

Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik

Diplomarbeit
Zur Erlangung des Grades
Diplom-Informatiker

**Evaluation von Verfahren
zur Aufbereitung von
Kehlkopfmikrofonsignalen
für eine robuste Spracherkennung**

Serguei Pronkine

Bonn, 2009

Betreuer:
Prof. Dr. Stefan Wrobel
Dr. Thomas Gärtner

Serguei Pronkine

Fraunhofer-Institut für Intelligente Analyse- und
Informationssysteme IAIS
Abteilung NetMedia
Schloss Birlinghoven
53754 Sankt Augustin

`serguei.pronkine@iais.fraunhofer.de`

Versicherung

Hiermit versichere ich, die vorliegende Arbeit selbständig verfasst und keine weiteren Hilfsmittel, als die im Rahmen der Arbeit angegebenen, genutzt zu haben.

Bonn,

Serguei Pronkine

Inhaltsverzeichnis

1. Einführung	7
1.1. Spracherkennung als Forschungsgebiet	7
1.2. Verwandte Arbeiten	9
1.3. Wissenschaftliche Fragestellung und Ziele	10
1.4. Aufbau der Arbeit	11
2. Grundlagen: Spracherkennung und Signalverarbeitung	13
2.1. Aufbau eines Spracherkennungssystems	13
2.2. Hidden-Markov-Modelle	16
2.3. Evaluationsmaße	18
2.4. Digitalisieren von Signalen	19
2.5. Signorräume	20
2.6. Fourier-Analyse	21
2.6.1. Fourier-Reihenentwicklung	21
2.6.2. Fourier-Transformation	22
2.6.3. Diskrete Fourier-Transformation	23
2.7. Filterung und Faltung	24
2.8. Merkmalsextraktion durch Mel-Frequency Cepstral-Analyse .	26
2.8.1. Fensterung	26
2.8.2. Spektralanalyse	29
2.8.3. Mel-Filterbank-Analyse	29
2.8.4. Dekorrelation	30
2.8.5. Dynamische Komponenten	31
2.8.6. Kodierungsparameter	32
3. Grundlagen: Robuste Spracherkennung mit Kehlkopfmikrofon	33
3.1. Charakteristik des Kehlkopfmikrofons	33
3.2. Lombard-Effekt	34
3.3. Lombard-Filter	35
3.3.1. Degradierungsmodell	36
3.3.2. Multilineare Regression	38
3.3.3. Zusammenfassung	41

3.4. Zero-Crossings with Peak Amplitudes	41
3.4.1. Zusammenfassung	45
4. Evaluationsumgebung	47
4.1. Sprachdatenbank	47
4.2. Signalanalyse	49
4.2.1. Vergleich von Signal und Störung	49
4.2.2. Motivation zum Lombard-Filter	50
4.2.3. Motivation zum ZCPA-Verfahren	52
5. Evaluierung der Verfahren	55
5.1. Aufbau der Evaluationsumgebung	55
5.1.1. Konfigurationsparameter	56
5.1.2. Phasen der Evaluierung	57
5.2. Auswertung des Lombard-Filters	58
5.3. Auswertung des ZCPA-Verfahrens	63
6. Zusammenfassung und Ausblick	67
A. Anhang	69
A.1. MoveOn Anwendungsgrammatik	69
A.2. Phonemalphabet	73
Literaturverzeichnis	75

Kapitel 1.

Einführung

Dieses Kapitel gibt einen Überblick über den Inhalt und den Aufbau dieser Arbeit. Dazu wird nach einer kurzen Einführung auf die Ziele der Arbeit und die Abgrenzung zu verwandten Arbeiten eingegangen.

1.1. Spracherkennung als Forschungsgebiet

Sprachliche Kommunikation ist eine der natürlichsten Modalitäten in der Kommunikation zwischen Menschen. Es liegt deshalb nahe, diese Schnittstelle auch in der Mensch-Maschine-Kommunikation zu verwenden. Der Einsatz einer Sprachschnittstelle eröffnet vielfältige Möglichkeiten wie vereinfachte Gerätesteuerung, Sprachdialogsysteme, Diktiersysteme und vieles mehr. Mit Einsatz der Spracherkennung ist es möglich geworden, große Audioarchive in Bibliotheken automatisch zu analysieren oder ein Navigationsgerät im Auto mit Sprachbefehlen zu programmieren, ohne sich von der Straße mit Betätigen von Tasten abzulenken. Im Gesundheitssektor unterstützt automatische Spracherkennung Ärzte bei der Erstellung medizinischer Berichte.

Die Forschung auf dem Gebiet der Spracherkennung begann bereits in den sechziger Jahren des letzten Jahrhunderts und war zunächst durch den Stand technischer Möglichkeiten eingeschränkt. Die ersten Spracherkennungssysteme analysierten isolierte Ziffern oder einsilbige Wörter mit Hilfe

einfacher Zeitbereichsmerkmale oder analoger Filterbänke. Wichtige Meilensteine in deren Entwicklung waren Algorithmen zur linearen Prädiktion und zum Vergleich zeitverzerrter Muster (Dynamic Time Warping) [Sakoe and Chiba, 1978]. Von entscheidender Bedeutung war das vom US-amerikanischen Verteidigungsministerium ins Leben gerufene Projekt DARPA SUR (Speech Understanding Research, 1971-76), das einen großen Erfolg verzeichnete [Schukat-Talamazzini, 1995].

In den achtziger Jahren setzte dann eine rasante Entwicklung ein. Die Parametrisierung des Sprachsignals durch die Mel-Cepstrum-Koeffizienten und deren dynamische Änderungen sowie die Diskretisierung der Daten mittels Vektorquantisierung wurden schnell *de facto* zu Standards [Schlüter, 2000; Schukat-Talamazzini, 1995]. Die Technik der Hidden-Markov-Modelle, die bereits seit 1975 bekannt war, erzielte gleichzeitig einen durchbruchartigen Erfolg. Mit der Technik wurde es erstmals möglich, die schwer durchschaubaren Zusammenhänge zwischen den Spracheinheiten (wie Wort, Silbe oder Phonem) und ihren akustischen Gegenstücken in einem Wahrscheinlichkeitsmodell zu fassen, dessen freie Parameter aus vorgelegten Sprachbeispielen geschätzt werden konnten [Schukat-Talamazzini, 1995]. Heute gehören zu den Vorreitern in der Industrie auf dem Gebiet der automatischen Spracherkennung IBM, Microsoft und Nuance Communications, die erfolgreich Produkte auf den Markt gebracht und somit zur Verbreitung und Popularisierung dieser Technologie in weiten Bereichen des privaten und geschäftlichen Lebens beigetragen haben.

Es ist auch heute noch praktisch unmöglich, ein System zu realisieren, welches das Spracherkennungsproblem allgemein löst. Die Schwierigkeiten dabei sind vielfältig:

- mehrere Sprecher,
- großes Vokabular,
- schlechte Artikulation,
- geräuschbelastete Umgebung,
- und vieles mehr.

Spracherkennungssysteme werden deshalb stets für *spezielle Anwendungen* konzipiert. Spezialfälle, für die das Erkennungsproblem unschwer zu lösen ist, sind beispielsweise folgende:

- einzeln gesprochene Wörter,
- kleines Vokabular,
- sprecherabhängige Spracherkennung

Eine der größten Herausforderungen auf dem Gebiet der Spracherkennung stellt der Umgebungslärm dar. In Anwesenheit einer Geräuschkulisse sinkt die Erkennungsleistung automatischer Spracherkennungssysteme meistens schnell ab, auch wenn man als Mensch unter gleichen Bedingungen dieselben Sprachsignale noch recht gut verstehen kann. Mit Einsatz robuster Algorithmen zur automatischen Spracherkennung und unter Verwendung mehrerer oder alternativer Aufnahmequellen ist es möglich, die Qualität unter schwierigen Bedingungen zu verbessern. Allgemein gilt ein System als robust, wenn seine Erkennungsleistung nur wenig von den Einflüssen wie Lärm, Echo, Sprecherakzent oder Ähnliches abhängt.

1.2. Verwandte Arbeiten

In der robusten Spracherkennung setzten sich im Umgang mit Umgebungsgeräuschen mehrere Techniken durch. Einige wichtige davon sind

- Subtraktion des Rauschens im Spektral- oder Merkmalsbereich,
- Extraktion geräuschbeständiger Merkmale,
- multikonditionales Training und
- Verwendung von Signalen aus mehreren oder alternativen Aufnahmequellen wie Mikrofon-Arrays oder Kehlkopfmikrofone.

Einigen dieser Techniken liegt das Modell des additiven Rauschens zu Grunde. Ein Sprachsignal wird dabei als Überlagerung des reinen Sprachsignals

und des Umgebungsrauschens modelliert. Ziel des Ansatzes der Spektralsubtraktion ist es, das Rauschen durch Subtraktion seines Energiespektrums zu reduzieren oder gar zu eliminieren [Benesty et al., 2008]. In Mansour and Juang [1989] wurde eine Technik mit Bezeichnung Short-Time Modified Coherence (SMC) vorgestellt, mit der es möglich war, eine robuste Repräsentation der Merkmale bei additivem weißen Rauschen zu gewinnen. Beim multikonditionalen Training wird das System anhand von Trainingsbeispielen aus verschiedenen Umgebungen trainiert, um möglichst realitätsnahe Bedingungen später am Einsatzort widerzuspiegeln [Benesty et al., 2008]. Die Verwendung von Kehlkopfmikrofonen erlaubt es, das Sprachsignal direkt am Hals des Sprechers aufzunehmen. Der Einfluss der Umgebung wird dabei auf ein Minimum reduziert. Zu den Hauptnachteilen gehört eine schlechtere Spektralcharakteristik als bei gewöhnlichen Nahsprechmikrofonen [Park et al., 2007].

1.3. Wissenschaftliche Fragestellung und Ziele

Wie eingangs erwähnt, beeinflussen ungünstige Umgebungsbedingungen die Qualität der Spracherkennung erheblich. Gründe dafür sind nicht nur die Belastung des Sprachsignals mit Umgebungsgeräuschen. Mit Einsatz von Kehlkopfmikrofonen gelingt es zum Beispiel, deren Einfluss von vornherein auf ein Minimum zu reduzieren — wenn auch nicht ganz zu eliminieren. Bei der Sprachkommunikation in lauten Umgebungen kommt zudem noch ein anderes Phänomen vor. Kommuniziert eine Sprecherin oder ein Sprecher in Anwesenheit einer Lärmquelle, verändert sie bzw. er die Artikulation und die Lautstärke, um sich trotzdem verständlich zu machen. Das resultiert in Variation der Tonhöhen, Tondauer und Veränderung der Intensität des Sprachsignals. Die Sprechgeschwindigkeit insgesamt ist ebenfalls betroffen. Dieser sogenannte Lombard-Effekt hat einen signifikanten negativen Einfluss auf den Erkennungsprozess. Auch wenn Kehlkopfmikrofone robust gegenüber Umgebungsgeräuschen sind, so hängt die Qualität der Spracherkennung auch bei diesen Signalen weiterhin — wenn auch gerin-

ger als bei Nahsprechmikrofonen — von der Umgebung ab. Während viele Arbeiten im Bereich der robusten Spracherkennung sich mit der Geräuschreduktion und der Verbesserung der Spracherkennung bei Nahsprechmikrofonen beschäftigen, ist es daher ebenfalls interessant, das Signal des Kehlkopfmikrofons zu verbessern, indem oben genannte Einflüsse weiter minimiert werden. Durch geeignete Kombination dieser robusten Mikrofone mit angepassten Verfahren der robusten Merkmalsextraktion und Merkmalsadaptation, bieten sich vielversprechende Möglichkeiten, die Qualität der automatischen Spracherkennung in gestörter Umgebung zu steigern. Ziel dieser Arbeit ist es, zwei vielversprechende Ansätze zu untersuchen, um die Robustheit von Kehlkopfmikrofonen im Spracherkennungsprozess weiter zu verbessern. Namentlich sind die Verfahren das “Lombard-Filter”, das den Einfluss des Lombard-Effekts verringern soll, und das auf Kehlkopfsignale angepasste “Zero-Crossings with Peak Amplitudes (ZCPA)” zur Extraktion robuster Merkmale. Beide Verfahren werden in dieser Arbeit exemplarisch auf dem “MoveOn Motorcycle Speech Corpus” evaluiert.

1.4. Aufbau der Arbeit

Der Rest der Arbeit ist wie folgt aufgebaut. In *Kapitel 2* werden grundlegende Begriffe der Sprachsignalverarbeitung eingeführt. Außerdem werden Grundlagen der Signalanalyse diskutiert, die für die Sprachsignalverarbeitung unentbehrlich sind. In *Kapitel 3* werden nach einer Charakteristik des Kehlkopfmikrofons drei Verfahren zur Aufbereitung von Kehlkopfmikrofonsignalen erläutert, die Gegenstand der Evaluation in der vorliegenden Arbeit in Hinblick auf eine robuste Spracherkennung sind. In *Kapitel 5* werden der Aufbau einer Testumgebung und Experimente beschrieben. Nach einer Präsentation der Evaluationsergebnisse werden diese analysiert und beurteilt. Im letzten *Kapitel 6* findet sich eine Zusammenfassung der durchgeführten Experimente und der wichtigsten Ergebnisse dieser Arbeit. Im Ausblick werden offene Fragen angesprochen, die künftige Arbeiten motivieren könnten.

Kapitel 2.

Grundlagen: Spracherkennung und Signalverarbeitung

In diesem Kapitel werden zunächst die wichtigsten Komponenten eines typischen Spracherkennungssystems und Evaluationsmaße für deren Leistung vorgestellt. Danach werden die Grundlagen digitaler Signalverarbeitung, also Signlräume und die darin definierten Transformationen, eingeführt. Als Vorbereitung darauf wird zuvor auf den Prozess der Signaldigitalisierung kurz eingegangen. Zum Schluss wird auf ein grundlegendes Verfahren zur Merkmalsextraktion eingegangen, das als Referenz bei den Experimenten verwendet wird.

2.1. Aufbau eines Spracherkennungssystems

Spracherkennung ist neben Sprechererkennung und Sprachsynthese ein zentraler Bereich der Sprachsignalverarbeitung. Die Aufgabe der Sprechererkennung ist es, den Sprecher einer gegebenen Aufnahme zu identifizieren, die Aufgabe der Sprachsynthese aus einem gegebenen Text ein Sprachsignal zu generieren. Im Gegensatz dazu besteht die Aufgabe der automatischen Spracherkennung¹ umgekehrt darin, ein Sprachsignal in den entsprechenden geschriebenen Text, die *Transkription*, umzusetzen.

¹engl. *automatic speech recognition, ASR*

Der prinzipielle Aufbau eines Spracherkennungssystems ist in Abbildung 2.1 gezeigt. Als Eingabe bekommt das System ein digitalisiertes Sprachsignal in Wellenform, also als Zeitfolge von Amplituden. Als Ausgabe erwartet man die erkannte Wortfolge als geschriebenen Text. Den Erkennungsprozess kann man grob in zwei Schritte unterteilen: Merkmalsextraktion und Klassifikation.

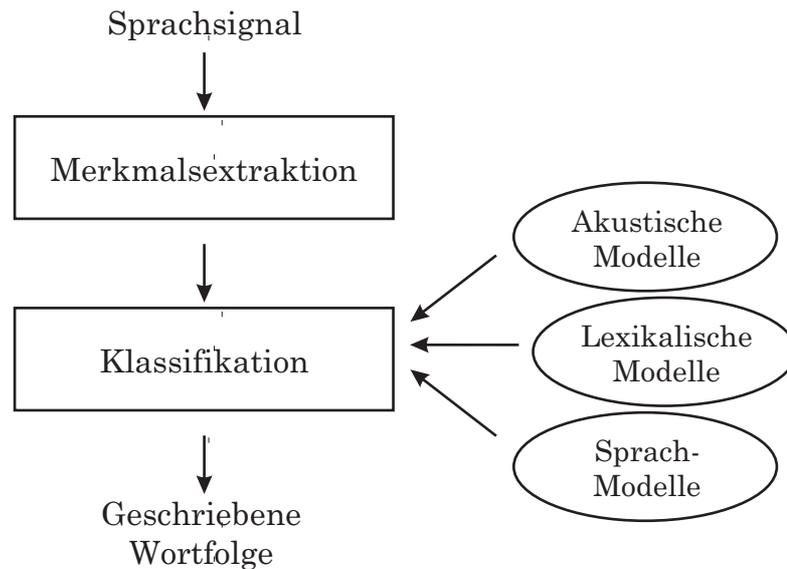


Abbildung 2.1.: Blockdiagramm: Aufbau des Spracherkennungssystems (in Anlehnung an Schlüter [2000]).

Der zeitliche Verlauf des Signals eignet sich kaum zur Spracherkennung, da dieser kaum Rückschlüsse auf die gesprochenen Laute erlaubt. Deshalb muss für das Signal eine geeignete Repräsentation gefunden werden. Dieser Prozess wird Merkmalsextraktion genannt.

Die Merkmalsrepräsentation sollte nur die für die Spracherkennung relevanten Informationen beinhalten. Sie muss also die Eigenschaft haben, dass sie die gesprochenen Laute möglichst gut charakterisiert, um diese voneinander abzugrenzen. Gleichzeitig sollte sie gegenüber Signalveränderungen, welche für die Klassifikation der Laute irrelevant oder gar störend sind.

Bei der Klassifikation werden die Merkmale lexikalischen Einheiten, wie

*Phonemen*², Wörtern oder ganze Sätzen, zugeordnet. Der Prozess muss auch die Variabilität in der Geschwindigkeit beim Sprechen berücksichtigen. Für diese Aufgabe hat sich der statistische Ansatz mit *Hidden-Markov-Modellen* (HMM, s. Abschnitt 2.2) *de-facto* als Standard durchgesetzt [Fink, 2003], [Schlüter, 2000].

Der Klassifikationsprozess stützt sich auf akustische Modelle, lexikalische Modelle und Sprachmodelle.

- Ein akustisches Modell erfasst Statistiken zur Aussprache einer lexikalischen Einheit wie Phonem, Silbe oder ganzes Wort. Jeder Einheit wird dabei ein akustisches Modell zugeordnet. Der Erkennungsprozess besteht darin, dass eine gegebene Merkmalsfolge (mit unbekannter Transkription) auf eine Folge von Hidden-Markov-Modellen abgebildet wird. Bei einer Einzelworterkennung mit Wortebene-HMMs, im einfachsten Fall, ist *ein* Modell gesucht, das die gegebene Merkmalsfolge mit der größten Wahrscheinlichkeit identifiziert. Im Falle der kontinuierlichen Sprache ist eine *Modellfolge* gesucht, deren Verbundwahrscheinlichkeit am größten ist.
- Ein lexikalisches Modell — im einfachen Fall ein Aussprachelexikon — definiert die Zusammensetzung lexikalischer Einheiten zu Wörtern. Es kann zusätzlich die Wahrscheinlichkeit angeben, mit der die Wörter in den Texten auftreten.
- Ein Sprachmodell definiert die Zusammensetzung der Worte zu Sätzen oder *Äußerungen*. Es beruht entweder ebenfalls auf einem statistischen Ansatz oder auf einem Regelwerk. Beim statistischen Ansatz wird die Sprache durch N-Gramme mit zugehörigen Auftretenswahrscheinlichkeiten in Texten repräsentiert. Bei einem Regelwerk handelt es sich um Grammatikregeln (sogenannte *task grammar*³), die die Möglichkeiten einschränken, Wörter miteinander zu kombinieren,

²Das Phonem ist die kleinste Einheit der lautlichen Sprache, die Bedeutungen unterscheidet. So gibt es z.B. für den Buchstaben 'r' mehrere Aussprachen, die zwar unterschiedliche Laute darstellen, jedoch zu einer Phonemklasse /r/ gehören, da sie beide gleiche Bedeutung haben.

³dt. etwa Anwendungsgrammatik

und somit den Hypothesenraum verkleinern.

2.2. Hidden-Markov-Modelle

Hidden-Markov-Modelle (HMM) sind für vielfältige Klassifikationsaufgaben neben der Spracherkennung nützlich wie z.B. Schriftenerkennung und Erkennung biologischer Sequenzen [Fink, 2003]. Die Darstellung in diesem Abschnitt folgt Young et al. [2006].

Unter einem *Hidden-Markov-Modell* λ versteht man zwei gekoppelte Zufallsprozesse. Der erste ist ein Markov-Prozess mit einer Anzahl Zuständen, die als S_1, S_2, \dots, S_N bezeichnet werden. Diese steuern den zweiten Zufallsprozess. Hier wird zu jedem diskreten Zeitpunkt t gemäß einer zustandsabhängigen Wahrscheinlichkeitsverteilung eine Beobachtung \mathbf{x}_t erzeugt. Beim Durchlaufen einer Sequenz von Zuständen $Q = q_1 q_2 \dots q_T$, mit $q_i \in \{S_1, S_2, \dots, S_N\}$, erzeugt das HMM eine Folge von *Beobachtungen* $\mathbf{X} = \mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_T$, $T \in \mathbb{N}$, die auch als *Emissionen* bezeichnet werden. Diese Zustände sind demnach emittierend. Zwei spezielle Zustände, der Startzustand S_1 und der Endzustand S_N , dienen der Verkettung mehrerer Modelle und sind nicht emittierend.

In Abbildung 2.2 ist der Aufbau eines typischen Hidden-Markov-Modells und dessen Funktionsweise gezeigt. Das HMM hat fünf Zustände S_1 bis S_5 . Die Kanten sind mit Übergangswahrscheinlichkeiten a_{ij} , $i, j \in \{S_1, \dots, S_5\}$ versehen. Selbstinduzierte Übergänge erlauben es, in einem Zustand zu verweilen. Andere Kanten führen zum nächsten oder übernächsten Zustand. Diese Topologie wird genutzt, um der Variabilität der Sprechgeschwindigkeit gerecht zu werden. In diesem Beispiel durchläuft das Modell die Folge $S_1, S_2, S_2, S_3, S_4, S_4, S_5$, wobei eine Sequenz $(\mathbf{x}_1, \dots, \mathbf{x}_5)$ mit Wahrscheinlichkeiten $b_j(\mathbf{x}_i) = P(\mathbf{x}_i | S_j)$, $i \in \{1, \dots, 5\}$ "beobachtet" wird. Die Verteilungen der Emissionswahrscheinlichkeiten können diskret oder kontinuierlich sein. Im letzteren Fall werden sie üblicherweise kompakt als Gaußsche Dichtefunktionen durch einen Mittelwert und Standardabweichung angegeben.

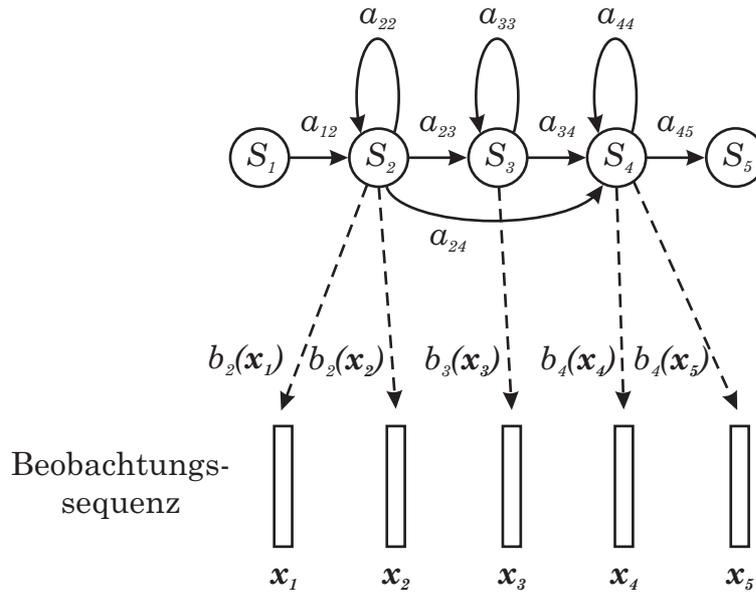


Abbildung 2.2.: Aufbau und Funktionsweise eines Hidden-Markov-Modells.

In der Spracherkennung stellen HMMs statistische Modelle der Wörter, Silben oder (meist) Phoneme dar. Ein HMM wird dabei einer solchen lexikalischen Einheit zugeordnet und Emissionen werden mit Merkmalsvektoren identifiziert. Das Erkennungsproblem (oder Dekodierung) kann nun wie folgt angegeben werden. Für eine gegebene Beobachtungssequenz \mathbf{X} und ein Modell λ ist diejenige Zustandsfolge $\hat{Q} = S_1 \hat{q}_1 \hat{q}_2 \dots \hat{q}_T S_N$ gesucht, die mit der Beobachtung mit der größten Posterior-Wahrscheinlichkeit übereinstimmt, d.h.

$$P(\mathbf{X}|\lambda) = \sum_{\mathbf{Q}} a_{q_0 q_1} \prod_{t=1}^{T-1} b_{q_t}(\mathbf{x}_t) a_{q(t)q(t+1)} \quad (2.1)$$

Diese Folge kann effizient mit Hilfe des *Viterbi-Algorithmus* berechnet werden. Unter allen Modellen wird als Hypothese das Modell ausgewählt, für das die Produktwahrscheinlichkeit (2.1) maximal ist.

2.3. Evaluationsmaße

Ein wichtiger Aspekt eines Spracherkenners ist seine Erkennungsleistung. Diese wird bestimmt, indem der Erkenner auf Testsprachsignale angewendet wird. Die vom ihm ausgegebene Transkription wird dann mit der korrekten Transkription unter Verwendung einer Abstandsmetrik verglichen und aus den Unterschieden die Erkennungsleistung ermittelt. Als Evaluationsmaße für die Erkennungsleistung sind unter anderem *Worterkennungsrate* und *Akkuratheit* (engl. *Accuracy*⁴) definiert.

Die Worterkennungsrate setzt sich aus der Anzahl der Wörter in der Referenztranskription N , Anzahl der Ersetzungen S (engl. *substitution*) und der Löschungen D (engl. *deletion*) wie folgt zusammen:

$$\text{Worterkennungsrate} = \frac{N - D - S}{N} \cdot 100\% \quad (2.2)$$

Diese Erkennungsrate bewegt sich zwischen 0% und 100%. Diese Kennzahl berücksichtigt jedoch keine Einfügungen. Deshalb wird noch die sogenannte *Akkuratheit* definiert. Mit I (engl. *insertion*) — Anzahl der Einfügeoperationen — gilt:

$$\text{Akkuratheit} = \frac{N - D - S - I}{N} \cdot 100\% \quad (2.3)$$

Akkuratheitswerte liegen im Intervall von $-\infty$ bis 100%. Dieses Maß ist repräsentativer für die Erkennungsleistung, da es nur dann bei 100% liegt, wenn die erkannte Äußerung zu 100% mit der korrekten Transkription übereinstimmt.

Die Anzahl der drei oben genannten Editieroperationen ist im Allgemeinen nicht eindeutig. Mit Verfahren der dynamischen Programmierung lässt sich jedoch die optimale Zuordnung (d.h. Zuordnung mit den wenigsten Fehlern) der erkannten Transkription zur Referenztranskription finden. Die minimale Anzahl Editieroperationen bezeichnet man als *Editier-* oder *Le-*

⁴dt. auch *Genauigkeit*

venshtein-Distanz⁵.

2.4. Digitalisieren von Signalen

Lautliche Sprache wird in Form von Schallwellen durch das Medium Luft übertragen und mit einem elektroakustischen Wandler, also einem Mikrofon, in ein analoges elektrisches Signal umgewandelt. Zur Verarbeitung von Signalen werden heute praktisch ausschließlich die Mittel der digitalen Signalverarbeitung eingesetzt. Deshalb wird ein analog aufgenommenes Signal im ersten Schritt digitalisiert.

Beim Digitalisieren wird ein analoges Signal abgetastet und quantisiert. Bei der Abtastung wird ein Signal $x(t)$ zu äquidistanten Zeitpunkten $t = nT, n \in \mathbb{N}$ abgetastet (engl. *sampling*). T ist das Abtastintervall, $f = 1/T$ die Abtastfrequenz. Nach dem Abtasttheorem von Shannon muss das Signal x mit der höchsten enthaltenen Frequenz Ω (*bandbegrenzte Signal*) mit mindestens doppelter Frequenz, 2Ω (*Nyquist-Frequenz*), abgetastet werden, damit x perfekt rekonstruiert werden kann. Anderenfalls entsteht Informationsverlust, und es können Aliasing-Effekte bei der Rekonstruktion auftreten.

Bei der (uniformen) Quantisierung bzw. Diskretisierung der Amplitude wird der Wertebereich auf eine Menge diskreter Werte abgebildet. Ziel dabei ist es, eine effiziente Speicherung ohne nennenswerte Verluste zu erreichen. Der Wertebereich wird dazu in $2^B, B \in \mathbb{N}$ Intervalle mit der Breite $\Delta x = 2 \cdot x_{max}/2^B$ zerlegt, und jedes x_t wird auf die am nächsten liegende Intervallmitte abgebildet. Dies setzt voraus, dass der Wertebereich auf das Intervall $[-x_{max}, x_{max}]$ eingeschränkt ist. B gibt dabei die Auflösung des Wertebereichs an. Für Sprachsignale sind die Abtastfrequenz von 16 kHz und eine Quantisierung mit Auflösung von 16 Bit ausreichend [Schukat-Talamazzini, 1995]. Für Musiksignale guter Qualität sind ca. 44,1 oder 48 kHz üblich.

⁵Benannt nach Vladimir I. Levenshtein, der das Verfahren 1965 veröffentlichte [Euler, 2006].

2.5. Signalräume

Die Signalanalyse in der Spracherkennung beruht zu einem wesentlichen Teil auf der Fourier-Transformation und dem Einsatz von Filtern. Die Fourier-Transformation überführt ein Signal, das von der Zeit abhängt, in eine Funktion der Frequenz und erlaubt damit einen Einblick in den Frequenzgehalt des Signals. Die Anwendung von Filtern ermöglicht unter anderem die Analyse eines ausgewählten Frequenzbereichs. Mathematische Grundlage der Filterung stellt die Faltung von Funktionen dar. Diese Begriffe werden nun nach und nach eingeführt. Die Darstellung in den Abschnitten 2.6 und 2.7 orientiert sich an Clausen and Müller [2001].

In der Analysis wird häufig von den Riemann-Integralen Gebrauch gemacht. Es gibt jedoch einige Funktionsfolgen, deren Grenzwerte nicht integrierbar sind und für die Riemann-Integration kein geeignetes Hilfsmittel für die Analyse darstellt. Deshalb bedient man sich sogenannter *Lebesgue-Integrale*, die unter Grenzprozessen wesentlich stabiler sind.

Für eine Funktion f sind die *Lebesgueschen Räume* $L^p(\mathbb{R})$ definiert durch

$$L^p(\mathbb{R}) : \{f : \mathbb{R} \rightarrow \mathbb{C} \mid f \text{ messbar und } \|f\|_p < \infty\}, \quad (2.4)$$

mit den Lebesgueschen Normen $\|f\|_p$, gegeben durch

$$\begin{aligned} \|f\|_p &:= \sqrt[p]{\int_{\mathbb{R}} |f(t)|^p dt} \quad \text{für } 1 \leq p < \infty \\ \|f\|_\infty &:= \text{ess sup}_{t \in \mathbb{R}} |f(t)| = \inf\{a \geq 0 \mid \mu(\{x : |f(x)| > a\}) = 0\} \\ &\quad \text{für } a \leq p \leq \infty. \end{aligned} \quad (2.5)$$

Ersetzt man in obigen Definitionen \mathbb{R} durch \mathbb{Z} , so ergeben sich die *Folgenräume* $\ell^p(\mathbb{Z})$. Diese Räume sind also besonders für die Analyse zeitdiskreter Signale geeignet, da diese Signale im mathematischen Sinne als Folgen interpretiert werden können.

2.6. Fourier-Analyse

Für die Zeit-Frequenz-Analyse eines (diskreten) Signals ist die sogenannte *diskrete Fourier-Transformation* grundlegend. Um diese Transformation einzuführen, bedarf es zuerst der Diskussion von Fourier-Reihen und der Fourier-Transformation für den kontinuierlichen Fall.

2.6.1. Fourier-Reihenentwicklung

Von allen Lebesgue-Räumen L^p und ℓ^p sind die sogenannten *Hilbert-Räume* mit $p = 2$ wichtig. Auf $H = L^2$ ist für $f \in L^2$ und $g(t) \in L^2$ ein Skalarprodukt definiert durch

$$\langle f|g \rangle := \int_{\mathbb{R}} f(t)\overline{g(t)}dt. \quad (2.6)$$

$H = L^2([0, 1])$ besitzt mehrere Orthonormalbasen (vgl. Satz zu Eigenschaften von Hilbert-Räumen in Clausen and Müller [2001]), unter anderen

$$\begin{aligned} \Phi_1 &= \{1, \sqrt{2}\cos(2\pi kt), \sqrt{2}\sin(2\pi kt) | k \in \mathbb{N}\} \quad \text{und} \\ \Phi_2 &= \{e^{2\pi ikt} | k \in \mathbb{Z}\}. \end{aligned} \quad (2.7)$$

Wie man im Folgenden sieht, ergibt sich die Fourier-Reihendarstellung einer periodischen Funktion aus der Darstellung zu einer Orthonormalbasis.

Sei $f : \mathbb{R} \rightarrow \mathbb{C}$ die periodische Fortsetzung einer Funktion aus $L^2([0, 1])$. Dann kann f zur Orthonormalbasis Φ_1 des Raumes als *Fourier-Reihe* dargestellt (oder “entwickelt”) werden:

$$f(t) = a_0 + \sqrt{2} \sum_{k=1}^{\infty} a_k \cdot \cos(2\pi kt) + \sqrt{2} \sum_{k=1}^{\infty} b_k \cdot \sin(2\pi kt). \quad (2.8)$$

Die Berechnung der Fourier-Koeffizienten $a_0, a_1, \dots, b_1, b_2, \dots$ erfolgt dabei

aus den Integralformeln:

$$\begin{aligned}
 a_0 &= \langle f|1 \rangle = \int_0^1 f(t) dt \\
 a_k &= \langle f|\sqrt{2}\cos(2\pi kt) \rangle = \sqrt{2} \int_0^1 f(t)\cos(2\pi kt) dt \\
 b_k &= \langle f|\sqrt{2}\sin(2\pi kt) \rangle = \sqrt{2} \int_0^1 f(t)\sin(2\pi kt) dt
 \end{aligned} \tag{2.9}$$

Die Fourier-Koeffizienten geben dabei an, mit welcher Intensität die Kosinus- und Sinusfunktionen verschiedener Frequenzen in f enthalten sind.

Für f wie oben ergibt sich aus der Darstellung zur Orthonormalbasis Φ_2 und der Eulerschen Identität $e^{2\pi ikt} = \cos(2\pi kt) + i \cdot \sin(2\pi kt)$ die Fourier-Reihe in komplexer Schreibweise durch

$$f(t) = \sum_{-\infty}^{\infty} c_k e^{2\pi ikt} \tag{2.10}$$

mit den Fourier-Koeffizienten

$$c_k = \langle f|e^{2\pi ikt} \rangle = \int_0^1 f(t) \overline{e^{2\pi ikt}} dt = \int_0^1 f(t) e^{-2\pi ikt} dt. \tag{2.11}$$

2.6.2. Fourier-Transformation

Die Fourier-Reihe erlaubt bis jetzt für eine periodische Funktion (bzw. die periodische Fortsetzung einer intervallbeschränkten Funktion) eine Darstellung als Überlagerung ganzzahliger Frequenzen. Ist eine Funktion f nicht periodisch, fällt aber für Werte gegen unendlich hinreichend gegen Null (was für gewöhnlich der Fall ist), dann gilt $f \in L^1(\mathbb{R})$, und man kann sie in Bezug auf ihre Frequenzen dennoch analysieren. Dabei wird der Frequenzraum kontinuierlich ($\omega \in \mathbb{R}$). Die *Fourier-Transformation* einer Funktion $f \in L^1(\mathbb{R})$ ergibt sich aus der komplexen Fourier-Reihendarstellung nach Formel 2.10 unter Berücksichtigung aller Frequenzen (und nicht nur ganz-

zahliger Frequenzen) durch

$$\hat{f}(\omega) := \int_{-\infty}^{\infty} f(t)e^{-2\pi i\omega t} dt. \quad (2.12)$$

Die Fourier-Transformierte einer Zeitfunktion bezeichnet man als (komplexes) *Spektrum*.

Die Fourier-Transformation lässt sich auch in $L^2(\mathbb{R})$ und $\ell^2(\mathbb{Z})$ übertragen (s. Clausen and Müller [2001]). Diese Räume sind zur Beschreibung für die Praxis relevanter kontinuierlicher und diskreter Signale geeignet.

Die Fourier-Transformierte \hat{x} einer Folge $x \in \ell^2(\mathbb{Z})$ ist definiert durch

$$\hat{x}(\xi) := \sum_{k=-\infty}^{\infty} x_k e^{-2\pi i k \xi}. \quad (2.13)$$

Für Folgen geht das Integral also in eine unendliche Summe über. Aus dieser Formel geht die nachfolgende diskrete Fourier-Transformation endlicher Folgen hervor.

2.6.3. Diskrete Fourier-Transformation

In praktischen Fällen hat man es mit diskreten Signalen endlicher Länge zu tun. Für diesen Fall ist die *diskrete Fourier-Transformation* definiert. Für den Vektor $v := (v_0, v_1, \dots, v_{N-1})^T \in \mathbb{C}^N, N \in \mathbb{N}$ ist die diskrete Fourier-Transformierte (DFT) als Vektor $\hat{v} \in \mathbb{C}^N$ definiert durch

$$\hat{v}_k := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} v_j e^{-2\pi i j k / N}, \quad k = 0, 1, \dots, N-1. \quad (2.14)$$

Die Komplexität für die Berechnung dieser Transformation liegt in $O(N^2)$. Mit einer Einschränkung für die Wahl von N ($N = 2^n, n \in \mathbb{N}$, Zweierpotenz) und durch geschickte Umsortierung der Multiplikationen und Additionen erhält man einen Algorithmus, der lediglich $O(N \cdot \log(N))$ solcher Ope-

rationen benötigt [Kiencke and Jäkel, 2008]. Diese *Schnelle Fourier Transformation*⁶ wird überwiegend für praktische Berechnungen benutzt.

2.7. Filterung und Faltung

Ein weiteres wichtiges Hilfsmittel zur Signalanalyse ist die Filterung. Die Filterung erlaubt es zum Beispiel, ein Signal bezüglich eines ausgewählten Frequenzbereichs zu analysieren. Faltung spielt dabei eine wichtige Rolle.

Abhängig vom gewünschten Effekt kann es notwendig sein, eine Menge von Filtern anzuwenden, eine sogenannte *Filterbank*. In der Filterbankanalyse passiert ein Sprachsignal eine Reihe von Filtern, die einzelne Bereiche von Frequenzen abdecken. Die Abstände der Mittenfrequenzen können gleichmäßig (uniform) oder nicht uniform sein, wie zum Beispiel in auditiv motivierten Mel- und Bark-Filterbänken (s. Abschnitt 2.8.3).

Je nach der Gestalt der Übertragungsfunktion, mit der ein Filter beschrieben wird, unterscheidet man Tief-, Hoch-, und Bandpassfilter. Das Ausgangssignal eines Filters entsteht durch die *Faltung* des Signals mit der Übertragungsfunktion des Filters. In der vorliegenden Arbeit wird von Filterbänken mit digitalen Bandpassfiltern endlicher Impulsantwort⁷ (d.h. die Impulsantwort enthält eine endliche Anzahl Koeffizienten ungleich Null) Gebrauch gemacht. Die mathematische Grundlage für (digitale) Filter sind digitale Systeme. Diese stellen einen Spezialfall sogenannter *linearer zeitinvarianter Systeme* dar.

Ein digitales System $T : E \rightarrow A$ transformiert ein Eingangssignal $x \in E$ in ein Ausgangssignal $y \in A$ (s. Abbildung 2.3). E und A bezeichnen dabei geeignete *Signalräume*, zum Beispiel Teilräume vom Raum zeitabhängiger Funktionen $\mathbb{R} \rightarrow \mathbb{C}$ (zeitkontinuierliche Signale) oder $\mathbb{Z} \rightarrow \mathbb{C}$ (zeitdiskrete Signale).

⁶engl. *Fast Fourier Transformation*, *FFT*

⁷engl. *Finite Impulse Response*, *FIR*



Abbildung 2.3.: Ein digitales System

Ein digitales System T heißt *linear*, wenn gilt

$$T[a_1x_1 + a_2x_2](n) = a_1T[x_1](n) + a_2T[x_2](n) \quad (2.15)$$

für beliebige a_1, a_2 und Signale x_1 und x_2 aus E .

Das System T heißt *linear zeitinvariant*, wenn zusätzlich gilt:

$$y[n_0] = T[x](n - n_0) \quad (2.16)$$

für $n, n_0 \in \mathbb{R}$.

Sind $x, y : \mathbb{Z} \rightarrow \mathbb{C}$ diskrete Signale, so heißt

$$(x * y)(n) := \sum_{k \in \mathbb{Z}} x(k)y(n - k) \quad (2.17)$$

die *Faltung* (engl. *convolution*) von x und y an der Stelle $n \in \mathbb{Z}$.

Mit dem folgenden Satz (2.18) wird es möglich das Ausgabesignal y eines Filters wahlweise im Frequenz- oder im Zeitbereich zu berechnen.

Seien $h, g \in \ell^2(\mathbb{Z})$, dann gilt entsprechend dem *Faltungssatz*:

$$\widehat{h * g} = \hat{h} \cdot \hat{g}. \quad (2.18)$$

Mit anderen Worten, die Faltung zweier Folgen geht unter Verwendung der Fourier-Transformation in die punktweise Multiplikation der Fourier-Transformierten über.

Das Ausgabesignal eines digitalen Filters wird für längere Signale aus Effizienzgründen für gewöhnlich im Spektralbereich berechnet. Dazu werden das

Eingabesignal und die Übertragungsfunktion des Filters mittels Diskreter Fourier-Transformation in den Spektralraum überführt und dort punktweise multipliziert. Die Übertragungsfunktion im Spektralraum wird auch Filterkern genannt. Gegebenenfalls muss das Ausgabesignal mittels entsprechender inverser Fourier-Transformation wieder in den Zeitraum rücktransformiert werden.

2.8. Merkmalsextraktion durch Mel-Frequency Cepstral-Analyse

Die Extraktion zuverlässiger Merkmale ist eine zentrale Aufgabe im Prozess der Spracherkennung. Wie eingangs erwähnt ist die Repräsentation des Sprachsignals im Zeitbereich sehr redundant und eignet sich kaum für eine zuverlässige Klassifizierung. Deshalb geht man für gewöhnlich zu einer geeigneteren Repräsentation des Signals über — der Merkmalsrepräsentation.

Die Mel-Frequency Cepstral-Analyse ist eine Möglichkeit, eine solche Merkmalsrepräsentation des Signals zu finden. Abbildung 2.8 veranschaulicht die Schritte zur Extraktion der *Mel-Frequency Cepstral-Koeffizienten* (engl. *Mel-Frequency Cepstral Coefficients, MFCC*) als Blockdiagramm. Als Eingabe wird ein Sprachsignal in Wellenform erwartet. Die Ausgabe ist eine Merkmalssequenz, also eine Sequenz von Merkmalsvektoren einer festen Länge.

Im Folgenden werden die einzelnen Schritte detailliert besprochen.

2.8.1. Fensterung

Die Signalanalyse erfolgt in der Regel in sich überschneidenden kurzen Zeitintervallen, in denen das Signal als stationär angenommen werden kann. Zu diesem Zweck wird es mit Hilfe einer speziellen *Fensterfunktion* multipliziert. Um einer möglichen Entstehung großer Sprünge an den Grenzen

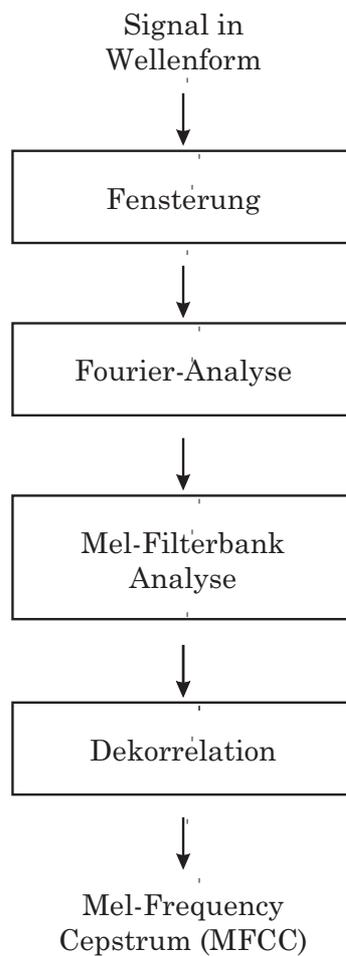


Abbildung 2.4.: Blockdiagramm: Extraktion des Mel-Frequency Cepstrums

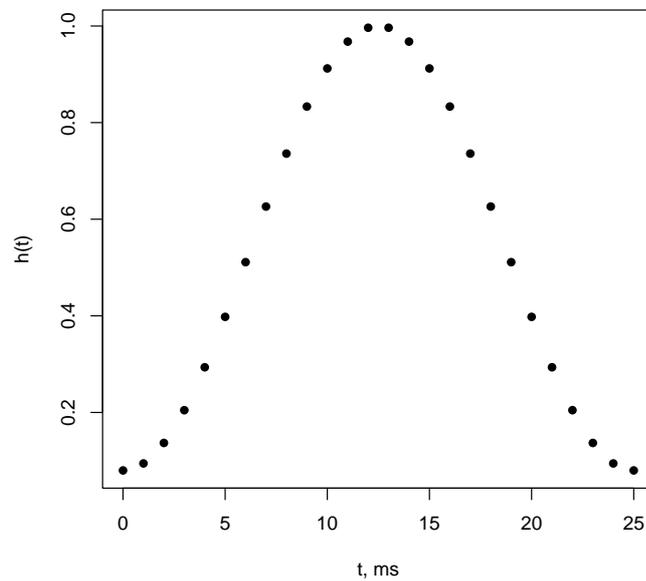


Abbildung 2.5.: Hamming-Funktion (hier mit $L = 25$).

entgegenzuwirken wird das Signal durch die Wahl einer geeigneten Funktion an den Seitenenden gedämpft. Eine sehr gängige Fensterfunktion, die häufig zum Einsatz kommt, ist die *Hamming*-Funktion. Sie ist definiert durch

$$h[t] = \begin{cases} 0,54 - 0,46 \cdot \cos\left(\frac{2\pi t}{L}\right), & 0 \leq t \leq L - 1, t \in \mathbb{Z} \\ 0, & \text{sonst.} \end{cases} \quad (2.19)$$

und ist in Abbildng 2.5 dargestellt. Das Hamming-Fenster ist insbesondere wegen seines vorteilhaften Auflösungsverhaltens im Zeit- und Frequenzbereich in der Praxis beliebt.

Zu jedem Zeitpunkt t_0 wird durch Verschiebung der Fensterfunktion und punktweise Multiplikation aus dem Signal x eine gewichtete Version von x

in einer kleinen Umgebung um t_0 berechnet:

$$x^{(t_0)}[t] = x[t] \cdot h[t - t_0]. \quad (2.20)$$

Außerhalb der Umgebung verschwindet das Signal.

Eine typische Fensterlänge zur Extraktion von MFCCs beträgt 25 ms bei einer Fensterverschiebung von 10 ms [Schukat-Talamazzini, 1995].

2.8.2. Spektralanalyse

Für jedes Fenster wird nun mittels Diskreter Fourier-Transformation ein Kurzzeit-Spektrum ermittelt. Das Zeitsignal x wird dabei nach Formel 2.14 in Abschnitt 2.6.3 in den Frequenzbereich überführt. Das Spektrum ist eine komplexe Funktion der Frequenz. Da die Phasenbeziehungen im Bereich der Spracherkennung keine große Rolle spielen, geht man an dieser Stelle zum reellen Betragsspektrum über. Die Faltungseigenschaft (vgl. Faltungssatz 2.18) bleibt dabei erhalten [Schukat-Talamazzini, 1995].

2.8.3. Mel-Filterbank-Analyse

Die menschliche Wahrnehmung der Tonhöhen ist nicht in allen Frequenzbereichen gleich. Untersuchungen über die subjektive Empfindung der Tonhöhen ergaben, dass die empfundene Höhe eines Tons sich nichtlinear mit dessen Frequenz verhält [Pfister and Kaufmann, 2008].

Eine Analyse mit der sogenannten Mel-Filterbank berücksichtigt diese frequenzabhängige Tonhöhenempfindung. Die Mel-Skala, die in Moore and Glasberg [1983] beschrieben wurde, steht mit der Frequenz f in folgendem Zusammenhang:

$$\text{Mel} = 1127 \cdot \ln \left(1 + \frac{f}{700 \text{ Hz}} \right)$$

Die Filterbank selbst stellt eine Menge von p Filtern mit Zentrumsfrequen-

zen, die äquidistant auf der Mel-Skala liegen, dar. Im Bereich der automatischen Spracherkennung werden überwiegend Filter in triangularer Form verwendet. Abbildung 2.6 veranschaulicht die Filterbank. Die Anzahl p der Filter kann empirisch bestimmt werden und liegt Empfehlungen nach zwischen 24 und 26 [Pfister and Kaufmann, 2008], [Young et al., 2006].

Zu den anderen verbreiteten Skalen zur Beschreibung der Tonhöhenempfindung gehören die Bark-Skala und die ERB-Skala (equivalent rectangular bandwidth, ERB⁸). Eine Beschreibung dazu findet sich zum Beispiel in Moore and Glasberg [1983].

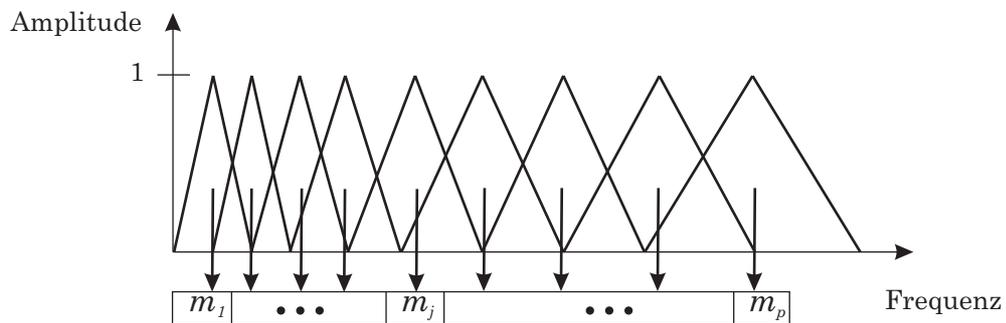


Abbildung 2.6.: Mel-Filterbank mit p Filtern.

Die Anwendung der Filterbank funktioniert nun wie folgt. Das Betragsspektrum wird mit jedem Filter punktweise multipliziert (vgl. Faltungssatz 2.18). Die Mel-Spektralkoeffizienten m_j , $j \in 1, \dots, p$ werden gewonnen, indem die Koeffizienten in jedem Kanal aufsummiert werden. Das Ergebnis wird als Mel-Spektrum bezeichnet. Wegen des großen Dynamikbereichs des Spektrums geht man für gewöhnlich anschließend zur logarithmierten Darstellung über.

2.8.4. Dekorrelation

Benachbarte Koeffizienten des Mel-Spektrums sind stark korreliert, da sich die Filterkerne der Mel-Filterbank überlappen (vgl. Abbildung 2.6). Für

⁸dt. etwa Äquivalentrechteck-Bandbreite

die Klassifikation mit den meisten Hidden-Markov-Modellen ist es jedoch notwendig, dass die Koeffizienten eines jeden Merkmalsvektors paarweise unabhängig sind [Young et al., 2006]. Ein endgültiger Merkmalsvektor wird deshalb durch Dekorrelation aus dem Mel-Spektrum gewonnen. Die Dekorrelation wird hier durch Anwendung der inversen Fourier- oder einer ähnlichen Transformation (wie zum Beispiel Cosinus-Transformation) erreicht. Dieses “Spektrum vom Spektrum” wird *Cepstrum* bezeichnet. Der Begriff “Cepstrum” wurde als Kunstwort aus “spectrum” abgeleitet [Fink, 2003].

2.8.5. Dynamische Komponenten

Die Leistung der Spracherkennung kann durch Hinzunahme *dynamischer Merkmale* signifikant gesteigert werden. Es handelt sich dabei um zeitliche Ableitungen der (statischen) Merkmale, die in den vorherigen Schritten gewonnen wurden. Es gibt verschiedene Möglichkeiten, dynamische Komponenten zu bestimmen. Hier wird exemplarisch eine dieser Möglichkeiten erläutert. Die zeitliche Ableitung erster Ordnung, die auch als *Delta-Koeffizient* bekannt ist, kann zum Beispiel als gewichtete Summe aus Koeffizienten in $2n + 1$ benachbarten Fenstern gewonnen werden. Seien $c_{t-n}, \dots, c_{t-1}, c_t, c_{t+1}, \dots, c_{t+n}$ statische (eindimensionale) Koeffizienten in der Umgebung der Zeit t . Dann ist ein Delta-Koeffizient d_t definiert durch

$$d_t = \frac{\sum_{j=1}^n j(c_{t+j} - c_{t-j})}{2 \sum_{j=1}^n j^2} \quad (2.21)$$

Üblicherweise wird n gleich zwei gewählt.

Die Berechnung der Deltas bezieht Vergangenheits- und Zukunftswerte der Merkmale mit ein, die am Anfang und Ende der Merkmalsrepräsentation eines Signals nicht definiert sind. Durch Replikation der Merkmale am Anfang bzw. Ende wird die obige Definition sinngemäß ergänzt.

Analog dazu ist die Ableitung zweiter Ordnung, ein *Accelerator-* oder *Delta-von-Delta-Koeffizient*, als Ableitung der Delta-Koeffizienten definiert und

wird nach derselben Formel 2.21 (mit entsprechenden Ersetzungen) berechnet.

2.8.6. Kodierungsparameter

Viele Kodierungsparameter, die in den obigen Abschnitten erwähnt wurden, hängen von der konkreten Anwendung ab. Die optimalen Werte müssen deshalb jeweils experimentell bestimmt werden. Es gibt jedoch gute Erfahrungswerte für den Bereich der Spracherkennung, die in der Regel zu guten Ergebnissen führen. Diese Parameter sind in Tabelle 2.1 zusammengestellt [Pfister and Kaufmann, 2008], [Young et al., 2006].

Tabelle 2.1.: Parameterwerte, die bei der Mel-Frequency Cepstral-Analyse häufig verwendet werden.

Kodierungsparameter	Wert
Abtastrate	8 oder 16 kHz
Länge des Analysefensters	25 ms
Verschiebung des Analysefensters	10 ms
Anzahl triangularer Filter	24 – 26
Anzahl Cepstral-Koeffizienten	12 – 16

Kapitel 3.

Grundlagen: Robuste Spracherkennung mit Kehlkopfmikrofon

In diesem Kapitel wird zuerst das Kehlkopfmikrofon als Aufnahmequelle beschrieben. Danach werden zwei Verfahren erläutert, die in der vorliegenden Arbeit im Hinblick auf eine robuste Spracherkennung evaluiert werden. Zum einen wird das sogenannte Lombard-Filter vorgestellt. Es soll für Robustheit gegenüber der Variabilität der Sprache beim Kommunizieren in geräuschbelasteten Umgebungen sorgen. Das Filter beruht auf einer Methode der Regressionsanalyse, der multilinearen Regression. Zum anderen wird das Verfahren Zero-Crossings with Peak Amplitudes vorgestellt, das auf einem auditiven Modell basiert und es ermöglichen soll, gegenüber Hintergrundgeräuschen robuste Sprachmerkmale zu extrahieren.

3.1. Charakteristik des Kehlkopfmikrofons

Ein Kehlkopfmikrofon ist ein Sensor, der am Hals in der Nähe des Adamsapfels getragen wird. Im Gegensatz zu gewöhnlichen Nahsprechmikrofonen, die den Luftschall aufnehmen, nimmt ein Kehlkopfmikrofon Vibrationen

der Haut auf, die von den Stimmbändern erzeugt, im Vokaltrakt verändert und verstärkt und durch den Kehlkopf übertragen werden.

Durch diese Funktionsweise wird die Auswirkung der Umgebung im Signal deutlich reduziert, da der Körperschall und nicht der Luftschall aufgenommen wird. Kehlkopfmikrofone sind deshalb sehr gut für den Einsatz in lauten Umgebungen geeignet und werden zum Beispiel im Militärbereich zur Kommunikation eingesetzt.

Der Hauptnachteil von Kehlkopfmikrofonen besteht darin, dass die Signale für gewöhnlich einen ungünstigeren Frequenzverlauf aufweisen als gewöhnliche Nahsprechmikrofone. Insbesondere höhere Frequenzen im Bereich zwischen ca. 2 und 4 kHz werden abgeschwächt und ab 4 kHz fast vollständig unterdrückt [Jung et al., 2007]. Dies führt teilweise zum Verlust der sogenannten Formanten¹, die wichtig für die Erkennung der Vokale sind. Dennoch sind Signale aus dem Kehlkopfmikrofon für das menschliche Ohr gut verständlich, da die wichtigsten Informationen für die Erkennung von Lauten und Wörtern in niedrigen Frequenzbereichen konzentriert sind. In der automatischen Spracherkennung wirken sich die fehlenden Frequenzanteile im Spektrum eines Kehlkopfmikrofonsignals jedoch deutlich negativ auf die Erkennungsleistung aus.

3.2. Lombard-Effekt

Die Verständigung in lauten Umgebungen, zum Beispiel an einer stark frequentierten Straße, stellt für den Menschen eine Herausforderung dar. Um sich trotzdem verständlich zu machen, sprechen Konversationspartner akzentuierter und lauter. Dadurch ändert sich die Intensität, die Ton-

¹Lokale Maxima des Leistungsdichtespektrums des Sprachsignals werden in der Phonetik als Formanten F_1 , F_2 , F_3 etc. bezeichnet. Diese spektralen Maxima rühren von Resonanzen des Vokaltraktes her und sind besonders bei Vokalen ausgeprägt vorhanden. Ein Formant wird mit den Parametern Mittenfrequenz (oder Formantfrequenz), Bandbreite und Amplitude beschrieben. Die Bezeichnungen F_1 , F_2 , F_3 etc. werden oft auch für die Formantfrequenzen verwendet. Für die Formantfrequenzen eines Lautes gilt: $F_1 < F_2 < F_3$ etc. Der tiefste Formant oder die tiefste Mittenfrequenz ist also stets F_1 [Pfister and Kaufmann, 2008].

lage und die Artikulation: die Betonung der Vokale verschiebt sich. (Im Grenzfall geht die Sprache ins Schreien über.) Diese Effekte werden als *Lombard-Effekt*² zusammengefasst. Da der Effekt teilweise unbewusst auftritt, wird er manchmal auch *Lombard-Reflex* genannt. Sprache, die unter diesen Umständen aufgenommen wurde, wird als Lombard-Sprache bezeichnet.

Studien zufolge sind Auswirkungen des betreffenden Effekts ziemlich komplex. Im Einzelnen zeigen sie, dass bei ausgeprägtem Lombard-Effekt die Längen der Vokale größer werden, während die der Konsonanten im Gegenteil dazu kleiner werden. Genannt wird auch eine Verschiebung der Intensitätsschwerpunkte (Formanten) aus den höheren und unteren Frequenzbereichen in einen mittleren Bereich. Die Auswirkungen hängen jedoch auch von der Art der Umgebungsgeräusche und deren Pegel ab [Hansen and Varadarajan, 2009].

Wie eingangs erwähnt, hat der Lombard-Effekt einen negativen Einfluss auf die Erkennungsleistung in der automatischen Spracherkennung. Ziel der im Folgenden erklärten Methode ist es, die Auswirkungen des Effekts zu reduzieren, um die Erkennungsleistung des Spracherkennungssystems zu verbessern.

3.3. Lombard-Filter

In einer Arbeit von Chi and Oh [1996] wird ein Verfahren beschrieben, das zum Ziel hat, den Lombard-Effekt auf der Merkmalsebene zu kompensieren. Zu diesem Zweck wurde ein Degradierungsmodell aufgestellt, das die spektralen Veränderungen der Sprache, die von dem sogenannten Lombard-Effekt herrühren, erfasst. Im Einzelnen werden Frequenzverformungen (engl. *frequency warping*) und Amplitudenskalierungen im Spektralbereich in Abhängigkeit vom Frequenzband modelliert. Dadurch wer-

²Die ersten wissenschaftlichen Beobachtungen dieses Effekts gehen zurück in das Jahr 1911, als der französische Wissenschaftler Étienne Lombard (1868-1920) diesen physiologischen Effekt in lauter Umgebung entdeckte [Hansen and Varadarajan, 2009].

den Variationen der Formantenhöhen und -breiten, der Tonhöhen und der Energie simuliert.

Zur Validierung des Verfahrens bauten die Autoren eine Testumgebung auf. Dabei wurden mit einem Nahsprechmikrofon saubere (“clean”) und durch den Lombard-Effekt gestörte (“Lombard”) Sprachsegmente in koreanischer Sprache aufgenommen. Die Lombard-Sprache wurde wie folgt aufgenommen. Probanden bekamen Kopfhörer aufgesetzt, durch die eine Geräuschkulisse simuliert wurde. Sie mussten währenddessen ins Mikrofon Wörter aufsprechen. Auf diese Weise wurde Lombard-Sprache *ohne* Hintergrundgeräusche aufgenommen.

Bei der Auswertung des Verfahrens wurde eine signifikante Verbesserung bei der Einzelworterkennung in koreanischer Sprache erzielt. Dabei wurden MFCC-Merkmale (vgl. Abschnitt 2.8) mit und ohne Lombard-Effekt-Unterdrückung gegenübergestellt. Die verwendeten Umgebungen umfassten Geräusche innerhalb von Ausstellungshallen, Telefonzellen, Autokabinen, Computerräumen und auf belebten Straßen unter verschiedenen Werten des Signal-zu-Rausch-Abstands.

Das Verfahren zur Lombard-Effektunterdrückung wird in dieser Arbeit fortan auch als *Lombard-Filter* bezeichnet. Die Basis des Lombard-Filters stellt ein *Degradierungsmodell* dar, das später die Adaptation an den Lombard-Effekt mittels eines Regressionsverfahrens rechtfertigt.

3.3.1. Degradierungsmodell

Um die Auswirkungen des Lombard-Effekts auf das Sprachsignal zu erfassen, wurde in Chi and Oh [1996] ein sogenanntes Degradierungsmodell (engl. *degradation³ model*) entwickelt. Darin werden zum einen Variationen der Formantenhöhe, deren Bandbreiten und Energien in einer Funktion “nichtlinearer Frequenzverformungen” $F(\cdot)$, zum anderen frequenzabhängige Amplitudenskalierungen in einer Funktion $A(\cdot)$ modelliert. Die Amplitudenskalierungen rühren von der Variation des Schalldruckpegels

³dt. etwa Herabstufung, Herabsetzung, Abstufung

(Lautstärke) beim Sprechen her. Mit der Bezeichnung des Spektrums $S(\omega)$ eines Sprachsignals ergibt sich eine Transformation im Spektralbereich

$$Y(\omega) = A(\omega)S(F(\omega)), \quad (3.1)$$

wobei $Y(\omega)$ dem Spektrum des Lombard-Signals entspricht.

Die Funktionen $F(\cdot)$ und $A(\cdot)$ sind einzeln unbekannt. Die Gesamtauswirkung kann aber durch Beispiele "sauberer" Merkmale und Lombard-Merkmale abgeschätzt werden. Im Folgenden findet sich die Herleitung hierzu.

Sei C_n^{clean} der n -te Cepstralkoeffizient der "sauberen" Sprache und $C_n^{Lombard}$ ein entsprechender Koeffizient der Lombard-Sprache. C_n^{clean} entsteht aus dem Mel-Kurzzeitspektrum $S(\omega)$ durch Logarithmierung und inverse Fourier-Transformation (vgl. Abschnitt 2.8):

$$C_n^{clean} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|S(\omega)|) e^{i\omega n} d\omega \quad (3.2)$$

Für Lombard-Sprache ergibt sich:

$$C_n^{Lombard} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|A(\omega)S(F(\omega))|) e^{i\omega n} d\omega, \quad (3.3)$$

woraus sich durch Anwenden der Fourier-Transformation 2.12 folgende Gleichung ergibt:

$$\log(|A(\omega)S(F(\omega))|) = \sum_{k=-\infty}^{\infty} C_k^{Lombard} e^{-i\omega k} \quad (3.4)$$

Durch Erweitern der Gleichung 3.2 mit $\omega = F(F^{-1}(\omega))$ und Einsetzen der Gleichung 3.4 erhält man:

$$\begin{aligned}
 C_n^{clean} &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \log(|S(F(F^{-1}(\omega)))|) e^{i\omega n} d\omega \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\sum_{k=-\infty}^{\infty} C_k^{Lombard} e^{-iF^{-1}(\omega)k} - \log(|A(F^{-1}(\omega))|) \right] e^{i\omega n} d\omega \quad (3.5)
 \end{aligned}$$

Mit den Bezeichnungen

$$L(n, k) = -\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-iF^{-1}(\omega)k} e^{i\omega n} d\omega \quad (3.6)$$

und

$$M(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |A(F^{-1}(\omega))| e^{i\omega n} d\omega \quad (3.7)$$

ergibt sich

$$C_n^{clean} = \sum_{k=-\infty}^{\infty} L(n, k) \cdot C_k^{Lombard} + M(n). \quad (3.8)$$

Hier wurde also gezeigt, dass die Transformation 3.1 des Signals im Spektralbereich in einer linearen Abhängigkeit zwischen den “clean”- und “Lombard”-Merkmalsvektoren im Cepstralbereich resultiert. Zur Auffindung dieser Abhängigkeit wird an dieser Stelle die Regressionsanalyse herangezogen.

3.3.2. Multilineare Regression

Einfaches lineares Modell

Die lineare Regression ist die wichtigste (und einfachste) Form in der Regressionsanalyse. Gegeben seien zwei reelle und korrelierte Variablen. Sei X die *unabhängige* und Y die von X *abhängige* Variable. Die Aufgabe der linearen Regressionsanalyse ist es, einen linearen Zusammenhang in Form

einer Geraden mit Hilfe einer Stichprobe $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ zu ermitteln. Diese Gerade wird als *Regressions-* oder *Ausgleichsgerade* bezeichnet.

Das lineare Modell kann angegeben werden als

$$y_i = \beta_0 + \beta_1 \cdot x_i + \varepsilon_i, \quad i = 1, \dots, N,$$

wobei N die Anzahl der Messungen oder Beobachtungen ist. (x_i, y_i) sind die Datenpunkte, β_0 und β_1 werden Regressionskoeffizienten genannt und ε_i Fehlergrößen, oder auch *Residuen*.

Ziel ist es, die Parameter β_0 und β_1 der Regressionsgeraden so zu bestimmen, dass die Gerade möglichst nah an allen Datenpunkten liegt. Diese Überlegung führt zur Bedingung

$$\sum_{i=1}^N \varepsilon_i^2 \rightarrow \min,$$

also zur Minimierung der Summe der Fehlerquadrate.

Unter den Annahmen, dass die Fehlergrößen rein zufällig sind und um den Nullpunkt streuen ($\mathbb{E}(\varepsilon) = 0$), und dass es genügend Beobachtungen gibt, ergibt sich die Lösung des Problems durch partielles Differenzieren und Nullsetzen der Ableitungen erster Ordnung (ohne Herleitung):

$$\beta_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \quad \text{und} \quad \beta_0 = \bar{y} - \beta_1 \bar{x}, \quad \text{wobei}$$

$$\bar{x} = \frac{1}{N} \sum_i x_i \quad \text{und} \quad \bar{y} = \frac{1}{N} \sum_i y_i.$$

Multilineares Regressionsmodell

Eine Generalisierung des obigen Modells ist die *multilineare* Regression. Sei $y \in \mathbb{R}$ nun von mehreren vorgegebenen Variablen $x_1, x_2, \dots, x_p \in \mathbb{R}$

abhängig. Das lineare Modell erhält dann die Form

$$y_i = \beta_0 \cdot 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, \dots, N$$

wobei N die Anzahl der Beobachtungen ist und $\varepsilon_1, \dots, \varepsilon_N$ wieder die Störgrößen repräsentieren.

Mit den Bezeichnungen

$$Y \in \mathbb{R}^{N \times 1}, \bar{\varepsilon} \in \mathbb{R}^{N \times 1}, \bar{\beta} \in \mathbb{R}^{(p+1) \times 1}$$

und X – erweiterte Matrix der Form

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & & & \vdots \\ 1 & x_{N1} & \cdots & x_{Np} \end{pmatrix} \in \mathbb{R}^{N \times (p+1)}$$

lässt sich die Beziehung zwischen X und Y für alle N Beobachtungen in Matrixform aufschreiben:

$$Y = X\bar{\beta} + \bar{\varepsilon}.$$

Bei der Schätzung der Regressionskoeffizientenvektors $\bar{\beta}$ wird wieder die Quadratsumme der Residuen nach der Methode der kleinsten Quadrate minimiert. Vorausgesetzt es gibt genügend Beobachtungen (das heißt $\text{Rang}(X) \geq p + 1$), dann erhält man als Lösung des Problems den Vektor der geschätzten Regressionskoeffizienten

$$\bar{\beta} = (X^T X)^{-1} X^T Y. \quad (3.9)$$

Für eine neue Eingabe $\hat{X} = (x_1, \dots, x_p)$ kann der Wert für y mit \hat{y} geschätzt werden durch

$$\hat{y} = \hat{X}\bar{\beta}. \quad (3.10)$$

Im Kontext des Lombard-Filters entspricht \hat{X} als unabhängige Variable einem Lombard-Koeffizientenvektor ($C^{Lombard}$) und y einer Komponente (C_n^{clean}) des entsprechenden “sauberen” Koeffizientenvektors. $\hat{\beta}$ wird separat für jede Komponente von C^{clean} berechnet und entspricht $L(\cdot, k)$ zusammen mit $M(\cdot)$ in Formel 3.8.

3.3.3. Zusammenfassung

Chi and Oh [1996] schlugen ein Modell zur Reduktion des Lombard-Effektes auf Merkmalsebene vor. Mit Mitteln der Regressionsanalyse wird der “Abstand” zwischen den Lombard-Merkmalen und akustischen Modellen, die mit “sauberen” Sprachmerkmalen trainiert wurden, reduziert.

Durch das Aufstellen eines Degradierungsmodells wurde gezeigt, dass Merkmalsvektoren der “sauberen” Sprache von denen der Lombard-Sprache im Cepstralbereich linear abhängig sind. Zur Auffindung dieser Beziehung wurde die multilineare Regression herangezogen. Dabei wurde jede Komponente eines “sauberen”-Vektors von allen Komponenten eines zum selben Phonem (oder Wort) gehörenden Lombard-Vektors abhängig gemacht.

Um die Methode anzuwenden, ist es also notwendig das Lombard-Filter mit einer Menge “Beobachtungen”, also mit Paaren von passenden Merkmalsvektoren, zu trainieren. Passende Merkmalspaare ergeben sich aus den Mengen von Merkmalen der “sauberen” und der Lombard-Sprache, die zum selben Phonem in zwei Äußerungen mit derselben Transkription gehören.

Um die Adaption anzuwenden, genügt es die Formel 3.10 auf jeden (statischen) Merkmalsvektor der Lombard-Sprache anzuwenden.

3.4. Zero-Crossings with Peak Amplitudes

Zero-Crossings with Peak Amplitudes, ZCPA (dt. etwa Nulldurchgänge mit Amplituden-Peaks) ist ein neben MFCC (vgl. Abschnitt 2.8) ein Merkmals-

extraktionsverfahren. Es wurde insbesondere dafür entwickelt, um möglichst lärmbeständige Sprachmerkmale zu extrahieren [Park et al., 2007]. Ursprünglich von Kim et al. [1999] vorgestellt, wurde es von Jung et al. [2007] auch an die Aufnahmecharakteristik des Kehlkopfmikrofons angepasst.

Die Autoren entwickelten ein *auditives* Modell zur Extraktion von Sprachmerkmalen in Anlehnung an die Funktionsweise des menschlichen Gehörs. In Kim et al. [1999] wurde die Robustheit des Verfahrens in Umgebungen mit unterschiedlichen Geräuschtypen gezeigt.

Der prinzipielle Aufbau des Modells ist in Abbildung 3.1 gezeigt. Es besteht

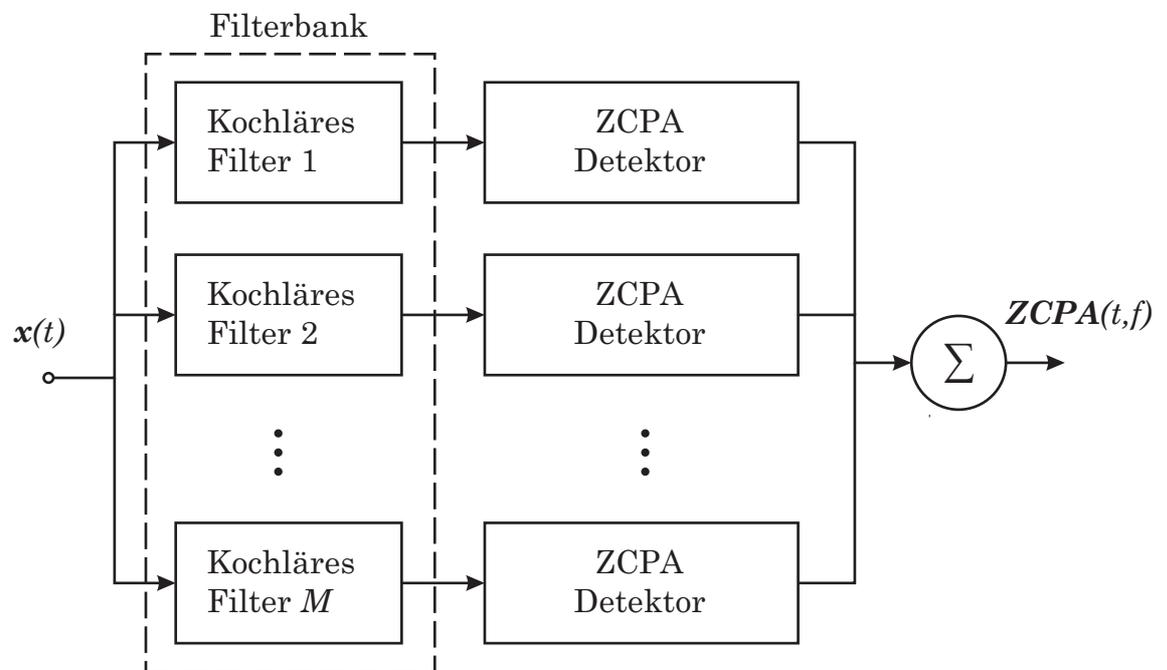


Abbildung 3.1.: Aufbau des ZCPA-Modells

aus einer Menge von Bandpassfiltern und Zero-Crossings- und Amplituden-Peaks-Detektoren. Die sogenannten *kochlären* Filter (lat. *cochlea*, zu dt. — Hörschnecke) teilen das Signal in mehrere Frequenzbänder auf, deren Zentrumsfrequenzen auf einer nichtlinearen Skala liegen, ähnlich wie bei einer Mel-Filterbank (vgl. Abschnitt 2.8.3). Die Skala simuliert die Empfind-

lichkeit der Hörschnecke des menschlichen Hörapparats. Die Detektoren registrieren Nulldurchgänge, die mit neuronaler Aktivität in Verbindung gebracht werden. Die Amplituden-Peaks werden im Modell mit der Stärke des Nervenreizes assoziiert. Das Modell ist biologisch und physiologisch motiviert. Weitere Einzelheiten zu diesem Modell können in Kim et al. [1999] nachgeschlagen werden.

Die kochläre Filterbank besteht aus $N = 16$ überlappenden Bandpassfiltern mit den Zentrumsfrequenzen F_n , die auf einer nichtlinearen Skala liegen:

$$F = 165,4 \cdot (10^{2,1x} - 1), \quad x \in [0, 1] \text{ und } n = 1, 2, \dots, N, \quad (3.11)$$

bei äquidistanter Aufteilung des Intervalls $[0, 1]$ für x .

Diese Formel rührt von einer psycho-akustischen Studie her, die von Greenwood [1990] beschrieben wurde, und simuliert das Auflösungsvermögen der Basilar-Membran im Mittelohr. Die Bandbreiten wurden proportional zu Äquivalentrechteck-Bandbreite gewählt (vgl. Moore and Glasberg [1983]).

Die Ausgabe aus jedem Kanal der Filterbank passiert als nächstes einen Zero-Crossings- und Amplituden-Peaks-Detektor (s. Abbildung 3.2).

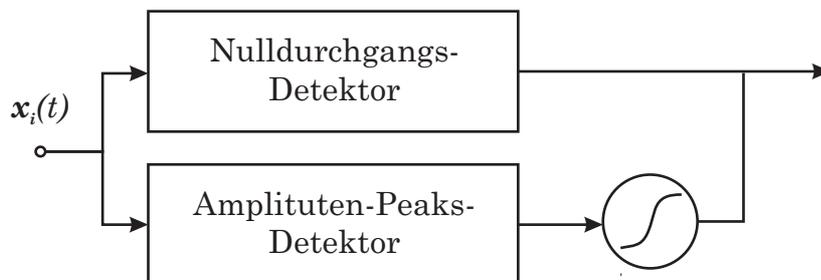


Abbildung 3.2.: ZCPA-Detektor

Der Detektor registriert Nulldurchgänge bei aufsteigender Flanke und gibt zu diesen Zeitpunkten die Peak-Amplitude zwischen den zwei letzten Durchgängen aus, zu allen anderen Zeitpunkten gibt er Null aus (vgl. Abbildung 3.3). Das Ergebnis ist ein ZCPA-kodiertes Signal.

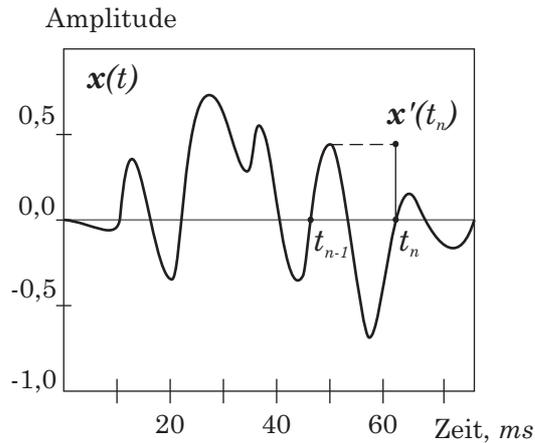


Abbildung 3.3.: ZCPA-Kodierung

Wichtig ist, dass die Zeit zwischen zwei Nulldurchgängen t_n und t_{n-1} zu diesem kurzen Ausschnitt die Frequenzinformation enthält. Die Frequenzhöhe ist gleich

$$f(t_n) = \frac{1}{t_n - t_{n-1}}. \quad (3.12)$$

Die Peak-Amplitude wird mit der Intensität der entsprechenden Frequenz in Verbindung gebracht. Die Amplitude wird zur Gewichtung der Frequenzen im nächsten Schritt verwendet.

Weiter werden Informationen aus allen Kanälen für ein Zeitfenster in einem Frequenzhistogramm mit einer festen Anzahl Frequenzintervalle (engl. *frequency bins*) akkumuliert, um letztendlich einen ZCPA-Merkmalvektor zu ermitteln. Die Aufteilung der gesamten Bandbreite in Frequenzintervalle und deren Anzahl ist wieder psycho-akustisch motiviert und richtet sich nach der Bark-Skala (vgl. oben Abschnitt 2.8.3). Die Bark-Skala ist durch

$$Bark = 13 \cdot \operatorname{atan}(0,00076 \cdot f) + 3,5 \cdot \operatorname{atan}\left(\frac{f^2}{7500^2}\right) \quad (3.13)$$

definiert, wobei f die Frequenz darstellt.

Sei k der Filterindex, Z_k die Anzahl aller Nulldurchgänge in einem Analysefenster zur Zeit t und $P_{k\ell}$ die Peak-Amplitude zwischen zwei aufeinanderfolgenden Nulldurchgängen, dann ist ein ZCPA-kodiertes Merkmal $y(t, i)$ zur Zeit t gegeben durch

$$y(t, i) = \sum_{k=1}^{N_{ch}} \sum_{\ell=1}^{Z_k-1} \delta_{ij\ell} g(P_{k\ell}), \quad 1 \leq i \leq N, \quad (3.14)$$

wobei N die Anzahl der Frequenzintervalle und δ_{ij} die Kronecker-Delta-Funktion ist. Diese ist definiert durch

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & \text{sonst.} \end{cases}$$

Der Index des Frequenzintervalls j_ℓ wird aus der Frequenz in Formel 3.12 ermittelt. $g(\cdot)$ ist eine monotone Funktion, die einen nichtlinearen Zusammenhang zwischen der Amplitude und deren Wahrnehmung im Gehör simuliert. Als eine geeignete Funktion geben die Autoren die Logarithmus-Funktion, $g(x) = \log(1 + x)$, an.

Jung et al. [2007] passten das Verfahren an die Aufnahmecharakteristik des Kehlkopfmikrofons an, indem sie die Fensterlänge, die vom Filterindex abhängig ist, vergrößerten, um den Frequenzen in niedrigen Frequenzbändern mehr Gewicht zu geben. Gleichzeitig schränkten sie die Anzahl der Filter der kochlären Filterbank zu höheren Frequenzen hin ein. Das spiegelt die Beschränktheit der Bandbreite in einem Kehlkopfmikrofonsignal wider.

3.4.1. Zusammenfassung

Zero-Crossings with Peak Amplitudes (ZCPA) ist ein Merkmalsextraktionsverfahren, das zur Gewinnung robuster Merkmale gegenüber Hintergrundgeräuschen entwickelt wurde. Die Grundlage stellt ein Modell dar, das stark biologisch motiviert ist. Das Verfahren erzeugt ein Quasi-Spektrum eines

Sprachsignals, indem es mehrere Frequenzbänder parallel analysiert und in jedem Band die Frequenz und deren Intensität ermittelt. Die Ergebnisse werden über alle Frequenzbänder fensterweise in einem Histogramm aggregiert. Ein auf die Weise gewonnenes Histogramm stellt einen ZCPA-Merkmalvektor dar.

Kapitel 4.

Evaluationsumgebung

In diesem Kapitel wird die in der Arbeit verwendete Sprachdatenbank vorgestellt. Im zweiten Abschnitt wird eine Signalanalyse beschrieben, in der einige wichtigen Indizien für das Auftreten des Lombard-Effekts aufgezeigt werde. Auf diese Analyse stützt sich die Wahl der Verfahren.

4.1. Sprachdatenbank

Diese Diplomarbeit entstand in Zusammenarbeit mit der Abteilung NetMedia des Fraunhofer-Instituts für Intelligente Analyse- und Informationssysteme¹ (Fraunhofer IAIS). NetMedia ist am EU-Projekt *MoveOn*² zur Entwicklung einer Kommunikationsschnittstelle für Motorradfahrer beteiligt. Diese soll dem Fahrer während der Fahrt unter anderem erlauben, den Bordcomputer mit Sprache zu steuern. Es werden dabei mehrere Sensoren (Nahsprech- und Kehlkopfmikrofone) als Aufnahmequellen eingesetzt, um Sprachbefehle aufzunehmen.

Im Rahmen des Projekts wurde eine umfangreiche Sprach- und Geräuschdatenbank, der *MoveOn Korpus*, aufgenommen [Winkler et al., 2008]. Diese beinhaltet Befehlsphrasen und anwendungsspezifische Einzelbefehle aus einem Vokabular von etwa 130 Wörtern in englischer Sprache (britisches Eng-

¹<http://www.iais.fraunhofer.de>

²<http://www.m0ve0n.net>

lich), die *Command&Control*-Phrasen genannt werden. Zusätzlich wurden phonetisch reichhaltige Sätze aufgenommen, das heißt Sätze, die phonetisch ausbalanciert sind und somit alle Phoneme und eine Vielzahl von Phonemkombinationen abdecken.

Insgesamt standen ca. zwölf Stunden reiner und geräuschbelasteter Sprachdaten zur Verfügung. Reine Aufnahmen wurden in einer Studioumgebung und geräuschbelastete unterwegs auf einem Motorrad in der Stadt sowie auf Landstraßen und Autobahnen aufgenommen. Die Daten wurden simultan mit einem Kehlkopfmikrofon und zwei Nahsprechmikrofonen, die links und rechts vom Mund in einen Motorradhelm eingebaut waren, aufgenommen.

Die Sprachdaten sind transkribiert. Die Aufnahmesitzungen, die in der Regel eine bis anderthalb Stunden dauerten, wurden zwecks leichter Handhabung in kurze *Segmente* mit nur einem Satz oder nur einer Phrase (auch *Äußerungen* genannt) unterteilt. Die Segmente haben eine typische Dauer von fünf bis zehn Sekunden. Sie liegen in folgendem Format vor:

Tabelle 4.1.: Kodierungsparameter Sprachsegmente

Parameter	Wert
Abtastfrequenz (sample rate)	16 kHz
Quantisierung	16 Bit
Format	RIFF WAVE

Die oben erwähnten *Command&Control*-Äußerungen sind nach einem festen Schema aufgebaut. Dieses Schema ist durch einen Satz Grammatikregeln in Backus-Naur-Form, der Anwendungsgrammatik (engl. *tast grammar*) definiert. Beispiele solcher Befehle sind: “BIKE CAM START VIDEO” oder “NAVIGATION OPEN”, ein Befehl, mit dem der Nutzer eine Navigationsanwendung aufrufen kann. Die Definition aller Produktionsregeln der Anwendungsgrammatik findet sich im Anhang A.1. Der Anteil der Äußerungen, die diesem Schema entsprechen, beträgt ca. 43% der rund 10.000 Äußerungen. Nur gültige *Command&Control*-Äußerungen können von dem in der vorliegenden Arbeit verwendeten Spracherkennungssystem

erkannt werden. Alle Testmengen enthalten deshalb nur solche Äußerungen.

4.2. Signalanalyse

In diesem Abschnitt erfolgt eine Analyse der Signale im MoveOn-Korpus, die Anhaltspunkte für die Wahl der Aufbereitungsverfahren für Sprachsignale liefert.

4.2.1. Vergleich von Signal und Störung

Eine wichtige Charakteristik eines Audiosignals (Sprache oder Musik), das durch Anwesenheit eines oder mehrerer Hintergrundgeräusche belastet ist, ist der Signal-zu-Rausch-Abstand (engl. *signal to noise ratio*, SNR). SNR ist definiert als Verhältnis der Energie des Nutzsignals (E_{Signal}) zu Energie des Rauschens ($E_{Rauschen}$) und wird typischerweise in Dezibel (dB) gemessen:

$$SNR = 10 \cdot \log \frac{E_{Signal}}{E_{Rauschen}} [dB]$$

Der Signal-zu-Rausch-Abstand charakterisiert die Qualität des Signals oder umgekehrt den Grad der "Verrauschtheit". Bei einem SNR von 30 dB und höher hört sich ein Signal praktisch rauschfrei an. Bei 0 dB haben das Signal und das Rauschen gleiche Lautstärke [Benesty et al., 2008].

Die obige Definition setzt voraus, dass das Nutzsignal und das Rauschsignal separat gemessen werden können. Wurden sie bereits gemischt aufgenommen (wie im Fall der MoveOn-Datenbank), kann man den Wert des wahren SNR nur schätzen. Zur Schätzung wurde ein standardisiertes Tool aus dem Paket zur Qualitätssicherung der Sprache vom National Institute of Standards and Technology, NIST verwendet (s. Wierzynski and Fiscus).

4.2.2. Motivation zum Lombard-Filter

Der Lombard-Effekt, der unter Einfluss von Hintergrundlärm oder Stress entsteht, hat einen negativen Einfluss auf die Erkennungsleistung eines Spracherkennungssystems.

Der Effekt hat mehrere Auswirkungen auf das Sprachsignal. Die Erhöhung des Pegels des Hintergrundrauschens ruft eine Erhöhung der Lautstärke der Sprache hervor. In Abbildung 4.1 ist die Abhängigkeit der Lautstärke (genauer des Schalldruckpegels) der Sprache von der des Hintergrundrauschens gezeigt. Es ist eine Korrelation der beiden Kenngrößen erkennbar. Dies deutet auf die Präsenz des Effektes in den Sprachdaten hin.

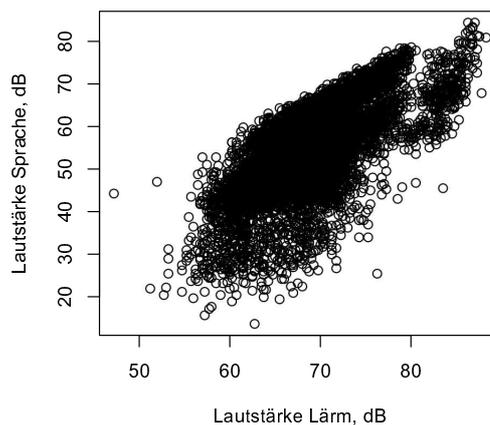
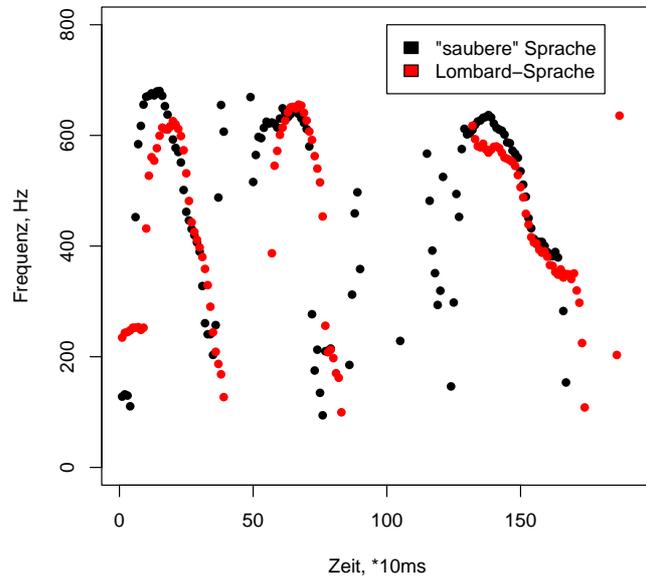
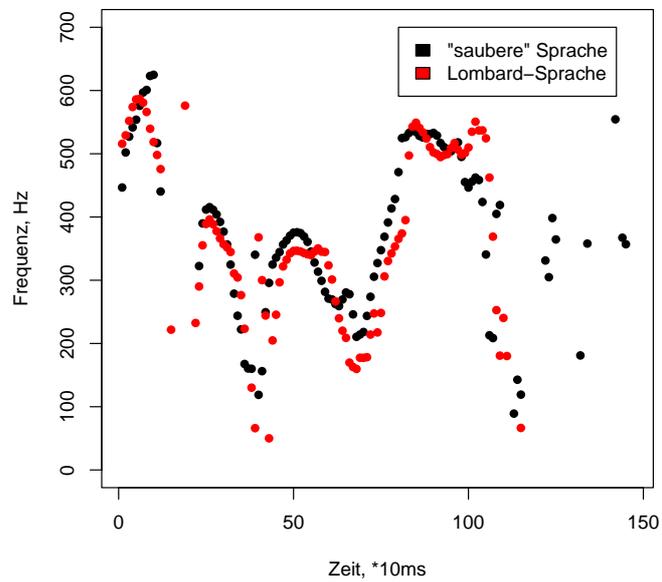


Abbildung 4.1.: Korrelation zwischen der Lautstärke der Sprache und der des Hintergrundlärms.

Eine andere Auswirkung des Effektes ist die Variation der Formantenhöhen in einem Signal. Neben auditiv wahrnehmbaren Veränderungen zeigen Grafiken in Abbildungen 4.2(a) und 4.2(b) Indizien für das Auftreten dieses Effektes. Sie zeigen Verläufe des ersten und wichtigsten Formanten F_1 für jeweils ein Paar von Sprachsegmente mit gleicher Transkription und vom gleichen Sprecher. Bei der Aussprache gleicher Laute variiert die Höhe der gezeigten Kenngröße deutlich.



(a) "NOTEPAD CLOSE"



(b) "TRAFFIC IS LIGHT"

Abbildung 4.2.: Verschiebungen des ersten Formanten F_1 in Lombard-Sprache (rot) gegenüber "sauberer" Sprache (schwarz). Angegeben sind die Transkriptionen bei jeder Grafik.

4.2.3. Motivation zum ZCPA-Verfahren

Die Wahl des ZCPA-Verfahrens ist durch die in Park et al. [2007] gezeigte Robustheit zu Hintergrundgeräuschen begründet. Von dem Verfahren ist bekannt, besonders geräuschbeständige Merkmale extrahieren zu können. Den Grad der Verrauschtheit der Daten kann mit dem Signal-zu-Rausch-Abstand charakterisiert werden. Bei einem SNR von ca. 30 dB und weniger spricht man von geräuschbelasteten Daten. Die Verteilung des SNR in allen Kehlkopfmikrofondaten im Vergleich zu Nahsprechmikrofondaten ist in Abbildung 4.3 gezeigt. Der größere Anteil der Nahsprechmikrofondaten kann als geräuschbelastet betrachtet werden. Auch wenn der SNR der Kehlkopfmikrofondaten deutlich höher ist, ist das Signal nicht rauschfrei.

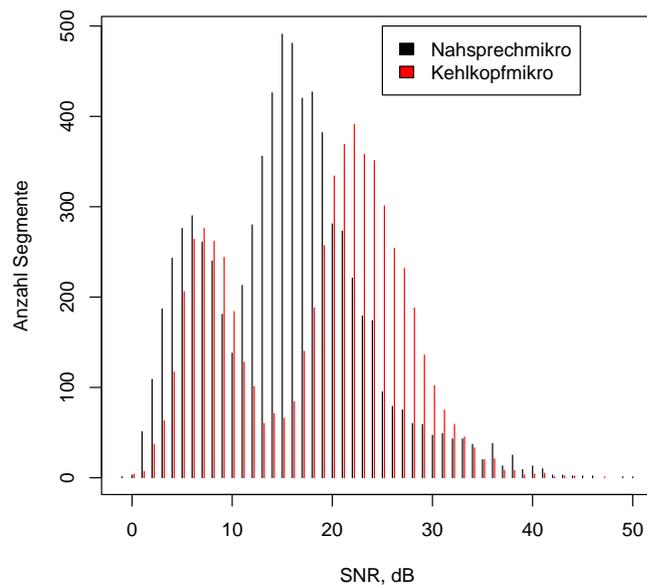


Abbildung 4.3.: SNR-Verteilungen für Nahsprech- und Kehlkopfmikrofondaten. Bedingt durch häufigere Ausfälle des Kehlkopfmikrofondats (oder dessen Aufnahmegeräts) existieren von diesem insgesamt weniger Segmente (6063) als vom Nahsprechmikrofon (7313). Das spiegelt den tieferen Verlauf des entsprechenden Histogramms wider.

Eine Adaptation des ZCPA-Verfahrens nach Park et al. [2007] an die Charakteristik des Kehlkopfmikrofons verspricht eine Verbesserung des Verfahrens speziell für dieses Mikrofon. Entsprechend ist das Verfahren nach Park et al. [2007] für diese Arbeit von besonderem Interesse.

Kapitel 5.

Evaluierung der Verfahren

Im vorliegenden Kapitel wird der experimentelle Teil der Arbeit dargestellt. Zuerst wird der Aufbau der Testumgebung für alle drei zu evaluierenden Verfahren beschrieben. Die Evaluierung selbst wird in mehreren Phasen durchgeführt, deren Ergebnisse hier präsentiert werden. Anschließend werden diese analysiert und beurteilt.

5.1. Aufbau der Evaluationsumgebung

Gegenstand der Evaluation sind drei Verfahren zur Merkmalsextraktion und Adaption für Spracherkennung, die in Kapitel 2 und Kapitel 3 vorgestellt wurden. Im Einzelnen werden im Rahmen dieser Arbeit Mel-Frequency Cepstral Coefficients (MFCC), das Lombard-Filter, das auf dem MFCC-Verfahren aufbaut und Zero-Crossings with Peak Amplitudes (ZCPA) evaluiert. Die Auswahl der Verfahren wurde im vorherigen Kapitel begründet.

Zu Evaluierungszwecken wurde eine Testumgebung mit Hilfe der HTK-Tools aufgebaut [Young et al., 2006]. Das Toolkit enthält Werkzeuge zum Training akustischer Modelle, zur Spracherkennung und zur Evaluation der Erkennungsleistung. Zusätzlich wurden im Rahmen dieser Arbeit das Lombard-Filter und das ZCPA-Verfahren in der Programmiersprache *Perl*¹

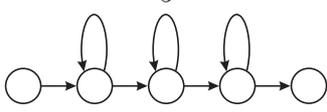
¹<http://www.perl.org>

implementiert. Im Folgenden wird die aufgebaute Umgebung als System referenziert.

5.1.1. Konfigurationsparameter

Für die evaluierten Verfahren wurden die meisten Parameter nach allgemein empfohlenen Erfahrungswerten festgelegt (s. Tabelle 2.1 in Kapitel 3). Die beiden Merkmalsextraktionsverfahren verwenden zur Fensterung die Hamming-Funktion mit Fensterlänge 25 ms. Die Verschiebung des Fensters beträgt 10 ms, was einer Fensterüberlappung von 15 ms entspricht. Sowohl MFCC als auch ZCPA extrahieren aus jedem Fenster sechzehn statische Merkmalskoeffizienten und jeweils gleiche Anzahl dynamischer Koeffizienten erster und zweiter Ordnung. Das Lombard-Filter verwendet wie in Abschnitt 3.3 beschrieben Trainingspaare um eine Adaptionmatrix der Größe 16×17 zu bestimmen. Eine Übersicht der Konfigurationsparameter findet sich in Tabelle 5.1.

Tabelle 5.1.: Allgemeine Konfigurationsparameter

Parameter	Wert
Merkmalsextraktion	
Anzahl statische Koeffizienten	16
Anzahl Delta-Koeffizienten	16
Anzahl Delta-von-Delta-Koeffizienten	16
Merkmalsvektorgroße	48
Analysefensterlänge	25ms
Fensterverschiebung	10ms
Fensterfunktion	Hamming
akustische Modelle	
Anzahl HMMs	46
Anzahl Zustände	5
Topologie	 <pre> graph LR S(()) --> Z1(()) Z1 --> Z1 Z1 --> Z2(()) Z2 --> Z2 Z2 --> Z3(()) Z3 --> Z3 Z3 --> E(()) </pre>
	Phonemebene-HMMs

Die Topologie der akustischen Modelle, die auf Hidden-Markov-Modellen basieren, wurde nach Empfehlungen in Young et al. [2006] gewählt. Jedem Phonem der englischen Sprache ist jeweils ein HMM zugeordnet. Die Liste aller Phoneme ist in Tabelle A.1 im Anhang aufgeführt. Jedes HMM besteht aus fünf Zuständen und weist eine lineare links-rechts Topologie auf. Diese Topologie schreibt vor, dass die Kanten eines jeden Zustands (außer Endzustand) entweder nur zum direkten Folgezustand gehen oder selbstinduziert sind. Die Übergangs- und Beobachtungswahrscheinlichkeiten sind Parameter, die während der Trainingsphase der HMMs bestimmt werden. Die Beobachtungswahrscheinlichkeiten sind als kontinuierliche Gaußsche Dichtefunktionen durch einen Mittelwert und eine Varianz gegeben. In Tabelle 5.1 sind die wichtigsten Konfigurationsparameter und die Topologie akustischer Hidden-Markov-Modelle aufgeführt.

5.1.2. Phasen der Evaluierung

Da die Verfahren auf unterschiedliche Aspekte der robusten Spracherkennung eingehen, wurde deren Evaluierung in jeweils zwei Phasen aufgeteilt.

Der Lombard-Effekt hängt, wie bereits erwähnt, sowohl vom Sprecher als auch von der Art des Lärms und dessen Pegel ab. In der ersten Phase wird zunächst eine sprecherübergreifende Auswertung durchgeführt, die allgemeine Auswirkung des Lombard-Filters aufzeigt. Sie basiert auf der gesamten Sprachdatenbank. In der zweiten Phase wird das Verfahren sprecherbezogen ausgewertet, um der Sprecherabhängigkeit des Effekts gerecht zu werden. Das System wird dabei bezüglich eines jeden der fünfzehn Sprecher einzeln ausgewertet und die Ergebnisse anschließend gemittelt. In beiden Phasen werden zum Training akustischer Modelle geräuschfreie bis geräuscharme (“saubere”) Segmente, wo kein Lombard-Effekt angenommen wird, und zum Training des Lombard-Filters Paare “sauberer” Segmente und Lombard-Segmente verwendet. Lombard-Segmente sind Segmente, die unter starker Geräuschbelastung aufgenommen wurden und deshalb Aus-

wirkungen des Lombard-Effekts zu erwarten sind. Der Grad der Geräuschbelastung wird anhand des Signal-zu-Rausch-Abstandes (SNR) (vgl. Abschnitt 4.2) beurteilt. “Saubere” Daten werden an die Bedingung $\text{SNR} \geq 20$ dB gekoppelt. Für Lombard-Daten wurde hingegen ein $\text{SNR} \leq 7$ dB angenommen. Beide Schwellenwerte sind Erfahrungsgrößen.

Das zweite Verfahren Zero-Crossings with Peak Amplitudes dient der Extraktion robuster Sprachmerkmale gegenüber Hintergrundgeräuschen. Die Robustheit der Merkmale bezüglich der MoveOn-Daten soll hier evaluiert werden. Die Auswertung wird im Folgenden auch hier in zwei Phasen aufgeteilt. In der ersten Phase erfolgt eine datenübergreifende Evaluation, die die ganze Sprachdatenbank umfasst. Dagegen wird in der zweiten Phase das Verfahren bezüglich unterschiedlich stark ausgeprägter Geräuschbelastung des Sprachsignals ausgewertet.

5.2. Auswertung des Lombard-Filters

Ziel der ersten Phase ist es, die allgemeine Auswirkung des Lombard-Filters auf die Erkennungsleistung des Systems bezüglich der gesamten Sprachdatenbank zu untersuchen. Zu diesem Zweck wurden akustische Modelle mit gering verrauschten (“sauberen”) Beispielen trainiert. Bei den “sauberen” Beispielen wird angenommen, dass diese nicht durch den Lombard-Effekt beeinflusst sind. Die Angaben zum Training akustischer Modelle und des Lombard-Filters sind in Tabelle 5.2 gezeigt. Die Trainingsmenge umfasst 3224 Sprachsegmente. Das Lombard-Filter wurde mit 4352 Paaren von Cepstral-Koeffizientenvektoren aus 98 Trainingssegmenten ebenfalls trainiert. Danach wurden dem System Segmente mit einem $\text{SNR} \leq 7$ dB (in denen der Lombard-Effekt vermutet werden kann) präsentiert. Nach einer Adaption an den Lombard-Effekt wurden dieselben Segmente dem System noch einmal präsentiert. Die Evaluationsergebnisse der ersten Phase sind in Tabelle 5.3 zusammengefasst.

Die Evaluierung zeigt, dass die Wortakkuratheit auf dem Testdatensatz vor der Adaption bei 78,40% liegt und nach der Adaption um 7,61% auf 70,79%

Tabelle 5.2.: Auswahl von Training- und Testdaten für die Evaluierung des Lombard-Filters (sprecherübergreifend)

Analyseverfahren	MFCC	MFCC mit Lombard-Filter
Trainingsdaten		
<u>Akustische Modelle</u>		
Auswahlbedingungen	Alle Sprecher, SNR \geq 20 dB	Alle Sprecher, SNR \geq 20 dB
Anzahl Segmente	3224	3224
<u>Adaption</u>		
Auswahlbedingungen	—	Paare mit (SNR \leq 7 dB, SNR \geq 20 dB)
Anzahl Paare von Segmenten	—	98
Anzahl Paare von Cepstral-Koeffizienten	—	4352
Testdaten		
Auswahlbedingungen	Alle Sprecher, SNR \leq 7 dB, Command&Control	Alle Sprecher, SNR \leq 7 dB, Command&Control
Anzahl Segmente	258	258

Tabelle 5.3.: Ergebnisse der Evaluierung des Lombard-Filters (sprecherübergreifend)

Analyseverfahren	MFCC	MFCC mit Lombard-Filter
Worterkennungsrate, %	79,65	72,03
Wortakkuratheit, %	78,40	70,79

absinkt. Die Wortakkuratheit weist etwa den gleichen Trend auf.

Die zweite Phase geht auf die Sprecherabhängigkeit des Lombard-Effekts ein. Hierin werden für jeden der fünfzehn Sprecher jeweils ein Satz bestehend aus 46 akustischen Modellen auf Phonembasis und einem Lombard-Filter trainiert, indem die Trainings- und die Testmengen auf einen einzelnen Sprecher eingeschränkt werden. Die Modelle wurden mit bis zu 259 Segmenten und die Lombard-Filter mit bis zu 1764 MFCC-Paaren trainiert. Die Einzelheiten zu Trainingsdaten sind in Tabelle 5.4 aufgeführt. Evaluiert wurde das System mit jeweils sechs bis zehn Segmenten. Die Ergebnisse für einzelne Sprecher und die Durchschnittswerte sind in Tabelle 5.5 zusammengefasst.

Tabelle 5.4.: Beschreibung der Trainingsdaten für das Lombard-Filter (sprecherspezifisch)

Sprecher	Training akustischer Modelle	Training Lombard-Filter		
	Anzahl Segmente	Anzahl Segmentepaare	Seg-Paare	Anzahl MFCC-Paare
m1405	70	6		841
m1646	41	3		248
m1875	80	8		809
m1908	259	10		2006
m2471	54	8		1222
m2776	46	7		1058
m4479	47	3		432
m5177	46	5		854
m5541	74	6		1097
m6988	43	3		221
m6991	112	8		1331
m8571	43	3		357
m9441	112	5		887
m9742	101	10		1764
m9969	40	4		628
Durchschnitt	77,87	5,93		917

In Abbildung 5.2 sind die Wortakkuratheiten mit und ohne Lombard-Filter für verschiedene Sprecher aufgezeigt. Die erste auffallende Beobachtung dieser Auswertung ist, dass unabhängig vom Lombard-Filter die Erkennungs-

Tabelle 5.5.: Ergebnisse der Evaluierung für das Lombard-Filter (sprecherspezifisch)

Analyseverfahren		MFCC	MFCC mit Lombard- Filter
Sprecher	Anzahl Testsegmente	Wort- akkuratheit,%	Wort- akkuratheit,%
m1405	9	75,00	82,14
m1646	11	2,63	0,00
m1875	14	64,86	51,35
m1908	23	91,25	81,25
m2471	24	6,94	2,78
m2776	21	76,56	73,44
m4479	6	29,41	35,29
m5177	11	44,12	38,24
m5541	13	-5,13	-2,56
m6988	10	47,62	38,10
m6991	22	83,58	74,63
m8571	6	80,00	95,00
m9441	14	93,02	76,74
m9742	10	100,00	100,00
m9969	14	7,84	11,76
Durchschnitt	13,87	53,18	50,54

leistung in der Testumgebung für alle Sprecher großen Schwankungen ausgesetzt ist. Die Kennzahlen für die Wortakkuratheit schwanken zwischen -5,13% und 100,00%. Für schlecht erkannte Sprachsegmente kann man diese Beobachtung bestätigen: die Verständlichkeit der Sprache in diesen Segmenten ist gering entweder aufgrund dominierender Hintergrundgeräusche, geringer Signallautstärke in den Aufnahmen. Möglicherweise ist es auch auf fehlerhafte Transkription zurückzuführen.

Die Ergebnisse zeigen, dass die Anwendung des Lombard-Filters bei einzelnen Sprechern eine signifikante Verbesserung der Erkennungsleistung des Systems mit sich bringt (Die Sprecher sind in der Tabelle 5.5 grau markiert). Für fünf von fünfzehn Sprechern wird die Wortakkuratheit durch die Anwendung des sprecherspezifischen Lombard-Filters um bis zu 15%

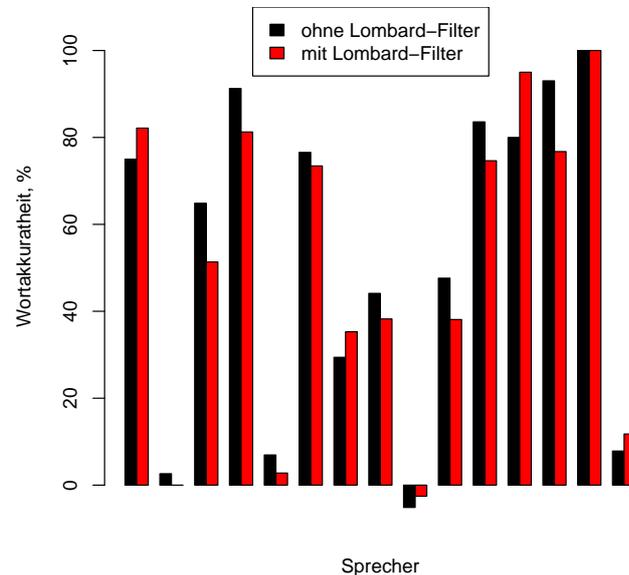


Abbildung 5.1.: Wortakkuratheiten für verschiedene Sprecher mit und ohne Lombard-Filter

verbessert(wie beim Sprecher “m8571”). In den meisten Fällen ist jedoch eine Verschlechterung der Systemleistung festzustellen (s. Abbildung 5.2). Im Mittel sinkt die Wortakkuratheit von 53,18% auf 50,54% nach Anwendung des Lombard-Filters ab, so dass beim Einsatz eines auf einen Sprecher zugeschnittenen Lombard-Filters nur im Einzelfall mit einer Verbesserung gerechnet werden kann. Offensichtlich sind die Auswirkungen des Lombard-Effekts so vielfältig, dass sie nicht alle mit einer linearen Adaptionmethode kompensiert werden können. Möglicherweise stimmt auch die Annahme nicht, dass der Lombard-Effekt überall dort vorhanden ist, wo der Signal-zu-Rausch-Abstand hinreichend klein ist.

In Kapitel 4 wurden Indizien für das Auftreten des Effektes gezeigt, indem für stichprobenartig ausgewählte Paare von Sprachsegmenten mit gleicher Transkription und vom selben Sprecher Verschiebungen und Dehnungen des ersten und wichtigsten Formanten F_1 aufgezeigt wurden. Dies muss

jedoch nicht für alle Aufnahmen stimmen, die unter einem hohen Lärmpegel aufgenommen wurden. Zudem wurden die aufgenommenen Sprachdaten nachgesprochen, so dass es sich bei den Äußerungen um eher überlegte Sprache handelt. Durch das Nachsprechen der Phrasen ist ein gewisser Nachahmungseffekt in der Aussprache zu erwarten, so dass der Lombard-Effekt bei einigen Sprechern eventuell schwächer ausfällt oder erst gar nicht auftritt.

Für weitere Arbeiten in diesem Bereich wird deshalb empfohlen, auf die Vielfältigkeit des betroffenen Effekts besser einzugehen, indem die Adaption komplexer gestaltet wird. Hier könnten Regressionsanalyse höherer Ordnung oder Mapping der Merkmale in Frage kommen. Zu dem wären Arbeiten zur automatischen Detektion des Lombard-Effekts interessant, um verlässigere Aussagen zum Auftreten des Effekts treffen zu können.

5.3. Auswertung des ZCPA-Verfahrens

Ziel der ersten Phase ist es, das ZCPA-Verfahren datenübergreifend zu evaluieren. Zu diesem Zweck werden akustische Modelle mit allen Sprechern und unter diversen Umgebungsbedingungen trainiert (multikonditionales Training). Die Anzahl der Trainingssegmente beträgt 5669. Die Testdaten beinhalten 142 Segmente, die ebenfalls bei unterschiedlichen Geräuschbelastungen aufgenommen wurden. Das entspricht etwa dem Fall einer praxisnahen Anwendung. Die Ergebnisse sind in Tabelle 5.6 zusammengefasst. Sie zeigen, dass die Erkennungsleistung des MFCC-Verfahrens im Vergleich zum ZCPA deutlich besser ausfällt. Die Wortakkuratheit des ersten übertrifft mit 85,65% den letzteren mit 53,01% um über 32,64%.

In der zweiten Phase wird das ZCPA-Verfahren bezüglich der Robustheit zu Hintergrundgeräuschen getestet. Da das System in einer realen Anwendung nicht alle Umgebungsbedingungen berücksichtigen kann, werden akustische Modelle hier nur mit "sauberen" Segmenten mit einem Signal-zu-Rausch-Abstand von 25 dB und mehr trainiert. Die Testdaten werden bezüglich ihrer Belastung durch Hintergrundgeräusche mit einer Schrittweite von 5

Tabelle 5.6.: Ergebnisse der Evaluierung des ZCPA-Verfahrens (datenübergreifend)

Analyseverfahren	MFCC	ZCPA
Anzahl Segmente	142	142
Worterkennungsrate, %	87,83	62,96
Wortakkuratheit, %	85,65	53,01
Satzerkennungsrate, %	83,80	54,93

dB partitioniert. Die Ergebnisse der Evaluierung sind in Tabelle 5.7 präsentiert.

Tabelle 5.7.: Ergebnisse der Evaluierung des ZCPA-Verfahrens (SNR-spezifisch)

Analyseverfahren		MFCC		ZCPA	
Auswahlbedingung für Testsegmente	Anzahl Segmente	Wortakkuratheit, %	Worterkennungsrate, %	Wortakkuratheit, %	Worterkennungsrate, %
$\text{SNR} \geq 25$ dB	53	80,82	84,25	56,16	63,70
$20 \leq \text{SNR} < 25$ dB	64	82,05	82,56	57,44	63,08
$15 \leq \text{SNR} < 20$ dB	27	80,49	81,71	60,69	64,63
$10 \leq \text{SNR} < 15$ dB	18	77,36	77,36	49,06	58,49
$5 \leq \text{SNR} < 10$ dB	59	70,62	72,68	22,16	35,05
$\text{SNR} < 5$ dB	12	62,79	65,12	48,84	55,81
Durchschnitt	38,83	75,69	77,28	49,06	56,70

Die Auswertung zeigt, dass sich das ZCPA-Verfahren auf Kehlkopfmikrofonsignalen wenig robust gegenüber Umgebungsgeräuschen auf der verwendeten MoveOn-Datenbank verhält. Mit sinkendem Signal-zu-Rausch-Abstand geht die Wortakkuratheit des Systems von 56,16% auf 22,16% zurück. Unter gleichen Bedingungen zeigt sich das MFCC-Verfahren sowohl leistungsfähiger als auch robuster. Die Erkennungsleistung sinkt hier nur langsam von 80,82% auf 62,79% ab. Dieser Sachverhalt ist in Abbildung 5.2 erkennbar.

Insgesamt ist festzustellen, dass das ZCPA-Verfahren sowohl datenübergreifend als auch SNR-spezifisch dem Referenzverfahren MFCC deutlich unterlegen ist und sich nur bedingt für die meisten Anwendungen eignet.

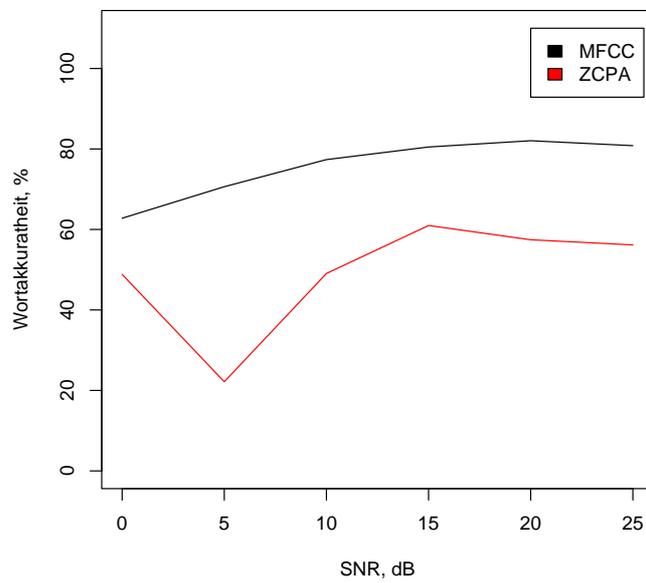


Abbildung 5.2.: Wortakkuratheiten der ZCPA-Verfahrens im Vergleich zu MFCC für verschiedene SNR-Werte.

Die Motivation für die Wahl des ZCPA-Verfahrens war seine Robustheit gegenüber Umgebungsgeräuschen, die in Park et al. [2007] gezeigt wurde. Auf der MoveOn-Sprachdatenbank brachte es die erhoffte Leistung jedoch nicht. Diese Ergebnisse sind auf die unterschiedlichen Rahmenbedingungen zurückzuführen, so wie zum Beispiel Einzelworterkennung in Park et al. [2007] gegenüber der Fließsprache in der MoveOn-Datenbank oder künstlich hinzugefügte Geräusche der Vergleichsstudie gegenüber realen Umgebungsgeräuschen in der vorliegenden Arbeit. Für weitere Untersuchungen wäre interessant, eine Kombination aus MFCC-Merkmalen und ZCPA-Merkmalen auszuprobieren.

Kapitel 6.

Zusammenfassung und Ausblick

Ziel dieser Arbeit war, zwei vielversprechende Verfahren, ein Merkmalsextraktions- und ein Adaptionsverfahren, im Hinblick auf eine robuste Spracherkennung zu evaluieren.

Der Lombard-Effekt, der in der Sprache unter geräuschbelasteten Bedingungen auftritt, hat einen signifikant negativen Einfluss auf die Qualität der automatischen Spracherkennung. Auf Sprachsignalebene spiegelt er sich in Verschiebungen der Formanten, Dehnung der Vokale und Erhöhung der Intensität wider. Ein Verfahren zur Kompensierung des Lombard-Effekts auf der Merkmalsebene, das in Chi and Oh [1996] beschrieben ist, wurde in dieser Arbeit implementiert und evaluiert. Für einige ausgewählte Sprecher war es gelungen, eine Verbesserung der Erkennungsleistung zu erreichen. Eine allgemeinere Verbesserung in einem praktikablen Umfang zu erzielen, war es jedoch nicht möglich.

Ein möglicher Grund ist, dass die Veränderungen des Sprachsignals, die durch den Lombard-Effekt induziert sind, zu vielfältig sind, als dass man sie allein mit einer linearen Adaptation kompensieren kann. Für künftige Arbeiten in diesem Bereich wird deshalb empfohlen, auf die Vielfältigkeit des betroffenen Effekts besser einzugehen, indem die Adaption komplexer gestaltet wird. Hier könnten das Mapping der Merkmale oder eine Regressionsanalyse höherer Ordnung in Frage kommen. Interessant wären auch Arbeiten zur automatischen Detektion des Lombard-Effekts, um verlässigere Aussagen zum Auftreten des Effekts treffen zu können.

Die Auswertung des auditiv motivierten Verfahrens zur Merkmalsextraktion Zero-Crossings with Peak Amplitudes zeigte, dass es wenig robust gegenüber Umgebungsgeräuschen im Vergleich zu MFCC und einer in [Benesty et al., 2008] beschriebenen Vergleichsstudie ist. Aber auch mit der Anpassung an die Aufnahmecharakteristik des Kehlkopfmikrofons, wie sie in Jung et al. [2007] beschrieben ist, liegt die Erkennungsleistung im Durchschnitt bei rund 55%, was sicherlich unzureichend für die meisten Applikationen ist. Das Referenzverfahren, Mel-Frequency Cepstral Analyse, erzielte hingegen eine insgesamt deutlich bessere Erkennungsleistung unter gleichen Bedingungen und zeigte eine gute Robustheit. Interessant wäre an dieser Stelle zu untersuchen, ob die Erkennungsleistung aus Kombination von Merkmalen aus den beiden Verfahren eine Verbesserung bringt.

Anhang A.

Anhang

A.1. MoveOn Anwendungsgrammatik

Eine Anwendungsgrammatik (engl. *task grammar*), wie sie in dieser Arbeit verwendet wurde, besteht aus einer Menge Produktionsregeln in erweiterter Backus-Naur-Form (EBNF). Eine Regel definiert Literale, die mit einem vorangestellten Dollar-Zeichen (\$) gekennzeichnet sind. Ein Literal wird einem Ausdruck zugewiesen, der aus einem oder mehreren anderen Literalen oder Terminalen (Elemente des Lexikons) besteht. Alternativen können in einer Regel durch die Verwendung eines vertikalen Strichs “|” getrennt werden. Eckige Klammern erlauben Angabe optionaler Literale und Terminale. Eine besondere Regel ist die letzte, die definiert, was ein Element der Sprache ist. Jede gültige Äußerung muss demnach aus dieser Regel durch Anwendung einer oder mehreren anderer Regeln ableitbar sein. Eine vollständige Definition der Ableitregeln kann Young et al. [2006] entnommen werden.

Die Anwendungsgrammatik, wie sie mit dem Spracherkenner in dieser Arbeit verwendet wurde, ist unten dargestellt:

```
$activate_command = CLOSE | OPEN;
$ahead_v = AHEAD;
$and_v = AND;
$authorisation_v = AUTHORISATION | AUTHORIZATION;
$book_v = BOOK;
$c_v = C;
$call_v = CALL;
$cam_v = CAM;
$cam_type = BIKE | HELMET;
$change_v = CHANGE;
$channel_v = CHANNEL;
$check_v = CHECK;
$confirm = CONFIRM | RECEIVED | AFFIRMATIVE;
$control_v = CONTROL;
$create_v = CREATE;
$digit = ONE | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT
        | NINE | ZERO | 0;
$deca = TEN | ELEVEN | TWELVE | THIRTEEN | FOURTEEN | FIFTEEN
        | SIXTEEN | SEVENTEEN | EIGHTEEN | NINETEEN;
$decade = TWENTY | THIRTY | FORTY | FIFTY | SIXTY | SEVENTY | EIGHTY
        | NINETY;
$direction = DOWN | UP;
$display_v = DISPLAY;
$do_v = DO;
$folder_v = FOLDER;
$general_command = CANCEL | DENY;
$go_v = GO;
$head_v = HEAD | HEADS;
$helmet_v = HELMET;
$image_v = IMAGE | PICTURE;
$image_command = CAPTURE;
$info = MESSAGE;
$info_command = REPEAT | LAST | NEXT | SKIP;
$intelligence_v = INTELLIGENCE;
$local_v = LOCAL;
$log_v = LOG;
$log_command = UPDATE;
$me_v = ME;
$mute_v = MUTE;
$navigation_v = NAVIGATION;
```

```
$new_v = NEW;
$note_v = NOTE;
$notepad_v = NOTEPAD;
$notepad_command = LAST | NEXT | START | STOP;
$off_v = OFF;
$on_v = ON;
$other_data = INFORMATION;
$playback_v = PLAYBACK | REPLAY;
$position_v = POSITION;
$radio_v = RADIO;
$recent_v = RECENT;
$relay_v = RELAY;
$scroll_v = SCROLL;
$set_v = SET;
$show_v = SHOW;
$sign_v = SIGN;
$spellingletter = ALPHA | BRAVO | CHARLIE | DELTA | ECHO | FOXTROT
                | GOLF | HOTEL | INDIA | JULIET | KILO | LIMA | MIKE | NOVEMBER
                | OSCAR | PAPA | QUEBEC | ROMEO | SIERRA | TANGO | UNIFORM
                | VICTOR | WHISKY | X RAY | YANKEE | ZULU;
$status_v = STATUS;
$status_command = EMERGENCY | PATROL | PURSUIT;
$switch_v = SWITCH;
$stalk_v = TALK;
$through_v = THROUGH;
$to_v = TO;
$up_v = UP;
$user_v = USER;
$video_v = VIDEO;
$video_command = START | STOP;
$volume_v = VOLUME;
$volume_command = DOWN | MUTE | UP;
$will_v = WILL;
$yes_v = YES;
$id_s = I D;
$authorisation_s = $authorisation_v
                 | ($book_v $me_v $on_v $call_v $sign_v);
$confirm_s = [$yes_v] $confirm
            | ([$yes_v] $will_v $do_v)
            | ($go_v $ahead_v);
```

```
$number_s = ($digit [$digit] [$digit]) | $deca | ($decade [$digit]);
$callsign_s = $spellingletter [$spellingletter] $number_s;
$numberplate_s = $spellingletter [$spellingletter] [$spellingletter]
$digit [$digit] [$digit] [$digit] [$spellingletter]
[$spellingletter] [$spellingletter] [$digit] [$digit];
$relay_to_s = [$relay_v] $to_v;
$relay_pos_to_s = [$relay_v] [$position_v] $to_v;
$display_type_s = $display_v | ($head_v $sup_v $display_v)
| ($helmet_v $display_v);
$display_data_s = $image_v | $video_v | $other_data;
$data_relay_s = ($local_v $control_v) | $display_type_s;
$note_relay_s = ($c_v $and_v $c_v $log_v) | ($intelligence_v $log_v);
$playback_s = $playback_v ($recent_v | $number_s | $id_s);
$folder_s = $callsign_s | $number_s | $folder_v;
$channel_s = $number_s | $channel_v;
$auth_s = ($change_v $user_v) | $callsign_s;
$status_s = [$set_v] $status_command;
$radio_s = $activate_command
| ($callsign_s $to_v [$radio_v] $callsign_s)
| ($stalk_v $through_v $callsign_s) | $check_v;
$cam_type_s = $cam_type $cam_v;
$camera_s = ($video_command $video_v)
| ($image_command $image_v)
| $activate_command;
$data_type_s = $image_v | $video_v;
$data_s = $relay_to_s $data_relay_s;
$notepad_s = ($notepad_command $note_v)
| ($relay_to_s $note_relay_s) | ($relay_to_s $data_relay_s)
| $activate_command | $playback_s;
$display_s = ($show_v $display_data_s)
| ($scroll_v $direction)
| $activate_command;
$info_s = $info_command;
$navi_s = ($relay_pos_to_s $data_relay_s) | $activate_command;
$log_s = $log_command | ($create_v $new_v) | $playback_s;
$volume_s = $volume_command;
$general_s = $general_command
| ($switch_v $mute_v)
| ($book_v $me_v $off_v);
$change_s = ($folder_v [$folder_s]) | ($channel_v [$channel_s]);
```

```
( SENT-START
  ( ($authorisation_v $auth_s)
  | ($authorisation_s $callsign_s)
  | ($status_v $status_s)
  | ($radio_v $radio_s)
  | ($cam_type_s $camera_s)
  | ($data_type_s $data_s)
  | ($notepad_v $notepad_s)
  | ($display_type_s $display_s)
  | ($info $info_s)
  | ($navigation_v $navi_s)
  | ($log_v $log_s)
  | ($volume_v $volume_s)
  | $general_s
  | $callsign_s
  | $numberplate_s
  | $confirm_s)
SENT-END )
```

A.2. Phonemalphabet

Tabelle A.1 zeigt die Liste aller Phoneme der englischen Sprache nach dem Standard SAMPA¹ (Speech Assessment Methods Phonetic Alphabet) erstellt wurde. Phoneme werden in der Literatur in der Regel durch Schrägstriche /·/ gekennzeichnet. /sil/ und /sp/ bezeichnen zwei besondere Phoneme: sie entsprechen der Stille (engl. *silence*, “*sil*”) oder einer Pause, respektive einer kurzen Pause (engl. *short pause*, “*sp*”).

¹<http://www.phon.ucl.ac.uk/home/sampa/index.html>

Tabelle A.1.: Liste der Phoneme der englischen Sprache nach dem SAMPA-Standard.

Bezeichnung	Bezeichnung
1. aa	24. oh
2. ah	25. sh
3. ax	26. jh
4. ey	27. z
5. b	28. f
6. ay	29. er
7. d	30. m
8. ih	31. ao
9. ng	32. g
10. l	33. hh
11. t	34. ea
12. iy	35. ow
13. aw	36. w
14. v	37. uh
15. ae	38. dh
16. s	39. th
17. uw	40. ch
18. k	41. ia
19. eh	42. oy
20. p	43. ua
21. n	44. zh
22. y	45. sil
23. r	46. sp

Literaturverzeichnis

Jacob Benesty, M. Mohan Sondhi, and Yiteng Huang, editors. *Springer Handbook of Speech Processing*. Springer-Verlag Berlin Heidelberg, 2008.

Sang-Mun Chi and Yung-Hwan Oh. Lombard effect compensation and noise suppression for noisy lombard speech recognition. *In Proceedings on 4th International Conference on Spoken Language*, 1996.

Michael Clausen and Meinard Müller. *Skript zur Vorlesung: Zeit-Frequenz-Analyse und Wavelettransformationen*. Universität Bonn, 2001.

Stephen Euler. *Grundkurs Spracherkennung*. Vieweg & Sohn Verlag / GWV Fachverlage GmbH, Wiesbaden, 2006.

Gernot A. Fink. *Mustererkennung mit Markov-Modellen*. B.G. Teubner Verlag / GWV Fachverlage GmbH, Wiesbaden, 2003.

D. Greenwood. A cochlear frequency-position function for several species - 29 years later. *The Acoustical Society of America*, 87, 1990.

John H.L. Hansen and Vaishnevi Varadarajan. Analysis and compensation of lombard speech across noise type and levels with application to in-set/out-of-set speaker recognition. *IEEE Transactions on Audio Speech and Language Processing*, 17(2), 2009.

Young-Giu Jung, Mun-Sung Han, and Sang Jo Lee. Development of an optimized feature extraction algorithm for throat signal analysis. *Electronics and Telecommunications Research Institute (ETRI) Journal*, 29 (3), 2007.

- Uwe Kiencke and Holger Jäkel. *Signale und Systeme*. Oldenbourg Verlag München, 2008.
- Doh-Suk Kim, Soo-Young Lee, and Rhee Man Kil. Auditory processing of speech signals for robust speech recognition in real-world noisy environments. *IEEE Transactions on Speech and Audio Processing*, 7(1), 1999.
- David Mansour and Biing Hwang Juang. The short-time modified coherence representation and its application for noisy speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37, 1989.
- B.C. Moore and B.R. Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *The Acoustical Society of America*, 74:750–753, 1983.
- Sang Kyoon Park, Rhee Man Kil, Young-Giu Jung, and Mun-Sung Han. Zero-crossing-based feature extraction for voice command systems using neck-microphones. *Proceedings of the 4th international symposium on Neural Networks: Advances in Neural Networks*, 2007.
- Beat Pfister and Tobias Kaufmann. *Sprachverarbeitung. Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer Verlag, 2008.
- Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1), 1978.
- Ralf Schlüter. *Investigations on Discriminative Training Criteria*. Rheinisch-Westfälische Technische Hochschule Aachen, 2000.
- Ernst Günter Schukat-Talamazzini. *Automatische Spracherkennung. Statistische Verfahren der Musteranalyse*. Vieweg Verlag, 1995.
- Casimir Wierzynski and Jon Fiscus. “*stnr.doc*” – Teil der Pakete NIST *SPeech Quality Assurance (SPQA) Version 2.3 und Speech File Manipu-*

lation Software (SPHERE) Version 2.5. National Institute of Standards and Technology.

Thomas Winkler, Theodoros Kostoulas, Richard Adderley, Christian Bonkowski, Todor Ganchev, Joachim Köhler, and Nikos Fakotakis. *The MoveOn Motorcycle Speech Corpus*. In Proceedings of the 6th International Language Resources and Evaluation (LREC'08), 2008.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Andrew Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Cambridge University Engineering Department, 2006.