunktion abkürzend als Raumimpulsantwort bezeichnet. Im *AEC* wird ein Filter mit endlicher Filterimpulsantwort (engl. *Finite Impulse Response*, *FIR*) zur Schätzung der Raumimpulsantwort verwendet, so dass im Allgemeinen Restechos im Ausgangssignal des *AEC* verbleiben. Diese werden durch ein Nachfilter soweit reduziert, dass sie durch den entfernten Sprecher nicht mehr wahrgenommen werden können.

Dieser zuvor beschriebene Ansatz zur Echounterdrückung hat den Nachteil, dass für eine verlässliche Entscheidung der *NSD* zunächst eine gute Schätzung der Raumimpulsantwort vorliegen muss. Die Raumimpulsantwort kann aber nur korrekt geschätzt werden, falls während der Adaption kein lokaler Sprecher aktiv ist. Somit bedingt die Schätzung der *NSD* auch die Adaption des *AEC* und umgekehrt. Geht man davon aus, dass das System zur ambienten Kommunikation fest im Haus installiert ist, kann eine Vorschätzung der Raumimpulsantwort

während der Installation vorgenommen werden. Diese wird als Startwert für die adaptiven Filter des *AEC* verwendet und der *NSD* kann von Beginn an gute Schätzungen für das Vorhandensein eines lokalen Sprechers vornehmen.

Detektion eines nahen Sprechers

Die Detektion eines nahen Sprechers erfolgt entsprechend [BMC00] durch die Kreuzkorrelation zwischen dem wiedergegeben Signal und dem aufgenommenen Signal. Dabei sei die Raumimpulsantwort mit $\boldsymbol{h} = [h_1, \ldots, h_N]^T$ gegeben. Dies führt auf die *NSD*-Entscheidungsvariable

$$\xi \;=\; \frac{\sqrt{\boldsymbol{h}^T \boldsymbol{\phi}_{xx} \boldsymbol{h}}}{\sqrt{\boldsymbol{h}^T \boldsymbol{\phi}_{xx} \boldsymbol{h} + \sigma_s^2}} \tag{7.3}$$

mit σ_s^2 als der Varianz des lokalen Sprechersignals und $\boldsymbol{\phi}_{xx}$ der Matrix der Autokorrelationsterme des wiedergegebenen Signals. Ist der lokale Sprecher inaktiv, so gilt $\xi = 1$, und für einen aktiven lokalen Sprecher ist $\xi < 1$.

Da die Filterung eines Signals effizienter im Frequenzbereich als im Zeitbereich durchgeführt werden kann, wird für die ambiente Kommunikation die in [GB01] vorgestellte Berechnung der Entscheidungsvariablen ξ im Frequenzbereich genutzt. Hierbei werden blockweise die Auto- und Kreuzkorrelation der Signale im Frequenzbereich geschätzt und anschließend zeitlich geglättet. Der Zähler der Entscheidungsvariablen in Gl. 7.3 wird durch eine Multiplikation der geschätzten Raumimpulsantwort mit der Kreuzkorrelation zwischen wiedergegebenem und aufgenommenem Signal näherungsweise bestimmt. Der Nenner ist durch die Autokorrelation des Mikrophonsignals gegeben.

Adaptive Filterung

Die Echounterdrückung ist eine Systemidentifikationsaufgabe, bei der das unbekannte Übertragungssystem zwischen Mikrophon und Lautsprecher durch ein adaptives Filter geschätzt werden soll [Hay02]. Dabei wird ein *FIR*-Filter zur Nachbildung der unbekannten Raumimpulsantwort blockweise durch Anwendung eines „*Normalized Least Mean Square*“-Algorithmus (*NLMS*-Algorithmus) adaptiert [BH03]. In Abb. 7.4 ist der prinzipielle Aufbau der

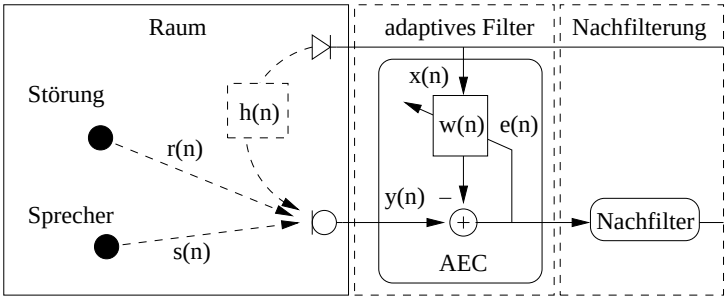


Abbildung 7.4: Blockschaltbild der adaptiven Filterung zur Echounterdrückung

Echounterdrückung dargestellt. Das aufgenommene Mikrophonsignal $y(n)$ setzt sich aus der lokalen Störung $r(n)$, dem lokalen Sprecher $s(n)$ und dem mit der Raumimpulsantwort $h(n)$ gefalteten Signal des entfernten Sprechers $x(n)$ zusammen.

Die Adaptionsgleichung des Filters w ist mit

$$w(n+1) \quad = \quad w(n) + \mu(n) \cdot \frac{x(n) \cdot e(n)}{|x(n)|^2} \tag{7.4}$$

gegeben, mit $\mu(n)$ als Schrittweite und $e(n)$ als Fehlersignal.

Die Vorteile des *NLMS*-Algorithmus liegen in der niedrigen Komplexität des Algorithmus (Filterlänge N, $O_{NLMS} \sim 2N$, [Hay02]) und seiner Robustheit gegenüber Störungen und falschen Entscheidungen zur Adaption. Nachteilig ist die langsame Konvergenz bei zeitlichen Änderungen des zu identifizierenden Systems, wobei dies in der Anwendung der ambienten Kommunikation eine geringere Rolle spielt. Aufgrund des festen Aufbaus stellt die Anordnung der Mikrophone und Lautsprecher ein zeitlich näherungsweise konstantes System dar, das nur geringe Anpassungen der geschätzten Filter bedarf. Folglich kann eine kleine Schrittweite $\mu(n)$ gewählt werden, wodurch der Einfluss fehlerhafter Entscheidungen durch den *NSD* minimiert wird.

Die Implementierung des *AEC* erfolgt, wie zuvor beim *NSD*, im Frequenzbereich mit Hilfe eines *Overlap-Save*-Verfahrens. Zusätzlich wird die Filterung partitioniert durchgeführt, um eine unabhängig von der verwendeten Filterlänge konstant niedrige Latenz des *AEC*-Moduls zu erzielen (vgl. [DES99]).

Das Ausgangssignal des *AEC* enthält neben lokalen Störungen $r(n)$ auch Restechos $b(n)$, weil das endliche Filter des *AEC* auf Grund seiner Länge nur einen Teil der Raumimpulsantwort nachbilden kann. Jedoch werden in einem nachgeschalteten Filter diese Restechos zusammen mit den lokalen Störgeräuschen soweit reduziert, dass sie für den Benutzer nicht mehr wahrnehmbar sind.

Nachfilter

Die Nachfilterung des *AEC*-Ausgangssignals wurde entsprechend dem Vorschlag in [LK07] implementiert. Das Ausgangssignal des *AEC* ergibt sich zu

$$e(n) \quad = \quad h(n) * x(n) + s(n) + r(n) - w(n) * x(n) \tag{7.5}$$

$$\quad = \quad \underbrace{(h(n) - w(n)) * x(n)}_{b(n)} + s(n) + r(n) \tag{7.6}$$

mit $b(n)$ als dem verbleibenden Restecho des entfernten Sprechers. Unter der Annahme, dass das Signal des lokalen Sprechers, die lokale Störung und das Restecho statistisch unabhängig sind gilt

$$E(m,\omega) \quad = \quad B(m,\omega) + S(m,\omega) + R(m,\omega) \tag{7.7}$$

mit $E(m,\omega), S(m,\omega), R(m,\omega)$ und $B(m,\omega)$ als den Frequenzspektren der Signale $e(n)$, $s(n)$, $r(n)$ und $b(n)$ im betrachteten Signalblock m. Grundidee in [LK07] ist die Einführung von vier Hypothesen über die Signalanteile im momentanen Mikrophonsignal:

- H_0: Störgeräusche $E(m,\omega) = R(m,\omega)$.
- H_1: Störgeräusche und lokaler Sprecher $E(m,\omega) = R(m,\omega) + S(m,\omega)$.
- H_2: Störgeräusche und entfernter Sprecher $E(m,\omega) = R(m,\omega) + B(m,\omega)$.

- H_3: Störgeräusche, entfernter Sprecher und lokaler Sprecher
  $E(m,\omega) = R(m,\omega) + B(m,\omega) + S(m,\omega)$.

Die Unterscheidung zwischen den beiden Hypothesengruppen H_0, H_1 und H_2, H_3 kann zuverlässig durch eine Sprachaktivitätsdetektion auf dem Signal des entfernten Sprechers durchgeführt werden. Der Test zwischen den Hypothesen innerhalb der Gruppen entspricht dem Problem der zuvor vorgestellten Detektion eines nahen Sprechers.

Die Übertragungsfunktion des Nachfilters ergibt sich nach [LK07] zu

$$F(m,\omega) = \frac{\xi(m,\omega)\cdot\zeta(m,\omega)}{\xi(m,\omega)\cdot\zeta(m,\omega)+\xi(m,\omega)+\zeta(m,\omega)} \tag{7.8}$$

mit dem a priori *SNR*

$$\xi(m,\omega) = \alpha_\xi\sigma\left(\frac{|E(m,\omega)|^2}{\widehat{R}_n(m,\omega)}-1\right)+(1-\alpha_\xi)\frac{|F(m-1,\omega)E(m-1,\omega)|^2}{\widehat{R}_n(m,\omega)} \tag{7.9}$$

und dem a priori Signal-zu-Echoverhältnis (engl. *Signal to Echo Ratio, SER*)

$$\zeta(m,\omega) = \alpha_\zeta\sigma\left(\frac{|E(m,\omega)|^2}{\widehat{R}_b(m,\omega)}-1\right)+(1-\alpha_\zeta)\frac{|F(m-1,\omega)E(m-1,\omega)|^2}{\widehat{R}_b(m,\omega)}. \tag{7.10}$$

Dabei sei $\widehat{R}_n(m,\omega)$ die Schätzung des Leistungsdichtespektrums des lokalen Rauschens, $\widehat{R}_b(m,\omega)$ die Schätzung des Leistungsdichtespektrums des Restechos und $\sigma()$ die Einheitssprungfunktion. Die Parameter werden zu $\alpha_\xi = 0{,}99$ und $\alpha_\zeta = 0{,}95$ gewählt.

Da die ambiente Kommunikation auch die Übertragung von Geräuschen aus der Umgebung der Kommunikationspartner optional mit einschließen soll, ist eine Modifikation des Filters aus Gl. 7.8 notwendig. Entsprechend der Idee aus [GJKV99] ergibt sich die neue Filterfunktion zu

$$\widetilde{F}(m,\omega) = \frac{\xi(m,\omega)\cdot\zeta(m,\omega)+\beta_\xi\,\xi(m,\omega)+\beta_\zeta\,\zeta(m,\omega)}{\xi(m,\omega)\cdot\zeta(m,\omega)+\xi(m,\omega)+\zeta(m,\omega)} \tag{7.11}$$

mit dem Parameter β_ξ zur Steuerung der Unterdrückung lokaler Störungen und β_ζ zur Beeinflussung der Restechounterdrückung. Dieser Ansatz bietet zudem den Vorteil, dass Störungen, wie z. B. *Musical Tones*, durch eine gute Wahl der Parameter vermieden werden können, indem eine Reststörung in den Signalen toleriert wird.

7.3 Echtzeitkommunikation

Das *SAInt*-Modul unterscheidet bei der Echtzeitkommunikation zwei Arten von Verbindungen. Zum einen sind dies lokale Verbindungen zwischen Personen im Haus und zum anderen externe Verbindungen zwischen lokalen und entfernten Personen. Im ersten Fall müssen die Personen im Haus lokalisiert und anschließend eine Audioverbindung über die entsprechenden Mikrophone und Lautsprecher aufgebaut werden. Der zweite Fall erfordert eine Positionsbestimmung des lokalen Teilnehmers und den Aufbau eines echtzeitfähigen Datenstroms über ein *IP*-basiertes Netzwerk. Eine Lokalisation von Nutzern erfolgt lokal immer über den *SAInt*-Dienst, der als Kontextnutzer in der Amigo *Middleware* agiert.

7.3.1 Lokalisation von Nutzern

Die Positionsdaten von Benutzern werden im Amigo System durch verschiedene Kontextquellen bereitgestellt. Dabei unterscheiden sich die Daten bezüglich der räumlichen und der zeitlichen Auflösung. Um eine kontinuierliche Suche nach Kontextquellen und der anschließenden Registrierung bei allen geeigneten Kontextquellen zu vermeiden, verwendet der *SAInt*-Dienst den Amigo *Location Management Service* (*LMS*). Der *LMS* übernimmt die Suche nach Kontextquellen und führt die unterschiedlichen Informationen in einer gemeinsamen Datenbank zusammen. Diese Datenbank wird als Kontextquelle anderen Diensten über die *IContextSource*-Schnittstelle zur Verfügung gestellt und kann über die Anfrage in Liste 7.1 über den Kontextbroker gesucht werden.

```
<?xml version=\"1.0\"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:j.1="http://amigo.gforge.inria.fr/owl/ContextTransport.owl#">
  <j.1:ContextSourceRegistration>
    <j.1:contextType>
        CombinedUserLocation
    </j.1:contextType>
    <j.1:timeliness>
        current
    </j.1:timeliness>
  </j.1:ContextSourceRegistration>
</rdf:RDF>
```

Liste 7.1: Anfrage des *SAInt*-Dienstes an den Kontextbroker zur Suche des *LMS*

Der *SAInt*-Dienst auf der *OSGI*-Plattform, welcher über eine *IPC*-Schnittstelle mit dem *SAInt*-Modul verbunden ist, sucht über den Kontextbroker nach laufenden *LMS*-Diensten und registriert sich dort. Während der Registrierung hinterlegt der *SAInt*-Dienst beim *LMS* die *SPARQL*-Frage in Liste 7.2, so dass im Falle einer Positionsänderung diese dem *SAInt*-Dienst unverzüglich mitgeteilt wird. Fortlaufend werden die Positionsinformationen über Nutzer von dem *SAInt*-Dienst an das *SAInt*-Modul weitergeleitet, so dass das *SAInt*-Modul eine automatische Sitzungsverwaltung durchführen kann.

Die *SPARQL*-Frage ist an der Position von Personen mit der Genauigkeit auf Raumebene interessiert und besitzt einen optionalen Teil, um präzisere Informationen abzufragen. Notwendig für die Funktion des Dienstes ist die Information über den Raum, in dem sich der Benutzer befindet. Die optionale Information, an welcher relativen Position im Raum die Person aktuell ist, ermöglicht im Falle verteilter Mikrophone und Lautsprecher die Auswahl der nächstgelegenen Hardware.

7.3.2 Sitzungsverwaltung

Die Sitzungsverwaltung dient dem Aufbau von externen Verbindungen und automatisiert den hierfür notwendigen Registrierungsprozess. Sobald eine Person vom System in einen Raum mit ausreichender Hardwareausstattung (Mikrophon und Lautsprecher) lokalisiert wird, führt das *SAInt*-Modul eine Registrierung dieser Person beim Kommunikationsdienst (*ACS*) durch. Andere *SAInt*-Module, welche mit dem gleichen Kommunikationsdienst verbunden sind, erhalten hierdurch die Nachricht, dass diese Person für eine Kommunikation zur Verfügung

```
PREFIX context:<http://amigo.gforge.inria.fr/owl/ContextTransport.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?user ?room ?time ?prob ?x ?y WHERE {
        ?ul rdf:type context:UserLocation .
        ?ul context:timestamp ?time .
        ?ul context:probability ?prob .
        ?ul context:isLocatedIn ?r .
        ?r context:identifier ?room .
        ?ul context:isLocationOf ?u .
          ?u context:identifier ?user .
        optional {?ul context:estimatedPosition ?ep .
        ?ep context:X ?x .
        ?ep context:Y ?y .
        ?ul context:relative2Space ?r.
        ?r context:identifier ?room.}
        };
```

Liste 7.2: *SPARQL*-Frage des *SAInt*-Dienstes an den *LMS*

steht. Verlässt diese Person den Raum und geht in einen Bereich ohne Hardware, so wird die Registrierung beim Kommunikationsdienst durch das *SAInt*-Modul zurückgezogen.

Die Echtzeitkommunikation besitzt eine benutzerorientierte Architektur, so dass Verbindungen an Personen und nicht an Geräte oder Räume gebunden sind. Eine Verbindung wird zwischen zwei Personen initiiert, indem entweder eine der Personen eine direkte Verbindungsanfrage zu einer anderen Person stellt oder indem eine Applikation versucht, zwei Personen zu verbinden. In beiden Fällen wird die *Webservice*-Methode „*Connect(Person A, Person B)*“ des *SAInt*-Dienstes verwendet, um eine Verbindung zu initialisieren.

Jede Verbindungsanfrage wird über die *IPC*-Schnittstelle an das *SAInt*-Modul weitergeleitet, welches die Position der Teilnehmer vom *SAInt*-Dienst abfragt. Falls beide Personen sich im Haus befinden und eine Möglichkeit zur Kommunikation durch Mikrophone und Lautsprecher besteht, so wird eine direkte Verbindung zwischen den Räumen hergestellt. Konnte nur ein Teilnehmer im Haus lokalisiert werden, so wird versucht, mittels eines Sitzungsprotokolls eine externe Verbindung zur anderen Person über den Kommunikationsdienst herzustellen. Hierzu sendet das lokale *SAInt*-Modul eine Verbindungseinladung über den Kommunikationsdienst an das *SAInt*-Modul des entfernten Teilnehmers. Akzeptiert dieser die Einladung zur Kommunikation, so teilt anschließend der Kommunikationsdienst den beiden *SAInt*-Modulen die *IP*-Adressen der Teilnehmer mit, so dass diese eine direkte Verbindung untereinander aufbauen können.

7.3.3 Datenaustausch

Der Datenaustausch zwischen entfernten Kommunikationsteilnehmern erfolgt verbindungslos über *UDP*-Verbindungen. Vorteil dieses Ansatzes ist die niedrige Latenz bei der Übertragung der Audiodaten, der jedoch durch mögliche Paketverluste oder die Vertauschung von Datenpaketen beim Empfang durch unterschiedliche Paketlaufzeiten (engl. *jitter*) erkauft wird. Die Audiodaten werden zunächst komprimiert, um die Datenrate zu reduzieren, und anschließend mit dem *Real-Time Transport Protocol* (*RTP*) [S$^+$03] in die *UDP*-Pakete verpackt.

Das im *RFC3489* durch die *Internet Engineering Task Force* vorgestellte „*Simple Traver-*

sal of User Datagram Protocol Through Network Address Translators“-Protokoll (*STUN*-Protokoll) [R+03] beschreibt die Detektion und Überwindung von Verfahren zur Übersetzung von Netzwerkadressen (*NAT*). Die Abb. 7.5 zeigt an einem Beispiel die Problemstellung beim Datenaustausch, hervorgerufen durch die Umsetzung von internen Adressen auf externe Adressen, und die Lösung des Problems durch die Verwendung des *ACS*.

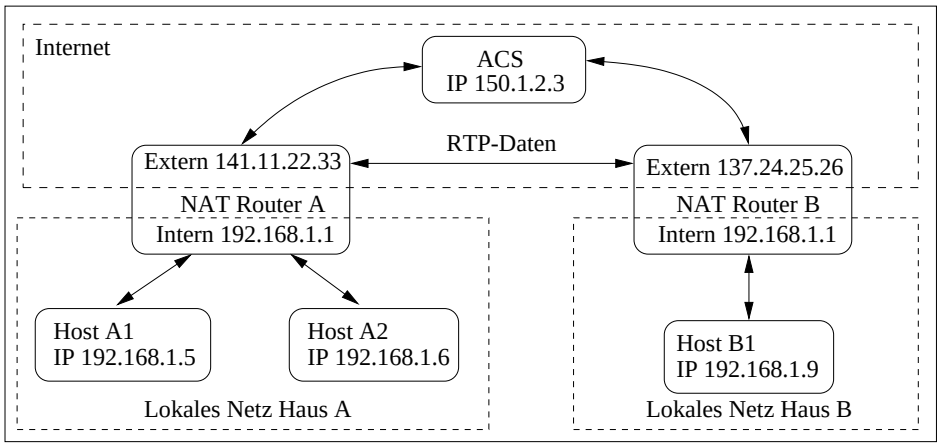


Abbildung 7.5: Beispiel für die *NAT*-Problematik der ambienten Kommunikation

Angenommen es soll eine Datenverbindung zwischen dem *Host* A1 und dem *Host* B1 aufgebaut werden. Beide *Hosts* kennen zwar ihre lokale Adresse, jedoch nicht die externe Adresse ihres Routers. Als gemeinsamer Anlaufpunkt zum Aufbau einer Kommunikationssitzung wird der *ACS* verwendet, der von beiden erreichbar ist. Sendet einer der *Hosts* ein Paket an den *ACS*, so ersetzt der jeweilige Router im Rahmen der *NAT* die Adresse im Paket durch seine eigene externe Adresse, bevor das Paket an den *ACS* weitergeleitet wird. Antwortet der *ACS* auf dieses Paket, so leitet der Router das Antwortpaket weiter an den entsprechenden *Host*, welcher zuvor eine Anfrage an den *ACS* gesendet hat.

Der *Host* A1 kann kein Paket direkt an den *Host* B1 senden, da er die externe Adresse des Routers B nicht kennt. Da beide *Hosts* auf dem *ACS* registriert sind, kennt dieser die externen Adressen der Router aus den empfangenen Paketen und kann diese bei einer Verbindungsanfrage an beide Kommunikationsteilnehmer übermitteln. Sobald die *Hosts* die externe Adresse des jeweiligen anderen Teilnehmers kennen, beginnen sie Pakete an diese Adresse zu senden. Empfängt der Router B nun ein Paket von Router A, so nimmt er an, dass es die Antwort auf das von *Host* B1 an *Host* A1 gesendete Paket ist und leitet es an den *Host* B1 weiter. Das gleiche führt entsprechend der Router A mit den von ihm empfangenen Paketen durch. Mit diesem Verfahren ist es möglich, die *NAT*-Verfahren *Full Cone*, *Restricted Cone* und *Port Restricted Cone* zu überwinden, falls für die Kommunikation mit dem *ACS* derselbe Port genutzt wird wie für den Datenaustausch zwischen den *Hosts*. Der *ACS* übernimmt somit neben der im *STUN*-Protokoll beschriebenen Überwindung der *NAT* auch die Sitzungsinitialisierung vergleichbar zu *SIP* [R+02].

Die Audiosignale selbst werden mit dem *Speex*-Codec (16 kHz Breitband) [PS08] komprimiert, um die benötigte Bandbreite zu reduzieren. Dabei übernimmt der *Speex*-Codec im Rahmen der Paketverlustverschleierung die Kompensation verlorengegangener Pakete. Mögliche Paketverluste durch Schwankungen in der Paketlaufzeit (sog. *Jitter*) werden im *SAInt*-Modul durch einen Paketpuffer reduziert.

7.4 Kontextbasierte Steuerung

Der Kern des Systems zur ambienten Kommunikation ist die kontextbasierte Steuerung, welche die Amigo *Middleware* verwendet, um relevante Kontextinformationen zu sammeln und auszuwerten. Hierbei sind Kontextquellen mit Positionsinformationen über Personen bzw. zentralisierte Dienste wie der *LMS* notwendig, um automatisierte Entscheidungen treffen zu können. Der *SAInt*-Dienst führt beim Start zunächst eine synchrone Abfrage aller Kontextquellen auf Informationen durch und registriert sich anschließend bei diesen für asynchrone Benachrichtigungen (vgl. Kap. 6.4.1). Ein erster Teil der kontextbasierten Steuerung ist die bereits vorgestellte automatische Sitzungsverwaltung aus Kap. 7.3.2. Diese führt, ausgehend von den Kontextinformationen über die Position der Nutzer, eine automatische Registrierung der Nutzer bei einem Kommunikationsdienst durch.

7.4.1 *Follow-Me*-Fähigkeiten

Die Idee in *Follow-Me*-Szenarien ist es, eine Kommunikation einem Sprecher folgen zu lassen, ohne dass dieser direkten Einfluss auf eine Anwendung nehmen oder Anweisungen geben muss. Hierzu muss das System den aktuellen Ort des Kommunikationsteilnehmers kennen und im Falle einer Positionsänderung eine Anpassung vornehmen. Da eine kontinuierliche, zyklische Abfrage von Positionsdaten zu einer hohen Belastung der *Middleware* führt, wird der Mechanismus des asynchronen Datenaustausches verwendet, um auf Änderungen des Kontextes zu reagieren.

Bewegt sich eine Person von einem Raum in einen anderen, so sollte dies durch eine der Kontextquellen registriert und an den *LMS* weitergemeldet werden. Da der *SAInt*-Dienst beim *LMS* eine *SPARQL*-Frage nach der Position aller Personen bei der Registrierung hinterlegt hat, wird die Änderung der Position zu einer Kontextinformation als Antwort auf die *SPARQL*-Frage führen. Folglich ruft der *LMS* die *Webservice*-Methode *notify* des registrierten *SAInt*-Dienstes mit der *SPARQL* Antwort als Übergabeparameter auf. Der *SAInt*-Dienst selbst signalisiert dem *SAInt*-Modul über die *IPC*-Schnittstelle, dass neue Kontextinformationen vorliegen, und übermittelt diese. Dies führt zu einer Überprüfung der Auswirkungen der neuen Kontextinformationen auf die laufenden Verbindungen und gegebenenfalls einer Anpassung dieser. Zudem werden die Registrierungen der Personen beim *ACS* entsprechend der neuen Daten vorgenommen.

Die Positionsänderung einer Person kann die folgenden Reaktionen hervorrufen. Tritt die Person in den von *SAInt* kontrollierten Bereich ein, so wird sie beim *ACS* registriert und in den Kontextinformationen als verfügbar für eine Kommunikation aufgeführt. Betritt eine Person einen Raum ohne Mikrophone und Lautsprecher, oder verlässt das Haus, so löscht das *SAInt*-Modul automatisch die Registrierung beim Kommunikationsdienst.

Sollte die Person eine laufende Verbindung während des Raumwechsels haben, so ergeben sich mehrere Möglichkeiten, wie das System reagiert. Wird ein Raum mit Mikrophonen und Lautsprechern betreten, so lenkt das *SAInt*-Modul das Gespräch ohne Unterbrechung des Datenstroms in den Raum um. Dieses übergangslose (engl. *seamless*) Umstellen der Verbindung erfolgt für das menschliche Gehör nicht wahrnehmbar, da es ohne Neuaufbau einer *RTP*-Verbindung auskommt und verzögerungsfrei umschaltet. Falls die Person einen Raum ohne Hardware betritt, so stoppt die Verbindung zum entfernten Sprecher und wird gehalten, bis eine konfigurierbare Zeitspanne erreicht ist oder die Person einen Raum mit Hardware

wieder betritt. Das Verhalten kann für jede Verbindung individuell eingestellt werden.

Besitzt ein Gerät im Raum einen alternativen *SAInt*-Dienst, z. B. ein Notebook mit Headset, und einen Anmeldungsmanager der die Anmeldedaten als Positionsinformationen an den *LMS* weitermeldet, so kann eine Übergabe der Verbindung (engl. *handover*) an den zweiten *SAInt*-Dienst durchgeführt werden. Der Nutzer könnte sich zum Beispiel auf dem Notebook anmelden und der *SAInt*-Dienst stellt daraufhin eine Verbindung her. Dies kann jedoch nicht übergangslos erfolgen, da die alte *RTP*-Verbindung beendet und eine neue aufgebaut werden muss. Daher vernehmen beide Nutzer währenddessen einen kurzen Aussetzer der Verbindung, bis die *RTP*-Verbindung wieder aufgebaut ist.

7.4.2 *SAInt* als Kontextquelle

Die ambiente Kommunikation verwendet nicht nur Kontextinformationen, um eine intelligente Steuerung zu realisieren, sondern sie ist gleichzeitig eine Kontextquelle für andere Applikationen und Dienste. Die Liste 7.3 zeigt die Registrierung des *SAInt*-Dienstes beim Kontextbroker als Kontextquelle.

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns="http://amigo.gforge.inria.fr/owl/ContextTransport.owl#"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://amigo.gforge.inria.fr/owl/ContextTransport.owl#">
 <ContextSourceRegistration>
    <timeliness>
       current
    </timeliness>
    <contextType>
      SeamlessAudioInterface
    </contextType>
 </ContextSourceRegistration>
</rdf:RDF>
```

Liste 7.3: Registrierung des *SAInt*-Dienstes beim Kontextbroker

Die Kontextinformationen eines *SAInt* umfassen die drei Bereiche Hardware, registrierte Benutzer und laufende Verbindungen, wie es in Abb. 7.6 beispielhaft dargestellt ist. Der Bereich Hardware informiert über die Räume, welche durch das *SAInt*-Modul mit einer Audioschnittstelle abgedeckt sind. Diese Information ist zeitlich konstant, da sie abhängig von der Hardware ist und sich somit nicht ohne Neustart des *OSGI-Bundles* ändert. Applikationen können also im vernetzten Haus zunächst nach laufenden *SAInt*-Diensten über den Kontextbroker suchen und sich bei diesen als Kontextnutzer registrieren. Dadurch sind sie in der Lage, die Abdeckung mit Audioschnittstellen im gesamten Netzwerk zu ermitteln.

Die Informationen über registrierte Benutzer und laufende Verbindungen zeigen den aktuellen Status der *SAInt*-Dienste. Hieraus erfahren Applikationen, welche Personen gerade über einen *SAInt*-Dienst erreichbar oder aber gerade durch eine laufende Kommunikation gebunden sind. Benutzer, die aktuell eine Kommunikation führen, werden in der Liste der registrierten Nutzer nicht aufgeführt, da jeder Nutzer nur eine Kommunikation führen kann und somit für neue Verbindungen nicht zur Verfügung steht.

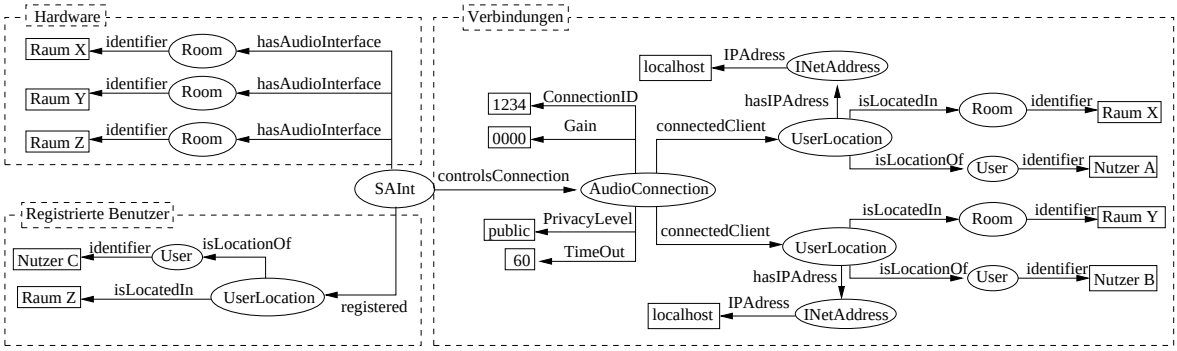


Abbildung 7.6: Beispiel für die Kontextinformationen des *SAInt*-Dienstes

Jede Verbindung zwischen zwei Personen besitzt einen eindeutigen Schlüssel („*ConnectionID*“), eine Charakterisierung der Privatsphäre („*PrivacyLevel*“), einen Schwellwert für das Halten unterbrochener Verbindungen („*Timeout*“ [s]) und einen Verstärkungsfaktor für die Wiedergabe der empfangenen Signale („*Gain*“ [dB]). Diese Parameter können mit Hilfe von *Webservice*-Methoden im Verlauf der Kommunikation durch Applikationen oder die Nutzer angepasst werden, um z. B. auf aktuelle Ereignisse zu reagieren. Die Informationen über die Kommunikationsteilnehmer („*connectedClient*“) zeigen die aktuellen Positionen der Personen oder aber die *IP*-Adressen der entfernten Teilnehmer. Handelt es sich um eine lokale Verbindung, so zeigen die *IP*-Adressen jeweils den Wert „*localhost*“ und die Räume stammen aus der Menge der mit Hardware ausgestatteten Räume. Bei externen Verbindungen ist der Raum des entfernten Teilnehmers auf „*unknown*“ gesetzt, und die *IP*-Adresse ist die Zieladresse des *RTP*-Datenstroms.

7.4.3 Schutz der Privatsphäre

Der Schutz der Privatsphäre ist entsprechend der Ergebnisse der Amigo Benutzerstudien [M$^+$05] bei der kontextabhängigen Steuerung mit berücksichtigt worden. Jede Verbindung besitzt eine Eigenschaft „*PrivacyLevel*“, die entweder öffentlich oder privat gesetzt werden kann. Betritt eine Person einen Raum mit einer laufenden, als privat gekennzeichneten Kommunikation so wird die Verbindung unterbrochen, bis die Person den Raum wieder verlässt. Das gleiche geschieht, falls ein Kommunikationspartner einen Raum betritt, in dem sich bereits eine Person befindet.

Jederzeit kann der „*PrivacyLevel*“ einer Verbindung mittels der *Webservice*-Methode „*configureCommunication(ID, String)*“ des *SAInt*-Dienstes konfiguriert werden. Startet eine Verbindung in einem Raum mit mehr als einer Person, so wird sie standardmäßig als öffentlich vermerkt, ansonsten werden zunächst die Standardwerte der Nutzer verwendet.

Der Sonderfall, dass eine Person mit einer laufenden Kommunikation auf eine weitere Person mit ebenfalls einer laufenden Kommunikation trifft, stellt kein Problem für den Schutz der Privatsphäre dar. Sind beide Kommunikationen öffentlich, so hören zwar die entfernten Teilnehmer jeweils das lokale Gespräch der beiden Personen, jedoch können die entfernten Personen sich gegenseitig nicht hören, da die Echounterdrückung die entsprechenden Signalanteile herausfiltert. Es ist somit entfernten Gesprächsteilnehmern nicht möglich, Rückschlüsse auf die Kommunikationspartner anderer Personen zu ziehen.

7.5 Visuelle Kommunikation

Im Folgenden wird ein System zur Kommunikation vorgestellt, welches aufbauend auf der Architektur von *SAInt* eine audio-visuelle Kommunikation realisiert. Ziel ist es hierbei, die durch *SAInt* ermöglichte Bewegungsfreiheit des Nutzers auch bei einer Übertragung von Videodaten beizubehalten.

7.5.1 Systemintegration

Das *SAInt*-Modul realisiert bereits die *Follow-Me*-Fähigkeiten für die Audiosignale der ambienten Kommunikation mit Hilfe des *SAInt*-Dienstes und der *Middleware*. Folglich liegt es nahe, die visuelle Kommunikation an die akustische Kommunikation zu binden und somit die gleichen Mechanismen zu nutzen. Die visuelle Kommunikation wird als optionale Komponente im System integriert. Sie wird genutzt, falls auf beiden Seiten der Kommunikation geeignete Hardware vorhanden ist.

Ein Unterschied bei der Aufnahme und Wiedergabe von Audio- und Videosignalen ist, dass die Soundkarte eines Computers mehrere Kanäle aufnehmen und wiedergeben, die Grafikkarte jedoch meist nur einen Monitor ansteuern kann. Ein Computer kann somit nur für einen Videodatenstrom genutzt werden. Daher wird zur visuellen Kommunikation das „*Seamless Audio and Video Interface*“-Modul (*SAVInt*-Modul) implementiert, welches die Videodaten einer Kamera aufnimmt und diese über *RTP* versenden kann. Empfangene Daten werden von diesem Modul über einen Ausgang am Bildschirm dargestellt. Zu jeder Kombination von Kamera und Bildschirm gehört folglich ein laufendes *SAVInt*-Modul. Die Videodaten der ambienten Kommunikation können sowohl von einer Netzwerkkamera als auch einer lokal an den Computer angeschlossenen Kamera (z. B. USB-Webcam) stammen. Sie werden mit dem *Theora*-Codec [The08] komprimiert und mittels *RTP* übertragen.

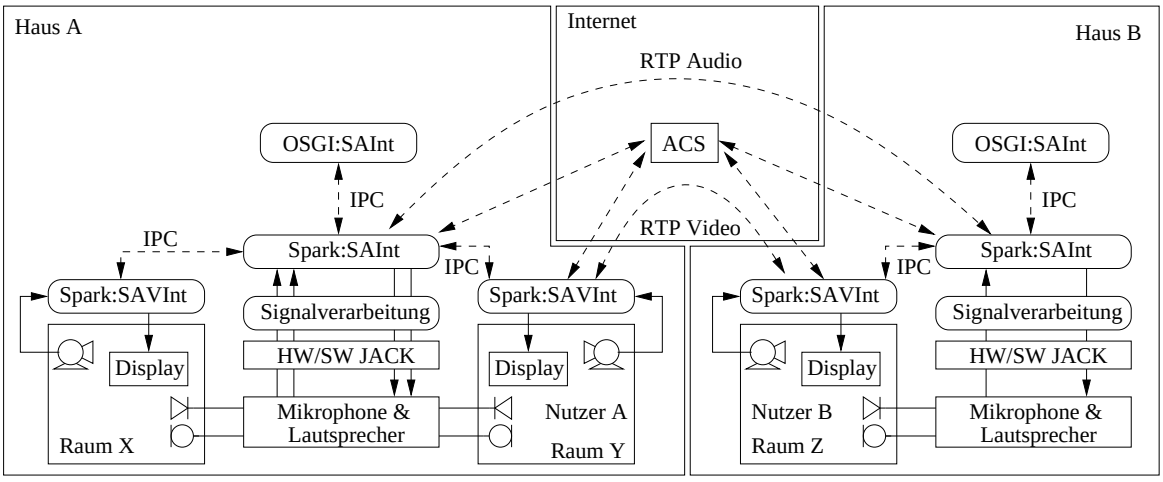


Abbildung 7.7: Blockschaltbild der Integration von *SAVInt*-Modulen in die *SAInt*-Architektur

In Abb. 7.7 ist ein Beispiel für die Kommunikation mit *SAVInt*-Modulen gegeben. Jedes *SAVInt*-Modul registriert sich bei einem laufenden *SAInt*-Modul über eine *IPC*-Schnittstelle mit der Information, welcher Raum durch die Kamera einsehbar ist. Ein *SAInt*-Modul kann mehrere *SAVInt*-Module steuern, wodurch im besten Fall alle Räume, welche durch die

angeschlossenen Mikrophone erreichbar, auch durch Kameras und Monitore versorgt sind. In Abb. 7.7 ist beispielhaft eine Anordnung für zwei Räume im Haus A und ein Raum in Haus B skizziert worden. Die Komponenten der *Middleware*-Schicht (*LMS*, Kontextquellen, etc.) bis auf den *SAInt*-Dienst wurden in dieser Skizze zur Vereinfachung weggelassen (vgl. Abb. 7.2).

Ein wesentlicher Vorteil der ambienten Kommunikation ist die Bewegungsfreiheit des Benutzers, so dass dieser sich frei im Raum und zwischen den Räumen bewegen kann. Diese Freiheit sollte bei der Integration von Videodaten mit berücksichtigt werden. Jedoch beinhaltet eine Kommunikation mit Videodaten zunächst den Nachteil, dass der überwiegende Teil von Kameratypen fest im Raum installiert wird und einen festen Blickwinkel hat. Benutzer, die sich frei bewegen, können somit aus dem Bild herauslaufen. Dies kann durch eine passende Wahl der Kameraposition und einer Weitwinkelaufnahme umgangen werden, jedoch führt dies zu einem Bild, in dem der Kommunikationspartner in vielen Positionen im Raum nur sehr klein dargestellt werden kann. Alternativ kann das in Kap. 4.4.4 (vgl. Abb. 4.24, S. 54) vorgestellte System zur Steuerung einer schwenk- und zoombaren Kamera genutzt werden. Hierzu wird im Videosystem ein *SAVInt*-Modul zur Übertragung und zum Empfang von Videodaten integriert, dessen Ausgang auf dem Bildschirm dargestellt wird. Das Audiosystem wird entsprechend der Abb. 7.3 (S. 94) um die Signalverarbeitung zur Echounterdrückung und Störgeräuschfilterung und um ein *SAInt*-Modul erweitert. Da die empfangenen Audiodaten des *SAInt*-Moduls über die Lautsprecher wiedergegeben werden, muss die Adaption der akustischen Strahlformung durch eine neue Logik gesteuert werden. Diese Logik sorgt dafür, dass, falls der entfernte Sprecher aktiv ist, die Adaption der Filter unterbrochen wird, um eine Ausrichtung der Kamera auf die Lautsprecher zu verhindern. Der Ablauf einer Kommunikation wird im Folgenden anhand eines Beispiels erläutert.

7.5.2 Kommunikationsbeispiel

Das Kommunikationsbeispiel nimmt an, dass der *SAInt*-Dienst im Haus A eine Kommunikation zwischen den Nutzern A und B mit Hilfe des *SAInt*-Moduls initiiert (vgl. Abb. 7.7). Das *SAInt*-Modul im Haus A sendet eine Verbindungsanfrage über den *ACS* an das *SAInt*-Modul im Haus B. Nachdem Nutzer B der Kommunikation zugestimmt hat, beginnen beide *SAInt*-Module die Audiodaten (vgl. Abb. 7.7, „*RTP* Audio“) zu den vom *ACS* übermittelten *IP*-Adressen zu senden.

Zeitgleich mit dem Start der Audioverbindung geben die *SAInt*-Module an die jeweiligen *SAVInt*-Module der Räume, in denen sich die Nutzer aufhalten, die Anweisung, eine Videoverbindung aufzubauen. Zu diesem Zweck registrieren sich die *SAVInt*-Module auf dem *ACS* und handeln eine Videoverbindung aus. Die Videodaten werden direkt mit einer *RTP*-Verbindung (vgl. Abb. 7.7, „*RTP* Video“) zwischen den *SAVInt*-Modulen ausgetauscht, so dass die Audiodaten und Videodaten getrennt übertragen werden. Eine getrennte Übertragung kann ohne Synchronisierung der Datenströme erfolgen, falls die Laufzeitdifferenz zwischen den beiden Datenströmen niedrig ist.

Die Videokommunikation des *SAVInt*-Moduls wird über die *IPC*-Schnittstelle des *SAInt*-Moduls kontrolliert. Sollte das *SAInt*-Modul durch den *SAInt*-Dienst die Beendigung der Verbindung signalisiert bekommen, so wird mit der Beendigung der akustischen Kommunikation auch die visuelle Kommunikation gestoppt.

7.5.3 *Follow-Me*-Fähigkeiten

Die *Follow-Me*-Fähigkeiten des Systems werden benötigt, sobald ein Benutzer den Raum wechselt. Entsprechend des obigen Beispiels nehmen wir an, dass der Nutzer A von Raum Y in den Raum X geht. In diesem Fall benachrichtigt der *SAInt*-Dienst das *SAInt*-Modul über den Positionswechsel des Nutzers. Das *SAInt*-Modul leitet den Audiodatenstrom in den Raum X um und stoppt über die *IPC*-Schnittstelle die Videoübertragung des *SAVInt*-Moduls aus Raum Y. Da in Raum X auch ein *SAVInt*-Modul verfügbar ist, initiiert das *SAInt*-Modul über die *IPC*-Schnittstelle eine Videoverbindung. Nach dem Aushandeln der Videoverbindung über den *ACS* startet diese mit einer leichten Verzögerung gegenüber der Audioverbindung. Im Gegensatz zur Audioverbindung, welche nahtlos die Räume wechseln kann, erzwingt die Videoverbindung bei jedem Raumwechsel einen Neuaufbau der *RTP*-Verbindung.

7.6 Demonstration

Im Rahmen des Amigo *Openday* im Februar 2008 wurde die ambiente Kommunikation zwischen Standorten in Deutschland, Frankreich und den Niederlanden demonstriert. Trotz der unterschiedlichen Ausstattung mit Hardware konnten die Komponenten der ambienten Kommunikation an allen Standorten verwendet werden. Dies wurde durch den modularen Aufbau der Software ermöglicht, welcher den Anforderungen eines Standortes entsprechend angepasst werden konnte. Zudem zeigte es die Flexibilität der Amigo *Middleware* in Bezug auf die Integration anderer Applikationen und Dienste. Ein Beispiel hierfür war die Nutzung des *SAInt*-Dienstes zur Kommunikation durch andere Applikationen. Hierbei nutzten die Applikationen die vom *SAInt*-Dienst exportierten *Webservice*-Schnittstellen zur Steuerung einer audio-visuellen Kommunikation. Der Standort in Deutschland verwendete die audio-visuelle Kamerasteuerung, um die Vorteile einer akustischen Kamerasteuerung zu demonstrieren.

8 Zusammenfassung

Im Rahmen dieser Arbeit wurde ein System zur akustischen Szenenanalyse entwickelt, welches fortlaufend die Identität und Position des aktuellen Sprechers ermittelt. Die Verwendung des Systems in einem Kommunikationsszenario führte zur Entwicklung einer audiovisuellen Sprecherprotokollierung, deren Fehlerrate durch eine Gesichtserkennung signifikant reduziert wurde. Des Weiteren wurden die Amigo *Middleware* und das System zur Verarbeitung von Kontextinformationen vorgestellt. Hierbei wurde die Einbindung der akustischen Szenenanalyse als Quelle von Kontextinformationen gezeigt. Anschließend wurde mit Hilfe der *Middleware* und den Amigo Diensten ein System zur ambienten Kommunikation realisiert. Dabei ermöglichte die Verfügbarkeit unterschiedlicher Kontextquellen eine kontextabhängige Steuerung.

Die zeitlichen Anforderungen des vernetzten Hauses an Informationsquellen wurde in dieser Arbeit als hoch eingestuft, da die Akzeptanz eines Systems durch seine Benutzer in Folge hoher Latenzen gefährdet ist. Die drei Schlüsselelemente der akustischen Signalverarbeitung in „intelligenten Umgebungen“ werden durch die automatische Spracherkennung, die akustische Szenenanalyse und die ambiente Kommunikation gebildet. Innerhalb dieser Arbeit wurden die Aspekte der akustischen Szenenanalyse und der ambienten Kommunikation näher untersucht.

Ausgehend von den zuvor identifizierten Forschungszielen wurde zunächst die Sprecherprotokollierung als Teil der akustischen Szenenanalyse betrachtet. Diese gliederte sich in die Aufgaben der Segmentierung der Daten in homogene Abschnitte und die anschließende Klassifikation dieser Segmente. Hierbei zeigte sich, dass die auf dem Bayes’schen Informationskriterium basierende Segmentierungstechnik sowohl von der Signalverarbeitung durch die akustische Strahlformung als auch von den Positionsdaten der Sprecher profitierte.

Die sequentielle Segmentierung und Identifikation von Sprechern in Datenströmen besaß den inhärenten Nachteil, dass frühzeitig getroffene Entscheidungen in der Segmentierung nicht rückgängig gemacht werden konnten. Dieser Nachteil resultierte aus den zeitlichen Anforderungen an die akustische Szenenanalyse, welche dem System Informationen mit einer möglichst geringen Latenz zur Verfügung stellen sollte. Da hierdurch weder iterative noch mehrstufige Verfahren verwendet werden können, wurde ein neuer Ansatz zur gleichzeitigen Segmentierung, Lokalisation und Sprecheridentifikation entwickelt. Grundidee dieses Ansatzes war die Verwendung eines *Hidden Markov Models* mit zeitveränderlichen Transitionswahrscheinlichkeiten, dessen Zustände die trainierten Sprecher repräsentierten. Die Berechnung der Transitionswahrscheinlichkeiten wurde realisiert über die Sprecherwechselinformationen, welche durch die akustische Positionsschätzung und das Bayes’sche Informationskriterium bereitgestellt wurden. Die Implementierung einer vorzeitigen Zurückverfolgung der Entscheidungen ermöglichte die Verwendung des Ansatzes auf kontinuierlichen Datenströmen mit geringer Latenz. Experimentell konnte gezeigt werden, dass der Median

der Entscheidungen für den aktuellen Sprecher bei weniger als einer halben Sekunde lag. Dabei führte die Begrenzung der maximalen Latenz auf zwei Sekunden nur zu einer geringen Erhöhung der Fehlerrate. Des Weiteren zeigten die Experimente, dass der neue Ansatz der gemeinsamen Segmentierung und Klassifikation höhere Klassifikationsraten erzielte als ein vergleichbares sequentielles Verfahren.

Die in dieser Arbeit betrachtete Umgebung war mit Mikrophonen und Kameras ausgestattet. Dies bot die Möglichkeit, die Sprecherprotokollierung in Kommunikationsszenarien durch Informationen aus der Bildverarbeitung zu verbessern. Das hierzu integrierte Videosystem ermöglichte die Detektion und Identifikation von Gesichtern. Ein Datenaustausch zwischen der akustischen Signalverarbeitung und der visuellen Datenverarbeitung führte zu einer Verbesserung beider Systeme. Die Kamera konnte durch die Kopplung der Systeme sowohl akustisch als auch anhand erkannter Gesichter automatisch gesteuert werden. Folglich war es möglich, die Kamera immer auf den aktuellen Sprecher auszurichten, selbst wenn dieser nicht in die Kamera schaute oder außerhalb des Kamerablickwinkels war. Detektierte und identifizierte das Videosystem das Gesicht eines Sprechers, so wurde diese Information an das System zur Sprecherprotokollierung weitergegeben. Die Integration der visuellen Informationen des Videosystems in den Prozess der akustischen Sprecherprotokollierung führte zu einer Erweiterung des zuvor vorgestellten Ansatzes. Die Emissionswahrscheinlichkeiten der *HMM*-Zustände wurden nun sowohl durch die akustischen Sprechermodelle als auch durch die visuellen Modelle der Nutzer bestimmt. Experimente zeigten, dass durch die Berücksichtigung der visuellen Informationen die Klassifikationsfehlerrate im Vergleich zu einem rein akustischen System um die Hälfte gesenkt werden konnte.

Ein weiteres Forschungsgebiet der akustischen Szenenanalyse ist die Identifikation akustischer Ereignisse, welche die aus der Sprecherprotokollierung bekannte Fragestellung „Wer spricht Wann und Wo?“ noch um die Komponente „Während Was passiert?“ erweitert. Im Rahmen dieser Arbeit wurden verschiedene Verfahren zur Modellierung der Ereignisse untersucht und die Verwendbarkeit der Merkmale aus der Sprecheridentifikation getestet. Zunächst wurden die Modellparameter zur Beschreibung der akustischen Ereignisse mittels eines „*Maximum Likelihood*“-Verfahrens geschätzt. Anschließend wurden Modelle mit dem diskriminativen Lernverfahren „*Maximum Mutual Information*“ trainiert. In Experimenten wurde gezeigt, dass die Modelle aus dem diskriminativen Lernverfahren eine niedrigere Klassifikationsfehlerrate ermöglichen als die Modelle aus der „*Maximum Likelihood*“-Parameterschätzung.

Die Datenbasis zur akustischen Ereignisdetektion stammte aus dem Bereich der professionell genutzten Arbeitsumgebungen und wurde im Projekt *CHIL* erstellt. Da die Datenbasis aus mehrkanaligen Aufnahmen bestand, konnte eine Verbesserung der Klassifikationsrate durch die Auswahl und Kombination von Kanälen erzielt werden. Die mittlere Klassifikationsrate lag im Fall der Einzelerkennung bei über $90\,\%$ und bei der Kombination mehrerer Kanäle sogar über $93\,\%$.

Die Gewinnung von Kontextinformationen war der erste Schritt zum Aufbau einer durch den Benutzer als „intelligent“ wahrgenommenen Umgebung. Erst die Integration von Kontextquellen, wie z. B. der akustischen Szenenanalyse, in einen Verbund von Diensten und Applikationen erlaubte das Treffen von kontextabhängigen und somit „intelligenten Entscheidungen“. Im Rahmen dieser Arbeit wurde die Integration der akustischen Szenenanalyse in die Amigo *Middleware* vorgestellt, wobei ein Schwerpunkt auf das Kontextmanagement gelegt wurde. Das Amigo System zum Kontextmanagement verwendete einen Kontextbroker

als zentralen Anlaufpunkt für Kontextquellen und Kontextnutzer. Die Interaktion der Dienste untereinander wurde über standardisierte *Webservice*-Schnittstellen realisiert, so dass eine offene, dienstorientierte Softwarearchitektur gebildet wurde.

Aufbauend auf der Amigo *Middleware* und den vorhandenen Kontextquellen wurde im letzten Teil der Arbeit ein System zur ambienten Kommunikation vorgestellt, welches als Beispiel einer kontextbewussten Anwendung angesehen werden kann. Hierbei wurden die Komponenten zur akustischen Signalverarbeitung vorgestellt, welche zur Unterdrückung von Echos und Störgeräuschen notwendig sind. Diese aus der Literatur entnommenen Verfahren wurden in ein echtzeitfähiges System integriert und um Komponenten zur Audio- und Videodatenkompression sowie zum Datenaustausch ergänzt. Hierdurch war es möglich, eine echtzeitfähige Kommunikation zwischen zwei beliebigen Standorten über ein gemeinsames *IP*-Netzwerk aufzubauen und gleichzeitig eine Datenverteilung im lokalen System vorzunehmen.

Die Steuerung der Datenströme innerhalb der ambienten Kommunikation erfolgte kontextbasiert durch die in der *Middleware* vorhandenen Daten über die Nutzerpositionen. Dabei stellte die audio-visuelle Sprecherprotokollierung, als Teil der akustischen Szenenanalyse, eine mögliche Kontextquelle neben anderen Verfahren zur Positionsbestimmung dar. Im Vergleich mit anderen Systemen, wie z. B. *RFID*-basierter Positionsschätzung, bot die akustische Szenenanalyse den Vorteil, dass keine zusätzlichen Geräte durch den Benutzer mitgeführt werden mussten. Das System der ambienten Kommunikation nutzte unter anderem die vorhandenen Kontextinformationen, um automatisiert die Sitzungsverwaltung für Benutzer durchzuführen. Des Weiteren standen dem Nutzer während der Kommunikation *Follow-Me*-Fähigkeiten zur Verfügung, d. h. der Nutzer konnte sich frei im Raum und zwischen den Räumen bewegen, während die kontextbewusste Steuerung die Audio- und Videodaten der Kommunikation dem Nutzer automatisch folgen ließ. Die Verwendung einer schwenk- und zoombaren Kamera, welche mit den kombinierten Ergebnissen der akustischen Positionsschätzung und der Gesichtsdetektion gesteuert wurde, ermöglichte eine automatische Ausrichtung der Kamera auf den aktuellen Sprecher.

Ausblick

Die hier vorgestellten Systeme zur akustischen Szenenanalyse und zur ambienten Kommunikation verwendeten vorab trainierte Modelle, die aus einer initialen Trainingsphase stammten. Der Aufwand eines solchen Trainings steht im Gegensatz zu den Ideen der ambienten Intelligenz, da dort die automatische Anpassung des Systems an den Benutzer gefordert wird. Daher ist der nächste Entwicklungsschritt des Systems, dass ein automatisches Training der Benutzer und der Hardwareausstattung durchgeführt wird. Anstatt Modelle für jeden Nutzer vorab zu trainieren, wird das System eigenständig neue Benutzer erkennen und für diese neue Modelle trainieren. Somit wird das System sukzessiv alle Nutzer beobachten und deren Modelle mit zunehmender Datenmenge immer besser trainieren. Im Bezug auf die Hardware wird das System um selbstkonfigurierende und selbstlernende Komponenten erweitert, so dass es z. B. eigenständig die Geometrie und Position von Mikrophongruppen bestimmen kann.

A Anhang

A.1 Herleitung $\triangle BIC$

Die *Likelihood* der Hypothese H_0 ist gegeben mit:

$$p(\boldsymbol{X}_{1:N_w}|H_0) = \prod_{k=1}^{N_w} \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_0|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left((\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)\right)\right) \tag{A.1}$$

$$= \left((2\pi)^D|\boldsymbol{\Sigma}_0|\right)^{\frac{-N_w}{2}} \exp\left(-\frac{1}{2}\sum_{k=1}^{N_w}\left((\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)\right)\right) \tag{A.2}$$

Logarithmieren der Dichtefunktion ergibt die *Log-Likelihood*:

$$\log\left(p(\boldsymbol{X}_{1:N_w}|H_0)\right)$$

$$= -\frac{DN_w}{2}\log(2\pi) - \frac{N_w}{2}log(|\boldsymbol{\Sigma}_0|) - \frac{1}{2}\sum_{k=1}^{N_w}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0) \tag{A.3}$$

Für die weiteren Umformungen werden einige Eigenschaften von Matrizen verwendet, die im Folgenden angegeben werden. Wenn die Matrix $\boldsymbol{A}$ bestehend aus den Elementen A_{ij} durch das Produkt zweier Vektoren $\boldsymbol{a}$ und $\boldsymbol{b}$ mit

$$\boldsymbol{A} = (A_{ij}) = \boldsymbol{a}\cdot\boldsymbol{b}^T = (a_i \cdot b_j) \tag{A.4}$$

dargestellt werden kann, so gilt für die Spur von $\boldsymbol{A}$:

$$spur(\boldsymbol{A}) = \sum_{i=1}^{N} A_{ii} = \sum_{i=1}^{N} a_i b_i = \boldsymbol{a}^T\boldsymbol{b}. \tag{A.5}$$

Somit kann die Summe aus Gl. A.3 umgeformt werden zu:

$$\sum_{k=1}^{N_w} \underbrace{(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}}_{\boldsymbol{a}_k^T} \underbrace{(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)}_{\boldsymbol{b}_k} = \sum_{k=1}^{N_w} spur\left(\underbrace{\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)}_{\boldsymbol{a}_k}\underbrace{(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T}_{\boldsymbol{b}_k^T}\right) \tag{A.6}$$

$$= spur\left(\sum_{k=1}^{N_w}\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\right)$$

$$= spur\left(\boldsymbol{\Sigma}_0^{-1}\sum_{k=1}^{N_w}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\right). \tag{A.7}$$

Da $\boldsymbol{\Sigma}_0$ mit

$$\boldsymbol{\Sigma}_0 = \frac{1}{N_w}\sum_{k=1}^{N_w}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T \tag{A.8}$$

aus den Merkmalsvektoren $\boldsymbol{X}_{1:N_w}$ geschätzt wird, folgt für Gl. A.7:

$$\sum_{k=1}^{N_w}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_0) = \textit{spur}\left(\boldsymbol{\Sigma}_0^{-1}\cdot N_w\cdot\boldsymbol{\Sigma}_0\right) \tag{A.9}$$

$$= N_w\cdot D. \tag{A.10}$$

Somit folgt für Gl. A.3:

$$\log\left(p(\boldsymbol{X}_{1:N_w}|H_0)\right) = -\frac{DN_w}{2}\log(2\pi) - \frac{N_w}{2}log(|\boldsymbol{\Sigma}_0|) - \frac{1}{2}N_w\cdot D \tag{A.11}$$

$$= -\frac{N_w}{2}\log(|\boldsymbol{\Sigma}_0|) - \frac{DN_w}{2}(1+\log(2\pi)). \tag{A.12}$$

Des Weiteren ist die *Likelihood* der Hypothese H_1 gegeben durch:

$$p(\boldsymbol{X}_{1:N_w}|H_1) = \prod_{k=1}^{N_w/2}\frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_1|^{\frac{1}{2}}}\cdot\exp\left(-\frac{1}{2}\left((\boldsymbol{x}(k)-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_1)\right)\right)$$
$$\prod_{k=N_w/2+1}^{N_w}\frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}_2|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\left((\boldsymbol{x}(k)-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_2)\right)\right) \tag{A.13}$$

$$= \left((2\pi)^D|\boldsymbol{\Sigma}_1|\right)^{\frac{-N_w}{4}}\exp\left(-\frac{1}{2}\sum_{k=1}^{N_w/2}\left((\boldsymbol{x}(k)-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_1)\right)\right)$$
$$\left((2\pi)^D|\boldsymbol{\Sigma}_2|\right)^{\frac{-N_w}{4}}\exp\left(-\frac{1}{2}\sum_{k=N_w/2+1}^{N_w}\left((\boldsymbol{x}(k)-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}(k)-\boldsymbol{\mu}_2)\right)\right). \tag{A.14}$$

Das Logarithmieren der Dichtefunktion der Hypothese H_1 und die Verwendung von Gl. A.4 und Gl. A.5 führt auf:

$$\log\left(p(\boldsymbol{X}_{1:N_w}|H_1)\right) = -\frac{DN_w}{2}\log(2\pi) - \frac{N_w}{4}log(|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|) - 2\frac{1}{2}\frac{DN_w}{2}. \tag{A.15}$$

Entsprechend der Definition für ΔBIC [DW00] berechnet sich dessen Wert aus der Differenz der Gleichungen Gl. A.12 und Gl. A.15 und deren zugehörigen Gewichtsterme für die Modellkomplexität zu:

$$\Delta BIC = BIC(H_1) - BIC(H_0) \tag{A.16}$$

$$= -\frac{N_w}{4}log(|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|) - \frac{DN_w}{2}(1+\log(2\pi)) - \xi\frac{m_1}{2}\log N_w$$
$$+\frac{N_w}{2}log(|\boldsymbol{\Sigma}_0|) + \frac{DN_w}{2}(1+\log(2\pi)) + \xi\frac{m_0}{2}\log N_w \tag{A.17}$$

$$= \frac{N_w}{2}log(|\boldsymbol{\Sigma}_0|) - \frac{N_w}{4}log(|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|) - \xi\frac{m_0}{4}\log N_w. \tag{A.18}$$

Im letzten Schritt wurde die Vereinfachung verwendet, dass die Hypothese H_1 doppelt so viele Modellparameter besitzt, wie die Hypothese H_0 ($m_1 = 2m_0$).

A.2 Herleitung *MMI*-Parameterschätzung

Gegeben seien jeweils N_k Merkmalsvektoren $\boldsymbol{X}_{k,1:N_k} = [\boldsymbol{x}_k(1), \ldots, \boldsymbol{x}_k(N_k)]$ für jede der $k = 1, \ldots, K$ Klassen, welche zur Parameterschätzung der Modelle verwendet werden sollen. Jede Klasse soll durch ein *GMM* mit M Mischungsverteilungen beschrieben werden. Die Zufallsvariabel der Klassenzugehörigkeit eines Merkmalsvektors $\boldsymbol{x}_k(n)$ werde mit Ω und die Zufallsvariabel der Zugehörigkeit zu einer Mischungsverteilung mit Z bezeichnet. Das Ziel der *MMI*-Parameterschätzung ist die Maximierung der Anzahl der korrekt klassifizierten Trainingsmerkmale [LP96]. Folglich muss für die Parameterschätzung der i-ten Klasse

$$P(\Omega = i|\boldsymbol{X}_{i,1:N_i};\boldsymbol{\Theta}) = \prod_{n=1}^{N_i} \frac{p(\boldsymbol{x}_i(n)|\Omega = i;\boldsymbol{\Theta}_i) \cdot P(\Omega = i)}{p(\boldsymbol{x}_i(n))} \tag{A.19}$$

$$= \prod_{n=1}^{N_i} \frac{p(\boldsymbol{x}_i(n)|\Omega = i;\boldsymbol{\Theta}_i) \cdot P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k;\boldsymbol{\Theta}_k) \cdot P(\Omega = k)}. \tag{A.20}$$

maximiert werden. Die Parameterschätzung soll anhand des logarithmierten Ausdrucks aus Gl. A.20 erfolgen.

$$Q_i(\boldsymbol{\Theta}) = \log\left(P(\Omega = i|\boldsymbol{X}_{i,1:N};\boldsymbol{\Theta})\right) \tag{A.21}$$

$$= \log\left(\prod_{n=1}^{N_i} \frac{p(\boldsymbol{x}_i(n)|\Omega = i;\boldsymbol{\Theta}_i) \cdot P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k;\boldsymbol{\Theta}_k) \cdot P(\Omega = k)}\right)$$

$$= \sum_{n=1}^{N_i} \log\left(\frac{p(\boldsymbol{x}_i(n)|\Omega = i;\boldsymbol{\Theta}_i) \cdot P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k;\boldsymbol{\Theta}_k) \cdot P(\Omega = k)}\right)$$

$$= \sum_{n=1}^{N_i} \Bigg[\log\left(p(\boldsymbol{x}_i(n)|\Omega = i;\boldsymbol{\Theta}_i) \cdot P(\Omega = i)\right) - \log\left(\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k;\boldsymbol{\Theta}_k) \cdot P(\Omega = k)\right)\Bigg] \tag{A.22}$$

Zunächst erfolgt die Berechnung des Gradienten zur Bestimmung der Parameterwerte $\boldsymbol{\Theta}_i$ durch:

$$
\begin{aligned}
&\nabla_{\boldsymbol{\Theta}_i} Q_i(\boldsymbol{\Theta}) \\
&= \sum_{n=1}^{N_i} \left[\frac{\nabla_{\boldsymbol{\Theta}_i} \left[p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) \right]}{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i)} - \frac{\nabla_{\boldsymbol{\Theta}_i} \left[\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k) P(\Omega = k) \right]}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k) P(\Omega = k)} \right] \qquad \text{(A.23)} \\
&= \sum_{n=1}^{N_i} \left[\frac{\nabla_{\boldsymbol{\Theta}_i} \left[p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) \right]}{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i)} - \frac{\nabla_{\boldsymbol{\Theta}_i} \left[p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) P(\Omega = i) \right]}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k) P(\Omega = k)} \right] \\
&= \sum_{n=1}^{N_i} \left[\underbrace{\left(1 - \frac{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k) P(\Omega = k)} \right)}_{\psi_i(n)} \nabla_{\boldsymbol{\Theta}_i} \left[\log \left(p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) \right) \right] \right] \\
&= \sum_{n=1}^{N_i} \left[\psi_i(n) \nabla_{\boldsymbol{\Theta}_i} \left[\log \left(p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) \right) \right] \right] . \qquad \text{(A.24)}
\end{aligned}
$$

Im Folgenden wird der Ausdruck

$$
\psi_i(n) = \left(1 - \frac{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k) P(\Omega = k)} \right) \qquad \text{(A.25)}
$$

zur Abkürzung der Schreibweise verwendet. Er kann interpretiert werden als die Wahrscheinlichkeit, dass ein Merkmalsvektor $\boldsymbol{x}_i(n)$ mit den aktuellen Modellparametern aller Klassen falsch klassifiziert wird. Die den *Likelihoods* $p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i)$ zugrundeliegenden Verteilungsdichtefunktionen sind Gauß'sche Mischungsverteilungen. Sie bestehen aus jeweils M Einzelverteilungen $p(\boldsymbol{x}_i(n)|\Omega = i, Z = m; \boldsymbol{\Theta}_i)$, welche mit

$$
c_{i,m} = P(Z = m|\Omega = i) \qquad m = 1, \ldots, M \qquad \text{(A.26)}
$$

gewichtet sind. Folglich sind die *Likelihoods* der Verteilungsdichtefunktionen mit

$$
p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) \;=\; \sum_{m=1}^{M} c_{i,m} \cdot p(\boldsymbol{x}_i(n)|\Omega = i, Z = m; \boldsymbol{\Theta}_i) \qquad \text{(A.27)}
$$

gegeben. Für die Berechnung des Mittelwertvektors oder der Kovarianzmatrix der j-ten Einzelverteilung folgt

$$\begin{aligned}
&\nabla_{\boldsymbol{\Theta}_{i,j}} \left[\log\left(p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)\right)\right] \\
=&\nabla_{\boldsymbol{\Theta}_{i,j}} \left[\log\left(\sum_{m=1}^{M} c_{i,m}\cdot p(\boldsymbol{x}_i(n)|\Omega=i,Z=m;\boldsymbol{\Theta}_i)\right)\right] && \text{(A.28)}\\
=&\frac{\nabla_{\boldsymbol{\Theta}_{i,j}}\left[c_{i,j}\cdot p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)\right]}{p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)} && \text{(A.29)}\\
=&\frac{c_{i,j}}{p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)}\nabla_{\boldsymbol{\Theta}_{i,j}}\left[p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)\right], && \text{(A.30)}
\end{aligned}$$

wobei die Umformung von Gl. A.29 auf Gl. A.30 berücksichtigt, dass der Gradient nicht für die Mischungsgewichte betrachtet wird. Die Anwendung der Bayes'schen Regel für bedingte Wahrscheinlichkeiten auf Gl. A.30 in der Form

$$\begin{aligned}
\frac{P(Z=j|\Omega=i)}{p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)} &= \frac{P(Z=j|\boldsymbol{x}_i(n),\Omega=i;\boldsymbol{\Theta}_i)}{p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)} && \text{(A.31)}\\
\Leftrightarrow \frac{c_{i,j}}{p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)} &= \frac{\gamma_{i,j}(n)}{p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)} && \text{(A.32)}
\end{aligned}$$

mit

$$\gamma_{i,j}(n) = P(Z=j|\boldsymbol{x}_i(n),\Omega=i;\boldsymbol{\Theta}_i) \qquad \text{(A.33)}$$

führt auf:

$$\begin{aligned}
&\nabla_{\boldsymbol{\Theta}_{i,j}} \left[\log\left(p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)\right)\right] \\
=&\frac{\gamma_{i,j}(n)}{p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)}\nabla_{\boldsymbol{\Theta}_{i,j}}\left[p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)\right] && \text{(A.34)}\\
=&\gamma_{i,j}(n)\nabla_{\boldsymbol{\Theta}_{i,j}}\left[\log\left(p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)\right)\right]. && \text{(A.35)}
\end{aligned}$$

Die Bestimmung der Mittelwertvektoren $\boldsymbol{\mu}_{i,j}$ der j-ten Einzelverteilung kann unter Verwendung von [BSMM01] mit

$$\begin{aligned}
\nabla_{\boldsymbol{\mu}_{i,j}} \log\left(p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)\right) &= \nabla_{\boldsymbol{\mu}_{i,j}} \log\left(\frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j})^T\boldsymbol{\Sigma}_{i,j}^{-1}(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j})\right)}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_{i,j}|}}\right) && \text{(A.36)}\\
&= \boldsymbol{\Sigma}_{i,j}^{-1}\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right) && \text{(A.37)}
\end{aligned}$$

erfolgen. Das Einsetzen der Teilergebnisse aus Gl. A.35 und Gl. A.37 in die Gradientengleichung aus Gl. A.24 liefert die Bestimmungsgleichung für die geschätzten Mittelwertvektoren $\widehat{\boldsymbol{\mu}}_{i,j}$ mit:

$$\begin{aligned}
&\nabla_{\boldsymbol{\mu}_{i,j}} Q_i(\boldsymbol{\Theta})\Big|_{\boldsymbol{\mu}_{i,j}=\widehat{\boldsymbol{\mu}}_{i,j}} \stackrel{!}{=} 0 && \text{(A.38)}\\
\Leftrightarrow 0 =& \sum_{n=1}^{N_i}\left[\psi_i(n)\cdot\gamma_{i,j}(n)\cdot\boldsymbol{\Sigma}_{i,j}^{-1}\left(\boldsymbol{x}_i(n)-\widehat{\boldsymbol{\mu}}_{i,j}\right)\right] \\
\Leftrightarrow \widehat{\boldsymbol{\mu}}_{i,j} =& \frac{\sum_{n=1}^{N_i}\left[\psi_i(n)\cdot\gamma_{i,j}(n)\cdot\boldsymbol{x}_i(n)\right]}{\sum_{n=1}^{N_i}\psi_i(n)\cdot\gamma_{i,j}(n)}. && \text{(A.39)}
\end{aligned}$$

Entsprechend der Herleitung für die Mittelwertvektoren $\widehat{\boldsymbol{\mu}}_{i,j}$ wird für die Schätzung der Kovarianzmatrizen $\widehat{\boldsymbol{\Sigma}}_{i,j}$ zunächst der Gradient aus Gl. A.35 mit Hilfe von [Fuk90] bestimmt:

$$\begin{aligned}
&\nabla_{\boldsymbol{\Sigma}_{i,j}} \log\left(p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)\right)\\
=&\nabla_{\boldsymbol{\Sigma}_{i,j}} \log\left(\frac{\exp\left(-\frac{1}{2}\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)^T \boldsymbol{\Sigma}_{\boldsymbol{i,j}}^{-1}\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)\right)}{\sqrt{(2\pi)^D\left|\boldsymbol{\Sigma}_{i,j}\right|}}\right) \qquad \text{(A.40)}\\
=&-\frac{1}{2}\nabla_{\boldsymbol{\Sigma}_{i,j}}\left[\log\left(|\boldsymbol{\Sigma}_{i,j}|\right)+\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)^T\boldsymbol{\Sigma}_{i,j}^{-1}\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)\right]\\
=&-\frac{1}{2}\left(\boldsymbol{\Sigma}_{i,j}^{-1}-\boldsymbol{\Sigma}_{i,j}^{-1}\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)^T\boldsymbol{\Sigma}_{i,j}^{-1}\right). \qquad \text{(A.41)}
\end{aligned}$$

Setzt man die Teilergebnisse aus Gl. A.41 und Gl. A.35 in die Gradientengleichung aus Gl. A.24 ein, so folgt:

$$\begin{aligned}
&\nabla_{\boldsymbol{\Sigma}_{i,j}} Q_i(\boldsymbol{\Theta})\Big|_{\boldsymbol{\Sigma}_{i,j}=\widehat{\boldsymbol{\Sigma}}_{i,j}} \stackrel{!}{=} 0 \qquad \text{(A.42)}\\
\Leftrightarrow 0=&\sum_{n=1}^{N_i}\left[\psi_i(n)\cdot\gamma_{i,j}(n)\left(\widehat{\boldsymbol{\Sigma}}_{i,j}^{-1}-\widehat{\boldsymbol{\Sigma}}_{i,j}^{-1}\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)^T\widehat{\boldsymbol{\Sigma}}_{i,j}^{-1}\right)\right]\\
\Leftrightarrow\widehat{\boldsymbol{\Sigma}}_{i,j}=&\frac{\sum_{n=1}^{N_i}\left[\psi_i(n)\cdot\gamma_{i,j}(n)\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)\left(\boldsymbol{x}_i(n)-\boldsymbol{\mu}_{i,j}\right)^T\right]}{\sum_{n=1}^{N_i}\psi_i(n)\cdot\gamma_{i,j}(n)}. \qquad \text{(A.43)}
\end{aligned}$$

Die Schätzung der Mischungsgewichte $\widehat{c}_{i,j}$ erfolgt mit Hilfe des Lagrange-Multiplikators, der in die Optimierung aus Gl. A.22 mit einbezogen wird:

$$Q_i'(\boldsymbol{\Theta},\lambda)=\sum_{n=1}^{N_i}\log\left(\frac{p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)P(\Omega=i)}{\sum_{k=1}^{K}p(\boldsymbol{x}_i(n)|\Omega=k;\boldsymbol{\Theta}_k)P(\Omega=k)}\right)+\lambda\left(\sum_{m=1}^{M}c_{i,m}-1\right) \qquad \text{(A.44)}$$

Die Berechnung des Gradienten für den Ausdruck in Gl. A.44 liefert die Bestimmungsgleichung für $c_{i,j}$ mit:

$$\begin{aligned}
&\nabla_{c_{i,j}} Q_i'(\boldsymbol{\Theta},\lambda)\Big|_{c_{i,j}=\widehat{c}_{i,j}} \stackrel{!}{=} 0 \qquad \text{(A.45)}\\
\Leftrightarrow&\sum_{n=1}^{N_i}\left[\frac{p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)P(\Omega=i)}{p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)P(\Omega=i)}\right.\\
&\left.-\frac{p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)P(\Omega=i)}{\sum_{k=1}^{K}p(\boldsymbol{x}_i(n)|\Omega=k;\boldsymbol{\Theta}_k)P(\Omega=k)}\right]+\lambda=0\\
\Leftrightarrow&\sum_{n=1}^{N_i}\left[\psi_i(n)\frac{p(\boldsymbol{x}_i(n)|\Omega=i,Z=j;\boldsymbol{\Theta}_i)}{p(\boldsymbol{x}_i(n)|\Omega=i;\boldsymbol{\Theta}_i)}\right]+\lambda=0. \qquad \text{(A.46)}
\end{aligned}$$

Unter Verwendung von Gl. A.32 folgt:

$$\sum_{n=1}^{N_i} \left[\psi_i(n) \frac{\gamma_{i,j}(n)}{\widehat{c}_{i,j}} \right] + \lambda = 0 \tag{A.47}$$

$$\Leftrightarrow \widehat{c}_{i,j} = \frac{-1}{\lambda} \sum_{n=1}^{N_i} \psi_i(n) \cdot \gamma_{i,j}(n). \tag{A.48}$$

Die Summation der M-Gleichungen aus Gl. A.48 führt mit

$$\sum_{m=1}^{M} c_{i,m} \quad = \quad 1 \tag{A.49}$$

zur Bestimmung des Lagrange-Multiplikators:

$$-\lambda \quad = \quad \sum_{n=1}^{N_i} \left(1 - \frac{p(\boldsymbol{x}_i(n)|\Omega = i; \boldsymbol{\Theta}_i) P(\Omega = i)}{\sum_{k=1}^{K} p(\boldsymbol{x}_i(n)|\Omega = k; \boldsymbol{\Theta}_k) P(\Omega = k)} \right). \tag{A.50}$$

Somit folgt für die Mischungsgewichte $\widehat{c}_{i,j}$:

$$\widehat{c}_{i,j} = \frac{\sum_{n=1}^{N_i} \psi_i(n) \cdot \gamma_{i,j}(n)}{\sum_{n=1}^{N_i} \psi_i(n) \cdot}. \tag{A.51}$$

Die *MMI*-Parameterschätzung ist ein *EM*-Algorithmus. Im ersten Schritt (*Expectation*) werden die Erwartungswerte der Wahrscheinlichkeit einer Fehlklassifikation (vgl. Gl. A.25) und die Zugehörigkeit zu einer Mischungsverteilung (vgl. Gl. A.33) mit Hilfe der aktuellen Modellparameter geschätzt. Im zweiten Schritt (*Maximization*) werden die im vorherigen Schritt berechneten Werte verwendet, um eine neue Schätzung der Modellparameter (vgl. Gl. A.51, Gl. A.39, Gl. A.43) durchzuführen und somit die Zielfunktion (Gl. A.20) zu maximieren.

A.3 Experimentelle Ergebnisse der Ereignisdetektion

Die folgenden zwei Tabellen enthalten die Klassifikationsraten der Ereignisidentifikation für jedes einzelne Mikrophon im Raum. Tab. A.1 gibt die Ergebnisse für die Testdaten auf DVD 2 und Tab. A.2 die Ergebnisse für DVD 3 wieder. Die beiden letzten Zeilen geben die beste und die schlechteste Klassifikationsrate für jedes Ereignis wieder, um die Spannbreite der Klassifikationsraten zwischen den 22 Mikrophonen aufzuzeigen.

	ap	cl	cm	co	do	ds	kj	kn	kt	la	pr	pw	st	un
Mik. 1	100,00	100,00	85,71	90,91	100,00	95,24	95,24	100,00	96,00	95,24	86,11	82,76	91,67	76,09
Mik. 2	100,00	100,00	89,29	90,91	100,00	95,24	95,24	100,00	96,00	95,24	88,89	82,76	87,50	80,43
Mik. 3	100,00	100,00	89,29	95,45	100,00	95,24	95,24	100,00	92,00	90,48	86,11	86,21	91,67	82,61
Mik. 4	100,00	100,00	89,29	90,91	85,00	85,71	100,00	100,00	100,00	90,48	91,67	96,55	83,33	80,43
Mik. 5	100,00	100,00	92,86	90,91	100,00	95,24	95,24	100,00	92,00	85,71	88,89	79,31	91,67	82,61
Mik. 6	100,00	100,00	96,43	90,91	80,00	85,71	95,24	100,00	92,00	90,48	94,44	89,66	87,50	82,61
Mik. 7	100,00	100,00	92,86	90,91	100,00	95,24	95,24	100,00	96,00	95,24	88,89	86,21	91,67	86,96
Mik. 8	100,00	100,00	89,29	95,45	100,00	95,24	95,24	100,00	96,00	90,48	86,11	82,76	91,67	82,61
Mik. 9	100,00	100,00	82,14	100,00	100,00	90,48	100,00	100,00	92,00	95,24	91,67	79,31	91,67	86,96
Mik. 10	100,00	100,00	85,71	100,00	100,00	95,24	100,00	100,00	92,00	80,95	91,67	82,76	91,67	86,96
Mik. 11	100,00	100,00	85,71	100,00	100,00	95,24	95,24	100,00	92,00	85,71	86,11	75,86	87,50	84,78
Mik. 12	100,00	100,00	85,71	100,00	100,00	95,24	100,00	100,00	92,00	85,71	97,22	72,41	91,67	86,96
Mik. 13	100,00	100,00	82,14	86,36	100,00	95,24	100,00	100,00	96,00	95,24	80,56	86,21	87,50	89,13
Mik. 14	100,00	100,00	85,71	86,36	100,00	95,24	90,48	100,00	92,00	90,48	77,78	89,66	91,67	89,13
Mik. 15	100,00	100,00	85,71	77,27	100,00	95,24	80,95	100,00	92,00	95,24	80,56	89,66	91,67	86,96
Mik. 16	100,00	100,00	89,29	86,36	100,00	85,71	95,24	93,75	92,00	95,24	88,89	93,10	75,00	76,09
Mik. 17	100,00	100,00	89,29	90,91	100,00	85,71	95,24	93,75	92,00	90,48	83,33	93,10	75,00	71,74
Mik. 18	100,00	100,00	89,29	86,36	100,00	85,71	100,00	93,75	92,00	90,48	88,89	96,55	83,33	78,26
Mik. 19	100,00	100,00	89,29	90,91	95,00	85,71	100,00	87,50	92,00	95,24	91,67	96,55	83,33	76,09
Mik. 20	100,00	100,00	89,29	90,91	100,00	85,71	100,00	87,50	88,00	100,00	88,89	93,10	83,33	76,09
Mik. 21	100,00	100,00	92,86	90,91	100,00	85,71	100,00	87,50	96,00	95,24	94,44	93,10	83,33	73,91
Mik. 22	100,00	100,00	89,29	90,91	100,00	85,71	95,24	93,75	88,00	90,48	91,67	96,55	83,33	76,09
Minimum	100,00	100,00	82,14	77,27	80,00	85,71	80,95	87,50	88,00	80,95	77,78	72,41	75,00	71,74
Maximum	100,00	100,00	96,43	100,00	100,00	95,24	100,00	100,00	100,00	100,00	97,22	96,55	91,67	89,13

Tabelle A.1: Klassifikationsraten der Ereignisse je Kanal für die Testdaten (DVD 2)

	ap	cl	cm	co	do	ds	kj	kn	kt	la	pr	pw	st	un
Mik. 1	100,00	100,00	96,00	90,48	100,00	95,00	86,96	88,24	100,00	90,48	72,09	87,50	90,48	80,95
Mik. 2	100,00	100,00	92,00	90,48	100,00	95,00	86,96	88,24	100,00	90,48	67,44	91,67	85,71	80,95
Mik. 3	100,00	100,00	96,00	90,48	100,00	95,00	91,30	100,00	100,00	90,48	72,09	87,50	85,71	83,33
Mik. 4	100,00	100,00	96,00	95,24	95,00	95,00	95,65	88,24	100,00	90,48	74,42	95,83	90,48	78,57
Mik. 5	100,00	100,00	92,00	95,24	100,00	95,00	95,65	94,12	100,00	90,48	58,14	87,50	90,48	83,33
Mik. 6	100,00	100,00	92,00	90,48	90,00	95,00	91,30	100,00	100,00	95,24	76,74	95,83	95,24	85,71
Mik. 7	100,00	100,00	92,00	90,48	100,00	95,00	95,65	94,12	100,00	90,48	74,42	87,50	90,48	83,33
Mik. 8	100,00	100,00	92,00	90,48	100,00	95,00	95,65	94,12	100,00	85,71	60,47	87,50	90,48	83,33
Mik. 9	100,00	100,00	92,00	95,24	100,00	95,00	95,65	94,12	100,00	95,24	72,09	91,67	85,71	83,33
Mik. 10	100,00	100,00	92,00	85,71	100,00	95,00	91,30	88,24	100,00	90,48	74,42	87,50	85,71	83,33
Mik. 11	100,00	100,00	96,00	90,48	100,00	95,00	91,30	88,24	100,00	85,71	69,77	87,50	85,71	83,33
Mik. 12	100,00	100,00	92,00	95,24	100,00	95,00	91,30	88,24	100,00	85,71	69,77	87,50	90,48	83,33
Mik. 13	95,00	100,00	96,00	90,48	100,00	95,00	100,00	94,12	100,00	100,00	62,79	83,33	90,48	80,95
Mik. 14	90,00	100,00	96,00	80,95	100,00	95,00	91,30	88,24	100,00	95,24	60,47	87,50	95,24	83,33
Mik. 15	90,00	100,00	84,00	80,95	100,00	95,00	86,96	100,00	100,00	90,48	67,44	87,50	95,24	78,57
Mik. 16	100,00	100,00	96,00	95,24	100,00	95,00	91,30	94,12	95,00	95,24	79,07	91,67	90,48	78,57
Mik. 17	100,00	100,00	96,00	95,24	100,00	95,00	86,96	88,24	100,00	90,48	76,74	95,83	80,95	83,33
Mik. 18	100,00	100,00	96,00	95,24	100,00	95,00	91,30	94,12	100,00	85,71	76,74	100,00	90,48	80,95
Mik. 19	100,00	100,00	96,00	95,24	100,00	95,00	95,65	88,24	100,00	85,71	81,40	95,83	90,48	78,57
Mik. 20	100,00	100,00	96,00	90,48	100,00	95,00	91,30	94,12	95,00	90,48	76,74	91,67	90,48	80,95
Mik. 21	100,00	100,00	96,00	95,24	100,00	95,00	91,30	94,12	100,00	90,48	76,74	95,83	90,48	80,95
Mik. 22	100,00	100,00	96,00	95,24	100,00	95,00	86,96	94,12	100,00	95,24	76,74	91,67	95,24	80,95
Minimum	90,00	100,00	84,00	80,95	90,00	95,00	86,96	88,24	95,00	85,71	58,14	83,33	80,95	78,57
Maximum	100,00	100,00	96,00	95,24	100,00	95,00	100,00	100,00	100,00	100,00	81,40	100,00	95,24	85,71

Tabelle A.2: Klassifikationsraten der Ereignisse je Kanal für die Testdaten (DVD 3)

A.4 *ML*- und *MMI*-Parameterschätzung

Die *Likelihood* eines Merkmalsvektors $\boldsymbol{x}$ für die i-te Klasse ($\Omega = i$) ist mit

$$p(\boldsymbol{x}|\Omega = i) = \sum_{m=1}^{3} c_{i,m} \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{i,m}, \boldsymbol{\Sigma}_{i,m}) \qquad i = 1, 2 \tag{A.52}$$

gegeben. Die Modellparameter der Klasse 1 ($\Omega = 1$) sind mit

$$c_{1,1} = \frac{3}{14}; c_{1,2} = \frac{7}{14}; c_{1,3} = \frac{4}{14}; \tag{A.53}$$

$$\boldsymbol{\mu}_{1,1} = \begin{pmatrix} -6 \\ -3 \end{pmatrix}; \boldsymbol{\mu}_{1,2} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}; \boldsymbol{\mu}_{1,3} = \begin{pmatrix} 4 \\ 4 \end{pmatrix}; \tag{A.54}$$

$$\boldsymbol{\Sigma}_{1,1} = \begin{pmatrix} 1{,}0 & 0{,}0 \\ 0{,}0 & 1{,}0 \end{pmatrix}; \boldsymbol{\Sigma}_{1,2} = \begin{pmatrix} 1{,}8 & 1{,}6 \\ 1{,}6 & 1{,}8 \end{pmatrix}; \boldsymbol{\Sigma}_{1,3} = \begin{pmatrix} 1{,}6 & 0{,}0 \\ 0{,}0 & 1{,}6 \end{pmatrix} \tag{A.55}$$

und die der Klasse 2 ($\Omega = 2$) mit

$$c_{2,1} = \frac{4}{14}; c_{2,2} = \frac{6}{14}; c_{2,3} = \frac{4}{14}; \tag{A.56}$$

$$\boldsymbol{\mu}_{2,1} = \begin{pmatrix} 3 \\ -2 \end{pmatrix}; \boldsymbol{\mu}_{2,2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \boldsymbol{\mu}_{2,3} = \begin{pmatrix} -4 \\ 4 \end{pmatrix}; \tag{A.57}$$

$$\boldsymbol{\Sigma}_{2,1} = \begin{pmatrix} 0{,}1 & 0{,}0 \\ 0{,}0 & 1{,}0 \end{pmatrix}; \boldsymbol{\Sigma}_{2,2} = \begin{pmatrix} 1{,}8 & 1{,}6 \\ 1{,}6 & 1{,}8 \end{pmatrix}; \boldsymbol{\Sigma}_{2,3} = \begin{pmatrix} 1{,}8 & 1{,}6 \\ 1{,}6 & 1{,}8 \end{pmatrix} \tag{A.58}$$

gegeben. Für die Parameter der *GMM* lieferten die Schätzverfahren die folgenden Werte:

- *ML*-Parameterschätzung (volle Kovarianzmatrizen)

$$c_{1,1} = 0{,}21; c_{1,2} = 0{,}50; c_{1,3} = 0{,}28; \tag{A.59}$$

$$\boldsymbol{\mu}_{1,1} = \begin{pmatrix} -6{,}00 \\ -3{,}00 \end{pmatrix}; \boldsymbol{\mu}_{1,2} = \begin{pmatrix} -0{,}98 \\ 0{,}01 \end{pmatrix}; \boldsymbol{\mu}_{1,3} = \begin{pmatrix} 4{,}01 \\ 4{,}01 \end{pmatrix}; \tag{A.60}$$

$$\boldsymbol{\Sigma}_{1,1} = \begin{pmatrix} 0{,}98 & -0{,}02 \\ -0{,}02 & 0{,}97 \end{pmatrix}; \boldsymbol{\Sigma}_{1,2} = \begin{pmatrix} 1{,}86 & 1{,}66 \\ 1{,}66 & 1{,}85 \end{pmatrix}; \boldsymbol{\Sigma}_{1,3} = \begin{pmatrix} 1{,}62 & 0{,}02 \\ 0{,}02 & 1{,}62 \end{pmatrix}; \tag{A.61}$$

$$c_{2,1} = 0{,}43; c_{2,2} = 0{,}29; c_{2,3} = 0{,}29; \tag{A.62}$$

$$\boldsymbol{\mu}_{2,1} = \begin{pmatrix} 1{,}01 \\ 0{,}00 \end{pmatrix}; \boldsymbol{\mu}_{2,2} = \begin{pmatrix} 3{,}01 \\ -1{,}97 \end{pmatrix}; \boldsymbol{\mu}_{2,3} = \begin{pmatrix} -3{,}96 \\ 4{,}03 \end{pmatrix}; \tag{A.63}$$

$$\boldsymbol{\Sigma}_{2,1} = \begin{pmatrix} 1{,}77 & 1{,}58 \\ 1{,}58 & 1{,}79 \end{pmatrix}; \boldsymbol{\Sigma}_{2,2} = \begin{pmatrix} 0{,}10 & 0{,}01 \\ 0{,}01 & 1{,}00 \end{pmatrix}; \boldsymbol{\Sigma}_{2,3} = \begin{pmatrix} 1{,}86 & 1{,}67 \\ 1{,}67 & 1{,}86 \end{pmatrix} \tag{A.64}$$

- *ML*-Parameterschätzung (diagonale Kovarianzmatrizen)

$$c_{1,1} = 0{,}27; c_{1,2} = 0{,}42; c_{1,3} = 0{,}32; \tag{A.65}$$

$$\boldsymbol{\mu}_{1,1} = \begin{pmatrix} -5{,}50 \\ -2{,}81 \end{pmatrix}; \boldsymbol{\mu}_{1,2} = \begin{pmatrix} -0{,}96 \\ 0{,}04 \end{pmatrix}; \boldsymbol{\mu}_{1,3} = \begin{pmatrix} 3{,}72 \\ 3{,}84 \end{pmatrix}; \tag{A.66}$$

$$\boldsymbol{\Sigma}_{1,1} = \begin{pmatrix} 2{,}08 & 0{,}00 \\ 0{,}00 & 0{,}99 \end{pmatrix}; \boldsymbol{\Sigma}_{1,2} = \begin{pmatrix} 0{,}99 & 0{,}00 \\ 0{,}00 & 1{,}00 \end{pmatrix}; \boldsymbol{\Sigma}_{1,3} = \begin{pmatrix} 2{,}08 & 0{,}00 \\ 0{,}00 & 1{,}72 \end{pmatrix} \tag{A.67}$$

$$c_{2,1} = 0{,}48; c_{2,2} = 0{,}24; c_{2,3} = 0{,}29; \tag{A.68}$$

$$\boldsymbol{\mu}_{2,1} = \begin{pmatrix} 1{,}21 \\ -0{,}10 \end{pmatrix}; \boldsymbol{\mu}_{2,2} = \begin{pmatrix} 3{,}01 \\ -2{,}19 \end{pmatrix}; \boldsymbol{\mu}_{2,3} = \begin{pmatrix} -3{,}98 \\ 4{,}02 \end{pmatrix}; \tag{A.69}$$

$$\boldsymbol{\Sigma}_{2,1} = \begin{pmatrix} 1{,}94 & 0{,}00 \\ 0{,}00 & 1{,}81 \end{pmatrix}; \boldsymbol{\Sigma}_{2,2} = \begin{pmatrix} 0{,}08 & 0{,}00 \\ 0{,}00 & 0{,}87 \end{pmatrix}; \boldsymbol{\Sigma}_{2,3} = \begin{pmatrix} 1{,}77 & 0{,}00 \\ 0{,}00 & 1{,}79 \end{pmatrix} \tag{A.70}$$

- *MMI*-Parameterschätzung (diagonale Kovarianzmatrizen)

$$c_{1,1} = 0{,}15; c_{1,2} = 0{,}55; c_{1,3} = 0{,}30; \tag{A.71}$$

$$\boldsymbol{\mu}_{1,1} = \begin{pmatrix} -4{,}55 \\ -2{,}46 \end{pmatrix}; \boldsymbol{\mu}_{1,2} = \begin{pmatrix} -0{,}76 \\ 0{,}07 \end{pmatrix}; \boldsymbol{\mu}_{1,3} = \begin{pmatrix} 2{,}86 \\ 2{,}76 \end{pmatrix}; \tag{A.72}$$

$$\boldsymbol{\Sigma}_{1,1} = \begin{pmatrix} 3{,}64 & 0{,}00 \\ 0{,}00 & 1{,}57 \end{pmatrix}; \boldsymbol{\Sigma}_{1,2} = \begin{pmatrix} 0{,}93 & 0{,}00 \\ 0{,}00 & 0{,}90 \end{pmatrix}; \boldsymbol{\Sigma}_{1,3} = \begin{pmatrix} 2{,}04 & 0{,}00 \\ 0{,}00 & 1{,}60 \end{pmatrix} \tag{A.73}$$

$$c_{2,1} = 0{,}66; c_{2,2} = 0{,}25; c_{2,3} = 0{,}09; \tag{A.74}$$

$$\boldsymbol{\mu}_{2,1} = \begin{pmatrix} 0{,}41 \\ -0{,}58 \end{pmatrix}; \boldsymbol{\mu}_{2,2} = \begin{pmatrix} 2{,}85 \\ 1{,}84 \end{pmatrix}; \boldsymbol{\mu}_{2,3} = \begin{pmatrix} -2{,}07 \\ -0{,}29 \end{pmatrix}; \tag{A.75}$$

$$\boldsymbol{\Sigma}_{2,1} = \begin{pmatrix} 1{,}35 & 0{,}00 \\ 0{,}00 & 1{,}01 \end{pmatrix}; \boldsymbol{\Sigma}_{2,2} = \begin{pmatrix} 0{,}83 & 0{,}00 \\ 0{,}00 & 0{,}84 \end{pmatrix}; \boldsymbol{\Sigma}_{2,3} = \begin{pmatrix} 9{,}32 & 0{,}00 \\ 0{,}00 & 9{,}24 \end{pmatrix} \tag{A.76}$$

Abkürzungsverzeichnis

ACS	***A**mbient **C**ommunication **S**ervice*
AD/DA	**A**nalog-**D**igital/**D**igital-**A**nalog
ADSL	***A**symmetric **D**igital **S**ubscriber **L**ine*
AEC	***A**daptive **E**cho **C**anceler*
AFE	***A**dvanced Front-end **F**eature **E**xtraction*
AI	**A**mbiente **I**ntelligenz
AMI	***A**ugmented **M**ulti-Party **I**nteraction*
ASA	**A**kustische **S**zenen**a**nalyse
BIC	***B**ayesian **I**nformation **C**riterion*
CHIL	***C**omputer in the **H**uman **I**nteraction **L**oop*
CMS	***C**ontext **M**anagement **S**ervice*
CTM	***C**lose **T**alking **M**icrophone*
DARPA	***D**efense **A**dvanced **R**esearch **P**rojects **A**gency*
DCT	**D**iskrete **C**osinus **T**ransformation
DER	***D**iarization **E**rror **R**ate*
DFT	**D**iskrete **F**ourier **T**ransformation
DIRAC	***D**etection and **I**dentification of **R**are **A**udiovisual **C**ues*
DNS	***D**omain **N**ame **S**ystem*
DSB	***D**elay **S**um **B**eamformer*
DTM	***D**istant **T**alking **M**icrophone*
EARS	***E**ffective, **A**ffordable, **R**eusable **S**peech-to-Text*
EER	***E**qual **E**rror **R**ate*
EIB	***E**uropean **I**nstallation **B**us*
ELDA	***E**valuations and **L**anguage resources **D**istribution **A**gency*
EM	***E**xpectation **M**aximization*
ETSI	***E**uropean **T**elecommunications **S**tandards **I**nstitute*
FAR	***F**alse **A**larm **R**ate*
FFT	***F**ast **F**ourier **T**ransformation*
FIR	***F**inite **I**mpulse **R**esponse*
FSB	***F**ilter **S**um **B**eamformer*
GCC	***G**eneralized **C**ross **C**orrelation*
GCC-PHAT	***G**eneralized **C**ross **C**orrelation* with **Pha**se **T**ransformation
GCF	***G**lobal **C**oherence **F**ield*
GMM	***G**aussian **M**ixture **M**odel*
GUI	***G**raphical **U**ser **I**nterface*
HMM	***H**idden **M**arkov **M**odel*
HTTP	***H**yper**t**ext **T**ransfer **P**rotocol*

HW/SW ..... **H**ard**w**are/**S**oft**w**are-Schnittstelle
IDCT ..... **I**nverse **D**iskrete **C**osinus **T**ransformation
IDFT ..... **I**nverse **D**iskrete **F**ourier **T**ransformation
IP ..... ***I**nternet **P**rotocol*
IPC ..... ***I**nter **P**rocess **C**ommunication*
IST ..... ***I**nformation **S**ociety **T**echnologies*
ITU ..... ***I**nternational **T**elecommunication **U**nit*
JACK ..... ***J**ack **A**udio **C**onnection **K**it*
LDA ..... **L**ineare **D**iskriminanz**a**nalyse
LDAP ..... ***L**ightweight **D**irectory **A**ccess **P**rotocol*
LMS ..... ***L**ocation **M**anagement **S**ervice*
LPCC ..... ***L**inear **P**rediction **C**epstral **C**oefficients*
LST ..... **L**okale **S**trukturtransformation
MACV ..... ***M**aximum **A**uto**c**orrelation **V**alue*
MAP ..... **M**aximum **A** **P**osteriori
MCE ..... ***M**inimum **C**lassification **E**rror*
MDR ..... ***M**issed **D**etection **R**ate*
MFCC ..... ***M**el-**F**requency **C**epstral **C**oefficients*
ML ..... ***M**aximum **L**ikelihood*
MMI ..... ***M**aximum **M**utual **I**nformation*
NAT ..... ***N**etwork **A**ddress **T**ranslation*
NIST ..... ***N**ational **I**nstitute of **S**tandards and **T**echnologies*
NLMS ..... ***N**ormalized **L**east **M**ean **S**quare*
NSD ..... ***N**ear **S**peaker **D**etector*
OSGI ..... ***O**pen **S**ervices **G**ateway **I**nitiative*
OWL ..... ***W**eb **O**ntology **L**anguage*
OWL-S ..... ***W**eb **O**ntology **L**anguage for Web **S**ervices*
PCA ..... ***P**rinciple **C**omponent **A**nalysis*
PDA ..... ***P**ersonal **D**igital **A**ssistent*
PLC ..... ***P**acket **L**oss **C**oncealment*
PTZ ..... ***P**an **T**ilt **Z**oom*
QoS ..... ***Q**uality **o**f **S**ervice*
RDF ..... ***R**esource **D**escription **F**ramework*
RFC ..... ***R**equests **F**or **C**omments*
RFID ..... ***R**adio **F**requency **Id**entification*
RMI ..... ***R**emote **M**ethod **I**nvocation*
RMS ..... ***R**oot **M**ean **S**quare*
ROC ..... ***R**eceiver **O**perating **C**haracteristic*
RPC ..... ***R**emote **P**rocedure **C**all*
RTP ..... ***R**eal-Time **T**ransport **P**rotocol*
SAInt ..... ***S**eamless **A**udio **Int**erface*
SAVInt ..... ***S**eamless **A**udio and **V**ideo **Int**erface*
SDI ..... ***S**ervice Discovery Protocol - **D**etection and **I**nteroperability*
SDP ..... ***S**ervice **D**iscovery **P**rotocol*
SER ..... ***S**ignal to **E**cho **R**atio*
SHM ..... ***Sh**ared **M**emory*

SII ..... ***S**ervice **I**nteraction **I**nteroperability*
SIP ..... ***S**ession **I**nitialization **P**rotocol*
SLP ..... ***S**ervice **L**ocation **P**rotocol*
SNR ..... ***S**ignal to **N**oise **R**atio*
SOAP ..... ***S**imple **O**bject **A**ccess **P**rotocol*
Spark ..... ***S**peech **p**rocessing **a**nd **r**ecognition tool**k**it*
SPARQL ..... ***S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage*
SSDP ..... ***S**imple **S**ervice **D**iscovery **P**rotocol*
STUN ..... ***S**imple **T**raversal of **U**ser Datagram Protocol Through **N**etwork Address Translators*
TCP ..... ***T**ransmission **C**ontrol **P**rotocol*
UBM ..... ***U**niversal **B**ackground **M**odel*
UDP ..... ***U**niversal **D**atagram **P**rotocol*
UPnP ..... ***U**niversal **P**lug a**n**d **P**lay*
URI ..... ***U**niform **R**esource **I**dentifier*
URN ..... ***U**niform **R**esource **N**ame*
VAD ..... ***V**oice **A**ctivity **D**etection*
VoIP ..... ***V**oice **o**ver **I**nternet **P**rotocol*
WSDL ..... ***W**eb **S**ervices **D**escription **L**anguage*
WWW ..... ***W**orld **W**ide **W**eb*
XAFE ..... *E**x**tended **A**dvanced Front-end **F**eature **E**xtraction*
XML ..... *E**x**tensible **M**arkup **L**anguage*

Formelzeichen

Akustische Szenenanalyse - Merkmalsextraktion

$\mathit{MACV}(q)$	q-ter Wert des *MACV*-Merkmalsvektors
$r(k)$	Normierte Autokorrelationsfunktion
$R(k)$	Autokorrelationsfunktion
$\tilde{x}(n)$	n-ter Abtastwert des gefensterten Mikrophonsignals

Akustische Szenenanalyse - Positionsschätzung

$\boldsymbol{a}_l(\overline{\alpha}_l, \beta_l)$	Richtungsvektor aus Winkelschätzung und Mikrophongruppenorientierung
c	Schallgeschwindigkeit in der Luft
$C_{ij,l}^{(GCC)}(\tau), C_{ij,l}^{(FSB)}(\tau)$	Interpolierte Fourier-Rücktransformierte der Kohärenzfunktion zwischen den Mikrophonen i und j der l-ten Mikrophongruppe, Schätzung durch *GCC* bzw. *FSB*
$f_i(n)$	n-ter *FSB*-Filterwert des i-ten Filters
$f_{\max}$	Maximale ohne Aliasingfehler auflösbare Frequenz
$F_i(k)$	k-tes *FSB*-Filterbin des i-ten Filters
$\boldsymbol{F}(k)$	Vektor der *FSB*-Filterbins
$\boldsymbol{g}_l$	Richtungsvektor der l-ten Mikrophongruppe
$\boldsymbol{G}$	Gitter der globalen Kohärenzfeldanalyse
$\mathit{GCF}(x, y)$	Globale Kohärenzfunktion am Ort $[x, y]$
$h_i(n)$	n-ter Abtastwert der Raumimpulsantwort zum i-ten Mikrophon
L	Anzahl der Mikrophongruppen
M_l	Anzahl der Mikrophone der l-ten Mikrophongruppe
$n_i(n)$	n-ter Abtastwert der Störung am i-ten Mikrophon
$\boldsymbol{P} = [x_p, y_p]^T$	Positionsschätzung in kartesischen Koordinaten
$\boldsymbol{r}_l$	Positionsvektor der l-ten Mikrophongruppe
$s(n)$	n-ter Abtastwert des Sprachsignals
$s_{ij,l}$	Abstand zwischen den Mikrophonen i und j der l-ten Mikrophongruppe
T	Abtastperiode
$[x, y]$	Gitterpunkt der globalen Kohärenzfeldanalyse
$x_i(n)$	n-ter Abtastwert des i-ten Mikrophonsignals
$x_{i,l}(n)$	n-ter Abtastwert des i-ten Mikrophons der l-ten Mikrophongruppe
$X_i(k)$	k-ter Frequenzbin des i-ten Mikrophonsignals
$\boldsymbol{X}(k)$	Vektor der Frequenzbins der Mikrophonsignale
$y(n)$	n-ter Abtastwert des *FSB*-Ausgangssignals
$Y(k)$	k-tes Frequenzbin des *FSB*-Ausgangssignals

$\alpha_{ij,l}$ ..... Schätzung des Einfallswinkels basierend auf den Mikrophonen i und j der l-ten Mikrophongruppe

$\overline{\alpha}_l$ ..... Gemittelte Schätzung des Einfallswinkels für die l-te Mikrophongruppe

β_l ..... Orientierung der l-ten Mikrophongruppe

$\gamma_{ij,l}$ ..... Gewichtsfaktor zur Positionsschätzung

$\lambda_{ij,l}^{(\max)}$ ..... Maximale Laufzeitdifferenz zwischen Mikrophon i und j der l-ten Mikrophongruppe

μ ..... Schrittweite des *FSB*

$\tau_{ij,l}^{(GCC)}, \tau_{ij,l}^{(FSB)}$ ..... Laufzeitdifferenz der Signale zwischen Mikrophon i und j der l-ten Mikrophongruppe, Schätzung durch *GCC* bzw. *FSB*

$\phi_{ij,l}^{(GCC)}(\tau), \phi_{ij,l}^{(FSB)}(\tau)$ .. Fourier-Rücktransformierte der Kohärenzfunktion zwischen den Mikrophonen i und j der l-ten Mikrophongruppe, Schätzung durch *GCC* bzw. *FSB*

$\mathbf{\Phi}_{xx}$ ..... Spektrale Kreuzleistungsdichtematrix der Mikrophonsignale

$\boldsymbol{\chi}_{ij}$ ..... Schnittpunkt zwischen der i-ten und j-ten Geraden

Akustische Szenenanalyse - Sprecherprotokollierung

$a_{ij}()$ ..... Transitionswahrscheinlichkeit von Zustand i auf Zustand j

$\tilde{a}_{ij}()$ ..... Nicht normierte Transitionswahrscheinlichkeit von Zustand i auf j

$\boldsymbol{A}$ ..... Transitionsmatrix des *HMM*

$\boldsymbol{B}$ ..... Menge der Verteilungsdichtefunktionen der *HMM*-Zustände

$b_j()$ ..... Emissionswahrscheinlichkeit des j-ten Zustandes

$c(k)$ ..... Binäre Zufallsvariable, die einen Sprecherwechsel anzeigt

$c_{i,m}, \tilde{c}_{i,m}$ ..... i-tes Gewicht der m-ten Gauß'schen Mischungsverteilung

D ..... Dimension der Merkmalsvektoren

H_0, H_1 ..... *BIC*-Hypothesen

$\boldsymbol{L}$ ..... Transformationsmatrix der linearen Diskriminanzanalyse

m_i ..... Anzahl der Modellparameter im i-ten Modell

$\boldsymbol{m}_{LDA}$ ..... Mittelwertvektor der linearen Diskriminanzanalyse

$\boldsymbol{m}_{PCA}$ ..... Mittelwertvektor der Hauptachsentransformation

N ..... Anzahl der Merkmalsvektoren des Segmentes

N_w ..... Anzahl der Merkmalsvektoren im Fenster

$p()$ ..... Verteilungsdichtefunktion einer kontinuierlichen Zufallsvariablen

$P()$ ..... Verteilungsdichtefunktion einer diskreten Zufallsvariablen

$P(S|x^{sid})$ ..... Wahrscheinlichkeit für Sprache

$\boldsymbol{P}$ ..... Transformationsmatrix der Hauptachsentransformation

$\boldsymbol{P}(k)$ ..... Positionsschätzung in kartesischen Koordinaten zum Zeitpunkt k

r ..... Relevanzfaktor der Bayes'schen Adaption

$x^{bic}(k)$ ..... Varianz der Δ*BIC*-Werte zum Zeitpunkt k

$x^{pos}(k)$ ..... Varianz der Position zum Zeitpunkt k

$\boldsymbol{x}(k)$ ..... Merkmalsvektor

$\boldsymbol{x}^{sid}(k)$ ..... Akustischer Merkmalsvektor zum Zeitpunkt k bestehend aus *MFCC* und *MACV*, sowie erster und zweiter Ableitung

$\boldsymbol{x}_M^{sid}(k)$ ..... *MFCC*-Merkmalsvektor und *MACV*-Wert

$\boldsymbol{x}^{sid}_{\Delta M}(k)$ ..... 1. Ableitung der *MFCC*- und *MACV*-Werte
$\boldsymbol{x}^{sid}_{\Delta\Delta M}(k)$ ..... 2. Ableitung der *MFCC*- und *MACV*-Werte
$\boldsymbol{x}^{vid}(k)$ ..... Visueller Merkmalsvektor zum Zeitpunkt k
$\boldsymbol{x}^{vid}_{\nu:k}$ ..... Folge der visuellen Merkmalsvektoren vom Zeitpunkt $k-\nu+1$ bis zum Zeitpunkt k
$\boldsymbol{X}_{1:N_w}$ ..... Menge von N_w Merkmalsvektoren
Z ..... Zufallsvariable der Mischungsverteilungszugehörigkeit
α, β ..... Glättungsparameter
$\gamma_{acc}, \gamma_{delta}$ ..... Gewichtungsfaktoren der *Score Level Fusion*
$\boldsymbol{\Gamma}(k)$ ..... Zeilenweise ausgelesenes Teilbild
$\Delta BIC(k)$ ..... Δ*BIC*-Wert eines um den Zeitpunkt k zentrierten Fensters
$k_{\max}$ ..... Zeitpunkt eines lokalen Maximums der Δ*BIC*-Werte
$k_{\min_L}, k_{\min_R}$ ..... Linkes bzw. rechtes lokales Minimum der Δ*BIC*-Werte
ϵ_i ..... Adaptionskoeffizient
κ ..... Heuristischer Gewichtungsfaktor
λ ..... Schwellwert der Δ*BIC*-Segmentierung
λ_{UBM} ..... Universelles Hintergrundmodell
$\lambda^{F}_{UBM}, \lambda^{M}_{UBM}$ ..... Universelles Hintergrundmodell für Frauen, Männer
$\Lambda(), \widehat{\Lambda}()$ ..... *Likelihood*-Verhältnis
$\boldsymbol{\mu}_{i,m}, \boldsymbol{\mu}_i, \tilde{\boldsymbol{\mu}}_{i,m}$ ..... Mittelwertvektoren
$\mu^{bic}(k)$ ..... Mittelwert der Δ*BIC*-Werte zum Zeitpunkt k
$\mu^{pos}(k)$ ..... Mittelwert der Position zum Zeitpunkt k
π_i ..... A priori Wahrscheinlichkeit des i-ten *HMM*-Zustandes
σ ..... Standardabweichung der Δ*BIC*-Werte
$\boldsymbol{\Sigma}_{i,m}, \tilde{\boldsymbol{\Sigma}}_i, \boldsymbol{\Sigma}_i$ ..... Kovarianzmatrizen
$\boldsymbol{\Theta}_i, \tilde{\boldsymbol{\Theta}}_i$ ..... Modellparameter der i-ten Gauß'schen Mischungsverteilung
τ_{avg} ..... Mittlere Verzögerung der Sprecherprotokollierung
$\tau_{\max}$ ..... Maximale Verzögerung des Viterbi-Dekodierers
ξ ..... Konstante
Ω ..... Zufallsvariable der Klassenzugehörigkeit
$\widehat{\Omega}$ ..... Sprecherhypothese
$\widehat{\boldsymbol{\Omega}}_{1:N}$ ..... Zustandssequenz über N Zustände im Trellisdiagramm
$\mathcal{I}$ ..... Anzahl trainierter Benutzer
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ ..... Normalverteilung mit Mittelwertvektor $\boldsymbol{\mu}$ und Kovarianzmatrix $\boldsymbol{\Sigma}$

Akustische Ereignisdetektion

$c_{k,m}, \widehat{c}_{k,m}$ ..... Gewicht der m-ten Mischungsverteilung der k-ten Klasse
D ..... Dimension der Merkmalsvektoren
K ..... Anzahl der Klassen
M ..... Anzahl der Mischungsverteilungen je Klasse
N_k ..... Anzahl der Merkmalsvektoren in der k-ten Klasse
$Q(\Theta)$ ..... Zielfunktion
r ..... Relevanzfaktor
$\boldsymbol{x}_k(n)$ ..... n-ter Merkmalsvektor der k-ten Klasse
$\boldsymbol{X}^{m}_{1:N}$ ..... Menge von N Merkmalsvektoren vom m-ten Mikrophon

Z ..... Zufallsvariable der Zugehörigkeit zu einer Mischungsverteilung
$\gamma_{k,m}(n)$ ..... Wahrscheinlichkeit der m-ten Mischungsverteilung der k-ten Klasse gegeben den n-ten Merkmalsvektor
λ ..... Lagrange-Operator
$\boldsymbol{\mu}_{k,m}, \widehat{\boldsymbol{\mu}}_{k,m}$ ..... Mittelwertvektor der m-ten Mischungsverteilung der k-ten Klasse
$\psi_{m,k}(n)$ ..... Gewichtsfaktor der m-ten Mischungsverteilung der k-ten Klasse gegeben den n-ten Merkmalsvektor
$\boldsymbol{\Sigma}_{k,m}, \widehat{\boldsymbol{\Sigma}}_{k,m}$ ..... Kovarianzmatrix der m-ten Mischungsverteilung der k-ten Klasse
$\boldsymbol{\Theta}_k$ ..... Modellparameter der k-ten Klasse
Ω ..... Zufallsvariable der Klassenzugehörigkeit
$\widehat{\Omega}$ ..... Schätzwert für das akustische Ereignis

Ambiente Kommunikation

$b(n), B(m,\omega)$ ..... Zeitsignal und Frequenzspektrum des Restechos
$e(n), E(m,\omega)$ ..... Zeitsignal und Frequenzspektrum des Fehlersignals
$F(m,\omega), \widetilde{F}(m,\omega)$ ... Filterfunktion
$h(n), \boldsymbol{h}$ ..... Raumimpulsantwort
H_0, H_1, H_2, H_3 ..... Hypothesen über vorliegende Signale
N ..... Filterlänge
$r(n), R(m,\omega)$ ..... Zeitsignal und Frequenzspektrum der lokalen Störungen
$\widehat{R}_n(m,\omega)$ ..... Schätzung des Leistungsdichtespektrums des lokalen Rauschens
$\widehat{R}_b(m,\omega)$ ..... Schätzungen des Leistungsdichtespektrums des Restechos
$s(n), S(m,\omega)$ ..... Zeitsignal und Frequenzspektrum des lokalen Sprechers
$\boldsymbol{x}(n)$ ..... n-ter Block des Eingangssignals
$w(n)$ ..... Adaptives Filter des *AEC*
$x_p(n)$ ..... Geglättete Spitzenwert der Energie eines Blocks
$y(n)$ ..... Mikrophonsignal
$\alpha_\xi, \alpha_\zeta, \beta$ ..... Glättungsparameter
β_ξ, β_ζ ..... Parameter zur Steuerung der Störungs- und Restechounterdrückung
$\Gamma(n)$ ..... Gewichtsfaktor für den n-ten Block des Eingangssignals
γ_T ..... Schwellwert des Begrenzers
$\zeta(m,\omega)$ ..... A priori *SER*
$\mu(n)$ ..... Schrittweite des adaptiven Filters
$\xi(m,\omega)$ ..... A priori *SNR*
ξ ..... Entscheidungsvariable *NSD*
σ_s^2 ..... Varianz des lokalen Sprechersignals
$\sigma()$ ..... Einheitssprungfunktion
τ_A, τ_R ..... Anstiegszeit, Abfallzeit der *NSD*
$\boldsymbol{\phi}_{xx}$ ..... Autokorrelationsmatrix des Eingangssignals

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Aar09] E. Aarts: „Ambient Intelligence: A new user experience“, 2009, [URL] http://www.research.philips.com/technologies/projects/ami/vision.html.

[AB79] J. B. Allen und D. A. Berkley: „Image method for efficiently simulating small-room acoustics“, *Journal of the Acoustic Society of America*, Band 65(4), S. 943–950, Apr. 1979.

[AFI+08] S. Araki, M. Fujimoto, K. Ishizuka, H. Sawada und S. Makino: „A DOA Based Speaker Diarization System for Real Meetings“, *Proc. Conference on Hands-Free Speech Communication and Microphone Arrays (HSCMA'08)*, S. 29–32, Trient, Italien, Mai 2008.

[AM04] E. Aarts und S. Marzano: *The New Everyday: Views on Ambient Intelligence*, 010 Uitgeverij, Rotterdam, Niederlande, 2004.

[AMI04] AMI: „Augmented Multi-Party Interaction“, Jan. 2004, [URL] http://www.amiproject.org/.

[Ami06] Amigo: „Ambient intelligence for the networked home environment“, 2006, [URL] http://www.hitech-projects.com/euprojects/amigo.

[APW06] X. Anguerra, J. Pardo und C. Wooters: „Speaker Diarization for Multiple Distant Microphone Meetings: Mixing Acoustic Features and Inter-Channel Time Differences“, *Proc. Conference of the International Speech Communication Association (Interspeech'06)*, S. 2194–2197, Pittsburgh PA, USA, Sep. 2006.

[AWH07] X. Anguera, C. Wooters und J. Hernando: „Acoustic Beamforming for Speaker Diarization of Meetings“, *IEEE Transactions on Audio, Speech and Language Processing*, Band 15(7), S. 2011–2022, Sep. 2007.

[B+01] D. Brickley *et al.*: „Semantic Web“, 2001, [URL] http://www.w3.org/2001/sw/.

[B+05a] T. Berners-Lee *et al.*: „Uniform Resource Identifier“, Jan. 2005, [URL] http://tools.ietf.org/html/rfc3986.

[B+05b] C. Busso *et al.*: „Smart Room: Participant and Speaker Localization and Identification“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, S. 1117–1120, Philadelphia PA, USA, Mär. 2005.

[B+08a] D. Beckett *et al.*: „Resource Description Framework“, Jan. 2008, [URL] http://www.w3.org/RDF/.

[B+08b] T. Bray *et al.*: „Extensible Markup Language“, Nov. 2008, [URL] http://www.w3.org/XML/.

[BFGP08] S. Borkowski, T. Flury, A. Gerodolle und G. Privat: „Ambient Communication and Context-Aware Presence Management“, *Communications in Computer and Information Science*, Band 11, S. 391–396, 2008.

[BH03] J. Benesty und S. Huang: *Adaptive Signal Processing: Applications to Real-World Problems*, Springer Verlag, Heidelberg, Deutschland, 2003.

[BHK97] P. Belhumeur, J. Hespanha und D. Kriegman: „Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Band 19(7), S. 711–720, Jul. 1997.

[BHL01] T. Berners-Lee, J. Hendler und O. Lassila: „The Semantic Web“, *Scientific American Magazine*, S. 1–4, Mai 2001.

[BI05] Y. Bromberg und V. Issarny: „INDISS: Interoperable Discovery System for Networked Services“, *Lecture Notes in Computer Science*, Band 3790, S. 164–183, Dez. 2005.

[BMC00] J. Benesty, R. Morgen und J. Cho: „A New Class of Doubletalk Detectors Based on Cross-Correlation“, *IEEE Transactions on Speech and Audio Processing*, Band 8(2), S. 168–172, Mär. 2000.

[BP08] C. Boukis und L. C. Polymenakos: „The Acoustic Event Detector of AIT“, *Lecture Notes in Computer Science: Multimodal Technologies for Perception of Humans*, Band 4625, S. 328–337, 2008.

[BS07] K. Bernardin und R. Stiefelhagen: „Audio-visual multi-person tracking and identification for smart environments“, *International Conference on Multimedia (MM'07)*, S. 661–670, Augsburg, Deutschland, Sep. 2007.

[BSMM01] I. Bronstein, K. Semendjajew, G. Musiol und H. Mühlig: *Taschenbuch der Mathematik*, Verlag Harri Deutsch, Frankfurt am Main, Deutschland, 2001.

[C+07] R. Chinnici *et al.*: „Web Services Description Language“, Jun. 2007, [URL] http://www.w3.org/TR/wsdl20/.

[Cam97] J. Campbell: „Speaker Recognition: A Tutorial“, *Proceedings of the IEEE*, Band 85(9), S. 1437–1462, Sep. 1997.

[Car08] M. J. Carey: „SOA What?“, *IEEE Computer*, Band 41(3), S. 92–94, Mär. 2008.

[CG98] S. S. Chen und P. S. Gopalakrishnan: „Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion“, *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, USA, Feb. 1998.

[CHI04] CHIL: „Computers in the Human Interaction Loop“, Jan. 2004, [URL] http://chil.server.de/.

[CSJ07] X. Chen, Y. Shi und W. Jiang: „Speaker Tracking and Identifying based on Indoor Localization Systems and Microphone Array“, *International Conference on Advanced Information Networking and Applications (AINA'07)*, S. 347–352, Niagarafälle, Kanada, Mai 2007.

[CW03] S. Cheng und H. Wang: „A Sequential Metric-based Audio Segmentation Method via the Bayesian Information Criterion“, *Proc. Eurospeech*, S. 945–948, Genf, Schweiz, Sep. 2003.

[DAM06] DAML: „Web Ontology Language for Web Services“, 2006, [URL] http://www.daml.org/services/owl-s/.

[DBA07] J. Dmochowski, J. Benesty und S. Affes: „A Generalized Steered Response Power Method for Computationally Viable Source Localization“, *IEEE Transactions on Speech and Audio Processing*, Band 15(8), S. 2510–2526, Nov. 2007.

[DES99] R. Derkx, G. Egelmeers und P. Sommen: „New Constraining Method for Partioned Block Frequency-Domain Adaptive Filters“, *IEEE Transactions on Signal Processing*, Band 50(9), S. 2177–2186, Sep. 1999.

[DHS01] R. Duda, P. Hart und D. Stork: *Pattern Classification - Second Edition*, John Wiley & Sons, Kanada, 2001.

[DIR06] DIRAC: „Detection and Identification of Rare Audiovisual Cues“, Jan. 2006, [URL] http://www.diracproject.org/.

[DW00] P. Delacourt und C. J. Wellekens: „DISTBIC: A speaker-based segmentation for audio data indexing“, *Speech Communications*, Band 32(1-2), S. 111–126, Sep. 2000.

[DY08] N. Dhananjaya und B. Yegnanarayana: „Speaker change detection in casual conversations using excitation source features“, *Speech Communications*, Band 50(2), S. 153–161, Feb. 2008.

[ECB06] E. Etter, P. D. Costa und T. Broens: „A Rule-Based Approach Towards Context-Aware User Notification Services“, *Proc. IEEE International Conference on Pervasive Services (ICPS'06)*, S. 281–284, Lyon, Frankreich, Jun. 2006.

[EFJS07] H. K. Ekenel, M. Fischer, Q. Jin und R. Stiefelhagen: „Multi-modal Person Identification in a Smart Environment“, *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, S. 1–8, Minneapolis MN, USA, Jun. 2007.

[EIB09] EIB: „European Installation Bus“, 2009, [URL] http://www.knx.org/.

[EK05] J. L. Encarnacao und T. Kirste: „Ambient Intelligence: Towards Smart Appliance Ensembles“, *Lecture Notes in Computer Science: From Human Computer Interaction to Human Artifact Interaction*, Band 3379, S. 261–270, Jan. 2005.

[ELD08] ELDA: Jan. 2008, [URL] http://www.elda.org/.

[ETS02] ETSI: „ES 202 212 V1.1.1: Speech Processing, Transmission and Quality aspects (STQ); Distributed Speech Recognition; Advanced front-end feature extraction algorithm; Compression algorithms“, 2002, [URL] http://www.etsi.org/.

[FCP+05] M. Friedewald, O. Costa, Y. Punie, P. Alahuhta und S. Heinonen: „Perspectives of ambient intelligence in the home environment“, *Telematics and Informatics*, Band 22(3), S. 221 – 238, 2005.

[FHY09] G. Friedland, H. Hung und C. Yeo: „Multi-modal speaker diarization of real-world meetings using compressed-domain video features“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'09)*, Taipei, Taiwan, Apr. 2009.

[FK04] B. Fröba und C. Küblbeck: „Face tracking by Means of Continous Detection“, *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04)*, S. 65–71, Washington D.C., USA, Mär. 2004.

[Fuk90] K. Fukunaga: *Statistical Pattern Recognition - Second Edition*, Academic Press, San Diego CA, USA, 1990.

[G+07] M. Gudgin *et al.*: „W3C Recommendation: SOAP - Simple Object Access Protocol“, Apr. 2007, [URL] http://www.w3.org/TR/soap/.

[GAW06] A. Gallardo-Antolin, X. Anguerra und C. Wooters: „Multi-Stream Speaker Diarization Systems for the Meetings Domain“, *Proc. Conference of the International Speech Communication Association (Interspeech'06)*, S. 2186–2189, Pittsburgh PA, USA, Sep. 2006.

[GB01] B. Gänsler und J. Benesty: „A frequency-domain double-talk detector based on a normalized cross-correlation vector“, *Signal Processing*, Band 81(8), S. 1783–1787, Aug. 2001.

[GDJ06] S. Guha, N. Daswani und R. Jain: „An Experimental Study of the Skype Peer-to-Peer VoIP System“, *Proc. IEEE International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Barbara CA, USA, Feb. 2006.

[GJKV99] S. Gustafsson, P. Jax, A. Kamphausen und P. Vary: „A postfilter for echo and noise reduction avoiding the problem of musical tones“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99)*, S. 873–876, Phoenix AZ, USA, Mär. 1999.

[GMB+05] N. Georgantas, S. B. Mokhtar, Y. Bromberg, V. Issarny, J. Kalaoja, J. Kantarovich, A. Gerodolle und R. Mevissen: „The Amigo Service Architecture for the Open Networked Home Environment“, *Proc. Working IEEE/ IFIP Conference on Software Architecture (WICSA'05)*, Pittsburgh PA, USA, Nov. 2005.

[Gru93] T. R. Gruber: „A Translation Approach to Portable Ontology Specifications“, *Knowledge Acquisition*, Band 5(2), S. 199–220, Jun. 1993.

[Hän01] E. Hänsler: *Statistische Signale - 3.Auflage*, Springer Verlag, Berlin, Deutschland, 2001.

[Här07] A. Härmä: „Ambient Telephony: scenarios and research challenges“, *Proc. Conference of the International Speech Communication Association (Interspeech'07)*, Antwerpen, Belgien, Aug. 2007.

[Hay02] S. Haykin: *Adaptive Filter Theory - Fourth Edition*, Prentice Hall, Upper Saddle River NJ, USA, 2002.

[HD08] X. He und L. Deng: *Discriminative Learning for Speech Recognition: Theory and Prcatice*, Morgan and Claypool, San Rafael CA, USA, 2008.

[HS05] R. Haeb-Umbach und J. Schmalenstroeer: „A Comparison of Particle Filtering Variants for Speech Feature Enhancement“, *Proc. Conference of the International Speech Communication Association (Interspeech'05)*, Lissabon, Portugal, Sep. 2005.

[ISC07] ISC: „Internet Systems Consortium“, Jan. 2007, [URL] http://www.isc.org/.

[ITU01] ITU: „ITU X.500 Specification“, Jan. 2001, [URL] http://www.itu.int/rec/T-REC-X.500/.

[J+05] M. Janse *et al.*: „Amigo Public Deliverable D4.1: Report on Specification and Description of Interfaces and Services“, Nov. 2005, [URL] http://www.hitech-projects.com/euprojects/amigo/deliverables/.

[JAC08] JACK: „Jack Audio Connection Kit“, Jan. 2008, [URL] http://jackaudio.org/.

[KC76] C. Knapp und G. Carter: „The generalized correlation method for estimation of time delay“, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Band 24(4), S. 320–327, Aug. 1976.

[KE06] C. Küblbeck und A. Ernst: „Face Detection and Tracking in Video Sequences Using the Modified Census Transform“, *Image and Video Computing*, Band 24(6), S. 564–572, Jun. 2006.

[KFH+08] V. Khalidov, F. Forbes, M. Hansard, E. Arnaud und R. P. Horaud: „Audio-Visual Clustering for Multiple Speaker Localization“, *Lecture Notes in Computer Science: Machine Learning for Multimodal Interaction*, S. 86–97, Sep. 2008.

[KHF04] T. Kinnunen, V. Hautamäki und P. Fränti: „Fusion of Spectral Feature Sets for Accurate Speaker Identification“, *Proc. Conference on Speech and Computer (SPECOM'2004)*, St. Petersburg, Russland, Sep. 2004.

[KMK07] M. Kotti, V. Moschou und C. Kotropoulos: „Speaker segmentation and clustering“, *Signal Processing*, Band 88(5), S. 1091–1124, Mai 2007.

[KSLK03] C. Kim, S. Seong, J. Lee und L. Kim: „WinScale: An Image-Scaling Algorithm Using an Area Pixel Model“, *IEEE Transactions on Circuits and Systems for Video Technology*, Band 13(6), S. 549–553, Jun. 2003.

[KTVL07] T. Kühnapfel, T. Tan, S. Venkatesh und E. Lehmann: „Calibration of Audio-Video Sensors for Multi-Modal Event Indexing“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'07)*, S. 741–744, Honolulu, Hawaii, USA, Apr. 2007.

[KYM+05] Y. Kida, H. Yamamoto, C. Miyajima, K. Tokuda und T. Kitamura: „Minimum classification error interactive training for speaker identification“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, Philadelphia PA, USA, Mär. 2005.

[LK07] S. Lee und N. Kim: „A Statistical Model-Based Residual Echo Suppression“, *IEEE Signal Processing Letters*, Band 14(10), S. 758–761, Okt. 2007.

[LLJ+08] P. Liu, C. Liu, H. Jiang, F. Soong und R. Wang: „A Constrained Line Search Optimization Method for Discriminative Training of HMMs“, *IEEE Transactions on Audio, Speech and Language Processing*, Band 16(5), S. 900–909, Jul. 2008.

[LP96] C. Lee und K. Paliwal: *Automatic Speech and Speaker Recognition: Advanced Topics*, Kluwer Academic Publishers, London, England, 1996.

[LYL07] J. Li, M. Yuan und C. Lee: „Approximate Test Risk Bound Minimization Through Soft Margin Estimation“, *IEEE Transactions on Audio, Speech and Language Processing*, Band 15(8), S. 2393–2404, Nov. 2007.

[LZ02] L. Lu und H. Zhang: „Real-Time Unsupervised Speaker Change Detection“, *Proc. International Conference on Pattern Recognition (ICPR'02)*, Quebec Stadt, Kanada, Aug. 2002.

[M+97] R. Moats *et al.*: „Uniform Resource Name Syntax“, Mai 1997, [URL] http://tools.ietf.org/html/rfc2141/.

[M+05] C. Margerkurth *et al.*: „Amigo Public Deliverable D1.2: Report on User Requirements“, Feb. 2005, [URL] http://www.hitech-projects.com/euprojects/amigo/deliverables/.

[MC03] C. Ma und E. Chang: „Comparison of Discriminant Training Methods for Speaker Verification“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Orlando FL, USA, Apr. 2003.

[MH00] G. Moschytz und M. Hofbauer: *Adaptive Filter*, Springer Verlag, Heidelberg, Deutschland, 2000.

[MH04] D. L. McGuinness und F. Harmelen: „Web Ontologie Language“, Feb. 2004, [URL] http://www.w3.org/TR/owl-features/.

[MKGI07] S. Mokhtar, A. Kaul, N. Georgantas und V. Issarny: „Efficient Semantic Service Discovery in Pervasive Computing Environments“, *Lecture Notes in Computer Science*, Band 4290, S. 240–259, 2007.

[MMF+06] S. Meignier, D. Moraru, C. Fredouille, J. Bonastre und L. Besacier: „Step-by-Step and Integrated Approaches in Broadcast News Speaker Diarization“, *Computer Speech & Languagee*, Band 20(2-3), S. 303–330, Jul. 2006.

[Mos05] D. Mostefa: „CHIL Speaker ID evaluation“, Jan. 2005, [URL] http://chil.server.de/.

[NCM91] Y. Normandin, R. Cardin und R. Mori: „High-Performance Connected Digit Recognition Using Maximum Mutual Information Estimation“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, S. 533–536, Toronto Ontario, Kanada, Apr. 1991.

[NGN09] NGN: „Next Generation Network“, 2009, [URL] http://de.wikipedia.org/wiki/Next_Generation_Network/.

[NIS08a] NIST: Jan. 2008, [URL] http://www.nist.gov/.

[NIS08b] NIST: „DARPA EARS Rich Transcription Evaluation Project“, Jan. 2008, [URL] http://www.nist.gov/speech/tests/rt/.

[NK05] M. Nishida und T. Kawahara: „Speaker Model Selection Based on the Bayesian Information Criterion Applied to Unsupervised Speaker Indexing“, *IEEE Transactions on Speech and Audio Processing*, Band 13(4), S. 583–592, Jul. 2005.

[NK07] A. Noulas und B. J. A. Krose: „On-line multi-modal speaker diarization“, *International Conference on Multimodal Interfaces (ICMI'07)*, S. 350–357, New York, USA, Apr. 2007.

[OSBC06] M. Omologo, P. Svaizer, A. Brutti und L. Cristoforetti: „Speaker Localization in CHIL Lectures: Evaluation Criteria and Results“, *Lecture Notes in Computer Science: Machine Learning for Multimodal Interaction*, Band 3869, S. 476–487, 2006.

[OSG08] OSGI: Jan. 2008, [URL] http://www.osgi.org/.

[PAW06] J. Pardo, X. Anguerra und C. Wooters: „Speaker Diarization for Multi-microphone Meetings Using Only Between-Channel Differences“, *Lecture Notes in Computer Science: Machine Learning for Multimodal Interaction*, Band 4299, S. 257–264, 2006.

[PAW07] J. Pardo, X. Anguera und C. Wooters: „Speaker Diarization For Multiple-Distant-Microphone Meetings Using Several Sources of Information“, *IEEE Transactions on Computers*, Band 9(56), S. 1212–1224, Sep. 2007.

[PS08] E. Prud'hommeaux und A. Seaborne: „SPARQL Protocol and RDF Query Language“, Jan. 2008, [URL] http://www.w3.org/TR/rdf-sparql-query/.

[PTDL07] M. Papazoglou, P. Traverso, S. Dustdar und F. Leymann: „Efficient Semantic Service Discovery in Pervasive Computing Environments“, *IEEE Computer*, Band 40(11), S. 38–45, 2007.

[R+02] J. Rosenberg *et al.*: „SIP: Session Initiation Protocol“, Jun. 2002, [URL] http://tools.ietf.org/html/rfc3261/.

[R+03] J. Rosenberg *et al.*: „STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)“, Mär. 2003, [URL] http://www.ietf.org/rfc/rfc3489.txt.

[R+08] F. Ramparany *et al.*: „Amigo Software Repository: Ontology“, Jan. 2008, [URL] http://amigo.gforge.inria.fr/owl/.

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

[RBH03] P. Rigole, Y. Berbers und T. Holvoet: „A UPnP Software Gateway Towards EIB Home Automation“, *Proc. Conference on Computer Science and Technology (CST'03)*, Cancun, Mexiko, Mai 2003.

[RO98] D. Rosenthal und H. Okuno: *Computational Auditory Scene Analysis*, Lawrence Erlbaum Associates, Inc., Mahwah NJ, USA, 1998.

[RPS+07] F. Ramparany, R. Poortinga, M. Stikic, J. Schmalenstroeer und T. Prante: „An open Context Information Management Infrastructure“, *Proc. IET International Conference on Intelligent Environments (IE'07)*, Ulm, Deutschland, Sep. 2007.

[RQD00] D. Reynolds, T. Quatieri und R. Dunn: „Speaker Verification Using Adapted Gaussian Mixture Models“, *Digital Signal Processing*, Band 10(1-3), S. 19–41, Jan. 2000.

[RS04] J. Ramirez und J. Segura: „Efficient Voice Activity Detection Algorithms Using Long-Term Speech Information“, *Speech Communication*, Band 42(3-4), S. 271–287, Apr. 2004.

[RT05] D. Reynolds und P. Torres-Carrasquillo: „Approaches and Applications of Audio Diarization“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, S. 953–956, Philadelphia PA, USA, Mär. 2005.

[S+03] H. Schulzrinne *et al.*: „RTP: A Transport Protocol for Real-Time Applications“, Jul. 2003, [URL] http://tools.ietf.org/html/rfc3550/.

[SBG+05] D. Sacchetti, Y. Bromberg, N. Georgantas, V. Issarny, J. Parra und R. Poortinga: „The Amigo Interoperable Middleware for the Networked Home Environment“, *Proc. Middleware*, Grenoble, Frankreich, Dez. 2005.

[SH06] J. Schmalenstroeer und R. Haeb-Umbach: „Online Speaker Change Detection by Combining BIC with Microphone Array Beamforming“, *Proc. Conference of the International Speech Communication Association (Interspeech'06)*, Pittsburgh PA, USA, Sep. 2006.

[SLH08] J. Schmalenstroeer, V. Leutnant und R. Haeb-Umbach: „Amigo Context Management Service with Applications in Ambient Communication Scenarios“, *Communications in Computer and Information Science: Constructing Ambient Intelligence*, Band 11(7), S. 397–402, 2008.

[SML+08] A. Salah, R. Morros, J. Luque, C. Segura, J. Hernando, O. Ambekar, B. Schouten und E. Pauwels: „Multimodal identification and localization of users in a smart environment“, *Journal on Multimodal User Interfaces*, Band 2(2), S. 75–91, Sep. 2008.

[Spe08] Speex: „Audio codec“, 2008, [URL] http://www.speex.org.

[Spe09] Speex: „Comparison of speech codecs“, 2009, [URL] http://www.speex.org/comparison/.

[SS07] A. Schill und T. Springer: *Verteilte Systeme*, Springer Verlag, Heidelberg, Deutschland, 2007.

[SSM05] S. Surcin, R. Stiefelhagen und J. McDonough: „Deliverable D7.4 Evaluation Packages for the First CHIL Evaluation Campaign“, Mär. 2005, [URL] http://chil.server.de/.

[STGW05] R. Sinha, E. Tranter, M. Gales und P. Woodland: „The Cambridge University March 2005 Speaker Diarization System“, *Proc. Conference of the International Speech Communication Association (Interspeech'05)*, Lissabon, Portugal, Sep. 2005.

[The08] Theora: „Theora codec“, 2008, [URL] http://www.theora.org/.

[TMNS05] A. Temko, D. Macho, C. Nadeu und C. Segura: „CHIL - Acoustic Event Detection; UPC-TALP database of isolated meeeting-room acoustic events“, 2005, [URL] http://chil.server.de/.

[TMZ+06] A. Temko, R. Malkin, C. Ziegler, D. Macho, C. Nadeu und M. Omologo: „Acoustic Event Detection and Classification in Smart-Room Environments: Evaluation of CHIL Project Systems“, *Jornadas en Tecnologia del Habla*, Band 4, S. 1–6, Nov. 2006.

[TMZ+07] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu und M. Omologo: „CLEAR Evaluation of Acoustic Event Detection and Classification Systems“, *Lecture Notes in Computer Science: Multimodal Technologies for Perception of Humans*, Band 4122, S. 311–322, 2007.

[TR06] S. Tranter und D. Reynolds: „An Overview of Automatic Speaker Diarization Systems“, *IEEE Transactions on Audio, Speech and Language Processing*, Band 14(5), S. 1557–1565, Sep. 2006.

[UPn08] UPnP: „Universal Plug-and-Play“, 2008, [URL] http://www.upnp.org/.

[VJ01] P. Viola und M. Jones: „Rapid Object Detection using a Boosted Cascade of Simple Features“, *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, S. 511–518, Kauai, Hawaii, USA, Dez. 2001.

[Wei99] M. Weiser: „The computer for the 21st century“, *ACM SIGMOBILE Mobile Computing and Communications Review archive*, Band 3(3), S. 3–11, Jul. 1999.

[WH05] E. Warsitz und R. Haeb-Umbach: „Acoustic Filter-and-Sum Beamforming By Adaptive Principal Component Analysis“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, S. 797–800, Philadelphia PA, USA, Mär. 2005.

[WH06] C. Wu und C. Hsieh: „Multiple Change-Point Audio Segmentation and Classification Using an MDL-Based Gaussian Model“, *IEEE Transactions on Audio, Speech and Language Processing*, Band 14(2), S. 647– 657, Mär. 2006.

[WH07] E. Warsitz und R. Haeb-Umbach: „Blind Acoustic Beamforming Based on Generalized Eigenvalue Decomposition“, *IEEE Transactions on Audio, Speech and Language Processing*, Band 15(5), S. 1529–1539, Jul. 2007.

[Wik09a] Wikipedia: „Digital Subscriber Line“, 2009, [URL] http://de.wikipedia.org/wiki/DSL.

[Wik09b] Wikipedia: „List of audio codecs“, 2009, [URL] http://en.wikipedia.org/wiki/List_of_codecs.

[WM09] M. Wölfel und J. McDonough: *Distant Speech Recognition*, Wiley, Chichester, England, 2009.

[WP00] B. Wildermoth und K. Paliwal: „Use of voicing and pitch information for speaker recognition“, *Proc. IEEE Conference on Speech Science and Technology (SST'00)*, S. 324–328, Canberra, Australien, Dez. 2000.

[WPH04] E. Warsitz, S. Peschke und R. Haeb-Umbach: „Adaptive Beamforming Combined with Particle Filtering for Acoustic Source Localization“, *Proc. IEEE International Conference on Spoken Language Processing (ICSLP'04)*, S. 367–370, Jeju, Korea, Okt. 2004.

[WSH07] E. Warsitz, J. Schmalenstroeer und R. Haeb-Umbach: „Zweistufige Sprache / Pause-Detektion in stark gestörter Umgebung“, *Proc. German Annual Conference on Acoustics (DAGA'07)*, Stuttgart, Deutschland, Mär. 2007.

[WWW02] WWW: „Web Services“, 2002, [URL] http://www.w3.org/2002/ws/.

[YKA02] M. Yang, D. Kriegman und N. Ahuja: „Detecting Faces in Images: A Survey“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Band 24(1), S. 34– 58, Jan. 2002.

[Z+06] K. Zeilenga *et al.*: „Lightweight Directory Access Protocol (LDAP) - Technical Specification Road Map“, Jun. 2006, [URL] http://tools.ietf.org/html/rfc4510/.

[ZLB+05] X. Zhu, C. Leung, C. Barras, L. Lamel und J.-L. Gauvain: „Speech activity detection and speaker identification for CHIL“, Jan. 2005, [URL] ftp://tlp.limsi.fr/public/mlmi05-limsidsad.pdf.

[Zöl97] U. Zölzer: *Digitale Audiosignalverarbeitung*, B.G. Teubner, Stuttgart, Deutschland, 1997.

[ZSN05] A. Zolnay, R. Schlüter und H. Ney: „Acoustic Feature Combination For Robust Speech Recognition“, *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, S. 457–460, Philadelphia PA, USA, Mär. 2005.

Eigene Publikationen

[HKS05] R. Haeb-Umbach, B. Kladis und J. Schmalenstroeer: „Speech Processing in the Networked Home Environment - A View on the Amigo Project“, *Proc. Conference of the International Speech Communication Association (Interspeech'05)*, Lissabon, Portugal, Sep. 2005.

[HPF+09] M. Hennecke, T. Plötz, G. A. Fink, J. Schmalenstroeer und R. Haeb-Umbach: „A Hierarchical Approach to Unsupervised Shape Calibration of Microphone Array Networks“, *Proc. IEEE Workshop on Statistical Signal Processing (SSP'09)*, Cardiff, England, Aug. 2009.

[HS05] R. Haeb-Umbach und J. Schmalenstroeer: „A Comparison of Particle Filtering Variants for Speech Feature Enhancement“, *Proc. Conference of the International Speech Communication Association (Interspeech'05)*, Lissabon, Portugal, Sep. 2005.

[RPS+07] F. Ramparany, R. Poortinga, M. Stikic, J. Schmalenstroeer und T. Prante: „An open Context Information Management Infrastructure“, *Proc. IET International Conference on Intelligent Environments (IE'07)*, Ulm, Deutschland, Sep. 2007.

[SH06] J. Schmalenstroeer und R. Haeb-Umbach: „Online Speaker Change Detection by Combining BIC with Microphone Array Beamforming“, *Proc. Conference of the International Speech Communication Association (Interspeech'06)*, Pittsburgh PA, USA, Sep. 2006.

[SH07] J. Schmalenstroeer und R. Haeb-Umbach: „Joint Speaker Segmentation, Localization and Identification for Streaming Audio“, *Proc. Conference of the International Speech Communication Association (Interspeech'07)*, Antwerpen, Belgien, Aug. 2007.

[SLH08] J. Schmalenstroeer, V. Leutnant und R. Haeb-Umbach: „Amigo Context Management Service with Applications in Ambient Communication Scenarios“, *Communications in Computer and Information Science: Constructing Ambient Intelligence*, Band 11(7), S. 397–402, 2008.

[SLH09a] J. Schmalenstroeer, V. Leutnant und R. Haeb-Umbach: „Audio-visual Data Processing for Ambient Communication“, *Proc. Conference on Artificial Intelligence (KI'2009)*, Paderborn, Deutschland, Sep. 2009.

[SLH09b] J. Schmalenstroeer, V. Leutnant und R. Haeb-Umbach: „Fusing Audio and Video Information for Online Speaker Diarization“, *Proc. Conference of the International Speech Communication Association (Interspeech'09)*, Brighton, England, Aug. 2009.

[SWH07] J. Schmalenstroeer, E. Warsitz und R. Haeb-Umbach: „Projekt Amigo - Sprachsignalverarbeitung im vernetzten Haus“, *Proc. German Annual Conference on Acoustics (DAGA'07)*, Stuttgart, Deutschland, Mär. 2007.

[WSH07] E. Warsitz, J. Schmalenstroeer und R. Haeb-Umbach: „Zweistufige Sprach / Pause-Detektion in stark gestörter Umgebung“, *Proc. German Annual Conference on Acoustics (DAGA'07)*, Stuttgart, Deutschland, Mär. 2007.