# Lippenlesen als Unterstützung zur robusten automatischen Spracherkennung

Diplomarbeit
von
Christoph Bregler

Betreuung:
Prof. Dr. Alex Waibel

Institut für Logik, Komplexität und Deduktionssysteme
Fakultät für Informatik
Universität Karlsruhe
D-76128 Karlsruhe

Ich versichere hiermit, die vorliegende Arbeit selbständig und ohne unzulässige Hilfsmittel angefertigt zu haben. Die verwendeten Quellen sind im Literaturverzeichnis aufgeführt.

Berkeley, den 17. September 1993

# Inhaltsverzeichnis

## 0.1 Danksagung

Diese Arbeit begann im Frühjahr 1992 in Professor Alex Waibels Gruppe an der Universität Karlsruhe und wurde Anfang 1993 am International Computer Science Insitute (ICSI) in Berkeley, U.S.A. in weiterer Zusammenarbeit mit der Karlsruher Gruppe fortgesetzt.

Ich möchte mich besonders bei Alex Waibel bedanken, der mir die Möglichkeit gab, dieses Projekt in seiner Gruppe aufzubauen. Sein Engangement und das außergewöhnlich unkonventionelle Klima in seiner Gruppe wirkte sehr motivierend auf diese Arbeit. Ihm und Jerry Feldman ist für ihre Unterstützung und Offenheit zu danken, die es mir ermöglichte dieses Projekt zusätzlich am ICSI weiterführen zu können.

Weiterhin möchte ich mich bei Hermann Hild, Uwe Meier, Steve Omohundro, Nelson Morgan, Yochai Konig und Joachim Köhler für ihre Unterstützung bedanken. Ohne Hermann Hilds signifikanten Beitrag mit seinem akustischen Erkennungssytem wäre die schnelle Realisation des Lippenlese-Projekts undenkbar gewesen. Steve Omohundros exzellentes "Vision"- und "Learning"-Wissen und die Zusammenarbeit mit Nelson Morgans Speech-Gruppe hatten starken Einfluß auf den aktuellen Ansatz.

Für das wichtige und sehr zeitraubende Daten-Sammeln gebührt Uwe Meier und Peter Scheytt und allen Sprachspendern großer Dank.

Und nicht zuletzt möchte ich meinen Eltern danken, die mir dieses Studium ermöglicht haben.

# Zusammenfassung

In dieser Arbeit wird gezeigt, wie die Erkennungsleistung in der automatischen Sprachverarbeitung durch zusätzliches Lippenlesen (sogenannte bimodale Spracherkennung) deutlich verbessert werden kann. Dies wird exemplarisch an einem existierenden Erkennungssystem, einem sogenannten modularen MS-TDNN demonstriert. Dazu werden für das akustische und visuelle Sprachsignal sowohl verschiedene Vorverarbeitungsmethoden als auch verschiedene Sensor-Fusionen untersucht. Die Verfahren wurden auf das äußerst schwierige Buchstabierproblem angewandt. Mit der bimodalen Erkennung konnte die Wortfehlerrate bis zu 50 % gegenüber rein akustischer Erkennung reduziert werden.

# Kapitel 1

# Einleitung

## 1.1 Geschichte

Maschinelle Spracherkennung hat in den letzten Jahren einen enormen Aufschwung erlebt. Obwohl das Gebiet so alt ist wie die Entwicklung des Computers selbst, ist es immer noch sehr schwierig, die Leistung automatischer Spracherkenner auf ein Niveau zu bringen, das reale Anwendungen ermöglicht. Das ultimative Ziel, ein System zu entwickeln, das eine Leistung vergleichbar zur menschlicher Spracherkennung besitzt, ist aber noch lange nicht erreicht.

Erste Erkennungssysteme waren fähig, isolierte Wörter eines stark begrenzten Vokabulars sprecherabhängig zu klassifizieren. Wichtige Schritte in Richtung besserer Leistung waren erhöhter Wortschatz, Sprecherunabhängigkeit und Erkennung von kontinuierlicher und spontaner Sprache. Parallel zur Erhöhung der Komplexität des Problems wurden auch die Erkennungssysteme komplexer. In der akustischen Vorverabeitung des Sprachsignals etablierten sich Methoden wie Fourier/Melscale Filterbänke, Cepstral Kodierung, "Linear Predictive Coding" (LPC), bis hin zu aufwendigen biologisch motivierten Gehör-Modellen ("auditory models").

Auf dieser Vorverarbeitung basierdende primitive "template matching" Algorithmen wurden flexibler durch Techniken der dynamischen Programmierung ("Dynamic Time Warping"), und durch Lernverfahren ergänzt oder ersetzt. Populäre Ansätze sind Vektorquantisierung, "Lerning Vector Quantization" (LVQ), "Hidden Markov Models" (HMM), bayessche Methoden und

letztendlich konnektionistische Ansätze. Sogenannte hybride Ansätze, in denen mehrere Techniken vereint werden, haben auch an Popularität gewonnen.

Die verschiedenen Stufen der Erkennung wie Segmentierung, Phonem-Klassifizierung und Wort-Klassifizierung wurden in klassischen Ansätzen getrennt gelöst, in neueren Ansätzen jedoch als stark voneinander abhängige Probleme erkannt und als global optimierendes Systeme implementiert. Die Erkennung wurde auf höhere Abstraktionsstufen erweitert. Der linguistische Kontext wurde zur Erkennung in niedrigeren Stufen verwendet. Statistische Sprach-Modelle ("language models") sind ein Beispiel erfolgreicher Ansätze. Semantische Deutung der gesprochenen Sprache und Übersetzung in andere Sprachen sind momentane Forschungsfronten.

## 1.2 Neue Fronten

Die Aktivität und Konkurrenz in der automatischen Spracherkennungs ist sehr hoch. Erkennungsraten von "state-of-the-art" Systemen liegen eng zusammen. Es gibt einige Standardprobleme, wie z.B. "Erkennung von buchstabierten Sequenzen" oder Standard-Datenbanken wie die DARPA "Resource Management" Datenbank oder der "World Street Journal Task". Übliche sprecherunabhängige Wort-Erkennungsraten für englisch buchstabierte Sequenzen liegen im Moment bei etwa 90 %; sprecherabhängig wurden sogar schon 100 % erreicht. Beim sprecherunabhängigen "Resouce Management Task" sind 75 % übliche Wort-Erkennungsraten (Perplexität[1] 1000).

Man fragt sich nun, warum bei solch guten Erkennungsraten Spracherkennungs-Systeme immer noch nur in den Forschungslabors zu bewundern sind und in den wenigsten Fällen ihren Weg zu praktischen Anwendungen gefunden haben.

Abgesehen davon, daß die meisten Forschungssysteme sehr starken Prototyp-Charakter besitzen und oft kein Echtzeit-Verhalten zeigen, versagen Erkenner bei einem Feldeinsatz meist an nicht beachteten Phänomene wie Hintergrundrauschen, mehrere Stimmen im Raum oder veränderte Mikrophon-Charakteristik. Für einen praktischen Einsatz in Büros, Cockpits, oder Werkhallen muß ein System jedoch stabil gegen solche Phänomene

[1] Perplexität entspricht der Entropie. Ganz grob ist es eine Schätzung für die durchschittliche Anzahl von möglichen Folgewörter pro Wort in der verwendeten Datenbasis.

sein. Die existierenden Vergleichs-Datenbanken sind aber meist in einem rauscharmen Raum mit hoch-qualitativen Mikrophonen aufgenommen und die sogenannten "state-of-the-art"-Erkennungssysteme sind speziell auf diese Datenbanken optimiert.

Neuere Ansätze adressieren dieses Problem mit Rauschreduktions-Verfahren, kanalinvarianten Vorverabeitungen (z.B. RASTA-PLP) oder Trainingsdaten, die in verrauschter Umgebung aufgenommen wurden.

In vielen Fällen ist es jedoch sinnvoll, neben all diesem a-priori-Wissen mehr Information direkt von der Quelle zu bekommen. Vom Menschen abgeschaut gibt es Verfahren, die auf Stereo-Mikrophonen oder ganz allgemein auf $n$ Mikrophonen basieren. Unter Ausnutzung der Phasen-Information oder anderen trickreichen Verfahren ist es möglich, verschieden Störquellen auszufiltern.

Akustische Signale sind aber nicht die Quelle selbst, sondern das Produkt von Lungen, Stimmbändern und Vokaltraktstellungen. Der Vokaltrakt und die Stimmbänder sind die wirklichen Quellen der gesprochenen Sprache (Bild 1.1). Würde man die zeitliche Veränderung der Grundfrequenz und der exakten Vokaltraktstellung direkt schätzen, könnte man 100%'tig Phoneme klassifizieren. LPC versucht z.B. vom akustischen Signal mit Hilfe einer Modellannahme (Vokaltrakt ist linearer digitaler Filter) Parameter des Vokaltraktes zu schätzen. Diese Modelannahme ist jedoch nicht vollständig und das Verfahren liefert nur die richtigen Filter-Koeffizienten in störfreier Umgebung.

Eine andere Möglichkeit, einen Teil der Vokaltraktstellung zu bestimmen ist die visuelle Schätzung des sichtbaren Teiles. Dieses Verfahren ist weitläufig unter dem Namen Lippenlesen bekannt. Die visuel sichtbaren Merkmale bestimmen aber nicht eindeutig den kompletten Vokaltrakt. Ergänzt man jedoch akustische Spracherkennung mit Lippenlesen, erhält man eine bedeutend robustere Schätzung der Signalquellen-Parameter. Diese Arbeit hier beschäftigt sich mit der visuellen Schätzung und der Integration in die konventionelle akustische Spracherkennung. Im Englischen wird dies allgemein "Speechreading" genannt, hier aber als "bimodale Spracherkennung" bezeichnet.

Das Phänomen wird unter zwei unterschiedlichen Gesichtspunkten betrachtet. Zum einen ist es eine Alternative in der robusten Spracherkennung unter verrauschten Bedingungen und zum anderen gehört es als Teilsystem in ein multimodales Projekt, in dem verschiedene Eingabe-Modalitäten wie On-

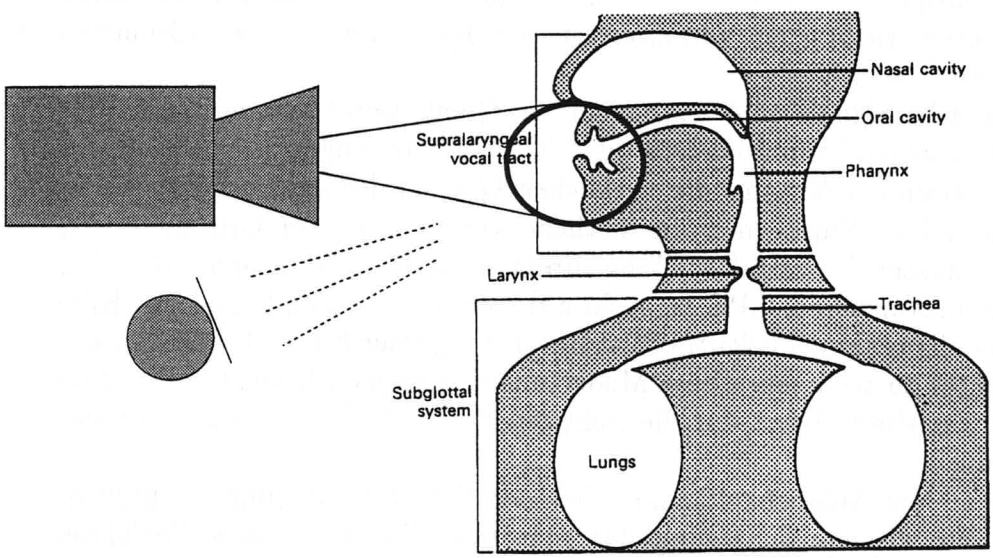


*Figure 2.1. The three physiologic components of human speech production.*

Abbildung 1.1: Vokal-Trakt

Line-Handschrifterkennung (OCR), Blickschätzung (eye tracking), Gestik-Erkennung, Spracherkennung und Lippenlesen zu einer Benutzerschnittstelle vereinigt werden (Bild 1.2).

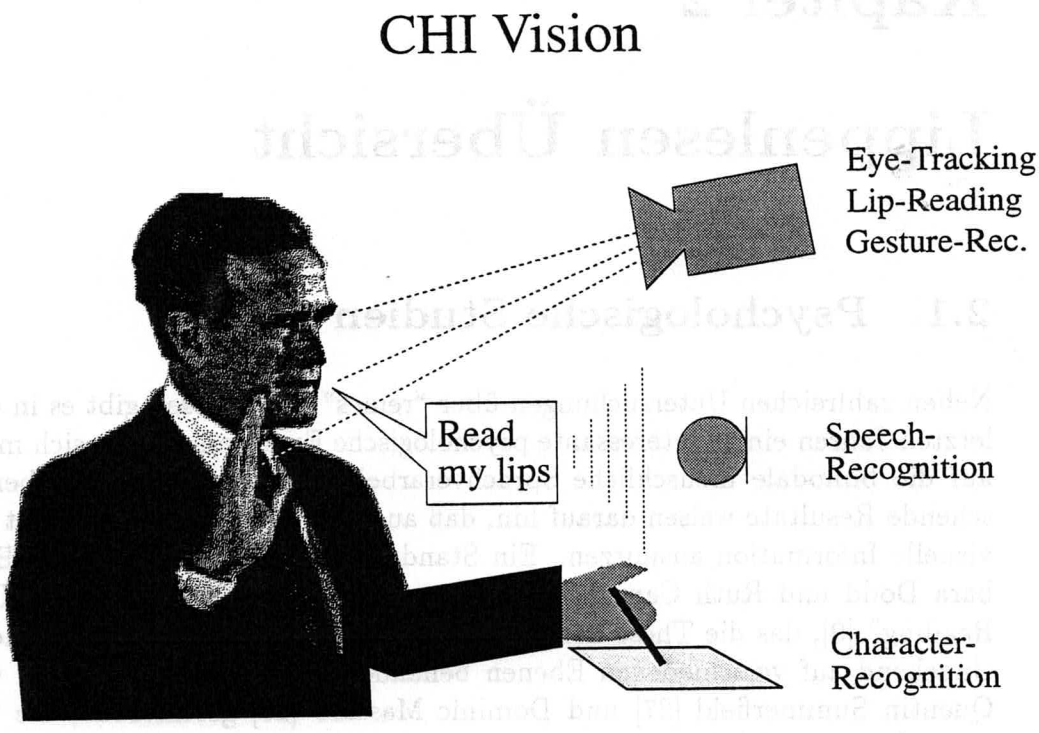


Abbildung 1.2: Vision der integrierten Erkennung verschiedener Eingabemodalitäten

# Kapitel 2

# Lippenlesen Übersicht

## 2.1 Psychologische Studien

Neben zahlreichen Untersuchungen über "reines" Lippenlesen gibt es in den letzten Jahren einige interessante psychologische Experimente, die sich mehr auf die bimodale menschliche Sprachverarbeitung konzentrieren. Überraschende Resultate weisen darauf hin, daß auch normal Hörende inherent die visuelle Information ausnutzen. Ein Standard-Werk ist das Buch von Barbara Dodd und Ruth Campbell "Hearing by Eye: The Psychology of Lip-Reading" [9], das die These der bimodalen Perzeption bei normal Hörenden eingehend auf verschiedenen Ebenen behandelt. Zwei Kapitel wurden von Quentin Summerfield [27] und Dominic Massaro [20] geschrieben, die unter anderem ausführlich den sogenannten McGurk-Effekt studiert haben. Es handelt sich dabei um eine bahnbrechende Arbeit von McGurk und MacDonald [22] über die "akustisch-visuelle Fusions-Illusion" (audio visual blend illusion). Versuchspersonen wurde das akustische Signal für das amerikanische "ba" und synchron dazu das Videoband für die Lippenbewegung von "ga" vorgespielt. In den meisten Fällen hatten die Versuchpersonen die Illusion ein "da" zu hören. Ich selbst war auch einmal ein Opfer dieses Effekts während eines Seminars von Massaro. Er erzeugte mit einer Workstation verschiedene künstliche Lippenbewegungen zum gleichen akustischen Signal. In einer ersten Phase sah die ganze Audienz die Computer-Animation mit dem Sprachsignal und hatte den Eindruck, es würden verschiedene Phoneme erzeugt. In einer zweiten Phase schloß eine Hälfte der Audienz die Augen und

hörte selbstverständlich immer das gleiche Phonem, die andere Hälfte mit offenen Augen hatte jedoch immer noch den Eindruck, verschieden Phoneme zu hören.

Nähere Untersuchungen ergaben, daß das erkannte Phonem in der Nähe des Mittelwertes des erzeugten akustischen Phonemes und des visuellen Phonemes (bezüglich eines Distanzmaßes) liegt. Die Illusion ist jedoch nicht nur auf Phoneme begrenzt. Zum Beispiel wird das akustische "bows" und das visuelle "goes" als "doze" oder "those" verstanden.

All diese Experimente deuten sehr wohl darauf hin, daß wir uns ein Modell des sichtbaren Vokaltraktes und dessen Korrelation zum akustisch hörbaren Signal bilden. Die widersprüchliche akustisch-visuelle Information beim McGurk Effekt kann in der Realität nicht vorkommen. Ein "bows" kann nie ohne initiales Zusammenpressen der Lippen erzeugt werden. Sehen wir jedoch Lippenbewegungen, in denen dieses Zusammenpressen nicht vorkommt ("goes"), ist die Wahrscheinlichkeit höher, daß wir uns vielleicht verhört haben. Das erkannte "doze" oder "those" ist keine Illusion, sonder einfach nur eine wahrscheinlichere Hypothese als "bows" selbst.

Neben dem Auflösen von Mehrdeutigkeiten gibt es jedoch noch ein anderes bekanntes Szenario, der sogenannte "Cocktail-Party-Effekt": man unterhält sich mit einer Person, zur gleichen Zeit unterhalten sich jedoch auch andere Personen mit der gleichen Lautstärke. In diesem Falle werden zusätzliche Informationen, wie z.B. Signalunterschiede zwischen linken und rechtem Ohr, sowie die visuelle Information stark ausgenutzt. Der sogenannte Blickkontakt ist nicht nur eine freundliche Geste, sondern macht es auch oft einfacher, sein Gegenüber zu verstehen.

## 2.2 Existierende maschinelle Ansätze

Motiviert durch diese psychologischen Experimente und ermöglicht durch das Aufkommen von besserer und schnellerer Video-Hardware und fortschrittlicheren Bildauswertungsalgorithmen, sind in den letzten Jahren auch einige interessante maschinelle Lippenleseansätze aufgekommen.

Als erste bekannte Arbeit im machinellem Lippenlesen gilt ein Patent von Ernie Nassimbene (IBM, U.S.Patent 3192321, 29. Juni 1965). Mit Hilfe einer Photozellenmatrix wurden Reflexionen von den Zähnen gemessen und als Unterstützung zur akustichen Spracherkennung benutzt. Das Projekt

wurde aber nicht lange fortgesetzt.

19 Jahre später (1984) veröffentlichte Eric Petajan (AT&T) eine erwähnenswerte Arbeit (Dissertation): "Automatic Lipreading to Enhance Speech Recognition" [10]. Die Bildauswertung basierte auf einem Hardware-Schwellwertoperator um Konturen zu erkennen. Voraussetzung war eine sehr gute Beleuchtung von mehreren Seiten. Basierend auf dieser Konturkodierung und einigen Heuristiken wurden die Nasenlöcher des Sprechers im Bild verfolgt und daraus die Position der Lippen berechnet. Für jedes Bild der abgespeicherten Sequenz wurden die Höhe und Breite der Lippenkontur, die Fläche und der Umfang berechnet. Mittels "Dynamic Time Warping" wurden dann Distanzen zu vier Beispielsequenzen pro Wortklasse berechnet. Das Distanzmaß basierte nicht auf den absoluten Werten, sondern auf dessen Änderung. Die Wortgrenzen selbst wurden mit einem akustischen Erkenner bestimmt (Voterm). Bimodale Erkennung wurde in zwei Schritten betrieben. Zuerst wurden die $n$ besten akustischen Hypothesen berechnet, dann basierend auf der visuellen Klassifikation die wahrscheinlichste Hypothese der n Besten ausgewählt. In Tests mit Zahlen und Buchstaben-Erkennung konnte Petajan zeigen, daß durch Lippenlesen die Erkennungsrate etwas verbessert wird. (Isolierte sprecherabhängige Zahlen: von 95% auf 100%, isolierte sprecherabhängige Buchstaben: von 64% auf 66%).

Von den zahlreichen folgenden Ansätzen möchte ich nur noch drei weitere Arbeiten nennen: Kenji Mase und Alex Pentland stellten ein System, basierend auf optischer Flußanalyse, vor [19]. Die durchschnittlichen Verschiebungsvektoren in vier Gebieten um die Lippen herum wurden berechnet und mittels prinzipieller Komponenten Analyse klassifiziert. Sprecherabhängige Test mit dem Zahlen-Vokabular ergaben zwischen 73% und 100% visuelle Erkennungsleistung.

Ben Yuhas hat ein neuronales Netz trainiert, um das akustische Frequenzspektrum von Vokalen mittels dem Grauwertbild der Lippenstellung zu schätzen [2]. Er beschränkte sich auf statische Bilder. Für die bimodale Erkennung wurde das gewichtete Mittel zwischen visueller Frequenzspektrum-Schätzung und dem wirklichen akustischen Frequenzspektrum gebildet. Die relative Gewichtung wurde heuristisch aus dem Signal/Rausch-Verhältnis berechnet. Mittels eines Vokalerkenners wurde eine Erkennungsrate z.B. von 80% bei einem Signal/Rausch-Verhältnis von 10db und einem Vokabular von 9 Vokalen erreicht.

Als einer der aktuellsten Ansätze gilt ein konnektionistisches System von

David Stork, Greg Wolff und Earl Levin [8]. Der verwendete Phonem-Klassifikator ist ein modulares Time-Delay-Neural-Network (TDNN [28]). Die Datenbasis bestand aus isolierten Buchstaben, die von fünf verschiedenen Sprechern aufgenommen wurden. Das Vokabular beschränkte sich auf die zehn englischen Buchstaben b,d,f,m,n,p,s,t,v, und z. Ein TDNN wurde auf den akustischen "Melscale"-Koeffizienten trainiert und ein zweites TDNN auf visuelle Merkmale. Den Sprechern wurden an zehn Punkten um die Lippen, die Nase und des Kinn reflektierende Punkte geklebt und diese mit einer speziellen Kamera aufgenommen. Die Koordinaten der Punkte wurden von einer externen Firma extrahiert, die auf Sport-Bewegungsstudien spezialisiert ist. Der Ansatz hatte also keine Bildauswertung. Neben diesem Schwachpunkt wurde auch nicht auf die Synchronisation zwischen Bild und Akustik geachtet, was eine erfolgreiche bimodale Sensorfusion nur auf der Phonem-Ebene ermöglicht. Die Ausgabe-Einheiten der TDNN's hatten die "Softmax"-Aktivierungsfunktion [7], die es ermöglicht die Ausgabeaktivität als Wahrscheinlichkeit zu interpretieren. Die akustische TDNN-Ausgabe wurde dann einfach mit der visuellen TDNN-Ausgabe multipliziert und als bimodale a-posteriori Wahrscheinlichkeit $p(C|A,V)$ interpretiert. Die akustische Erkennungs-Rate war mit 64% überraschend niedrig, die visuelle Erkennung betrug 51% sowie die bimodale Erkennungsrate 91%. Die exakte Größe der Datenbank und die Aufteilung in Trainings- und Test-Daten wurde in den Veröffentlichungen nicht genau beschrieben.

# Kapitel 3

# Versuchsaufbau

Ein Ziel des hier beschriebenen Ansatzes ist es, in der Zukunft ein komplettes echtzeitfähiges System zu bilden, das basierend auf Kamera und Mikrophon bimodale Erkennung in kontinuierlich gesprochener Sprache betreiben kann. Für reine akustische Systeme und für rein Bildauswertungssysteme gibt es zahlreiche etablierte Hardware- und Software-Umgebungen. Der "Knackpunkt" liegt jedoch in der Kombination beider Modalitäten und der Anforderung, daß die Hardware echtzeitfähig ist.

## 3.1 Hardware

In bimodaler Spracherkennung hat man es mit zwei sehr unterschiedlichen Eingabequellen zu tun. Das schlägt sich auch in der angebotenen Schnittstellentechnik nieder. Das akustische Signal ist relativ niederfrequent (8-16 KHz Abtastrate), es ist aber eine feine Quantisierung gefragt (12-16 bit). Das Bildsignal ist sehr hochfrequent (bei 30 Vollbilder/sec mit 256x256 pixel: 1,97 MHz) jedoch eine Grauwertquantisierung von 6-8 bit ist schon ausreichend.

Als akustische Schnittstelle wird ein Analog/Digital-Wandler von der Firma Gradient namens Desklab benutzt. Er hat einen Mikrophoneingang, Lautsprecher, internen digitalen Puffer und eine SCSI-Schnittstelle. Dieses Produkt ist international in der maschinellen Sprachverarbeitung etabliert. Die angebotene Software für DEC-Ultrix-Workstations und Sun-Workstations ist sehr einfach zu programmieren. Das Sprachsignal wird blockweise mit bis zu 16KHz und bis zu 16 bit Auflösung in den internen

Puffer gelesen und dann blockweise über die SCSI-Schnittstelle in die Workstation kopiert.

Die Schnittstelle für eine Videokamera ist aufwendiger und stark von der verwendeten Workstation abhängig. Diese Hardware wird im Allgemeinen "framegrabber" genannt und ist oft der Flaschenhals des kompletten Systems. Zur Auswahl standen Lösungen für NeXT, DEC und SGI. Dank eines sehr schnellen internen Bus-Systems haben wir uns für die DEC Lösung entschieden. Die Karte heißt DECVideo und ist eine DEC eigene Multimedia-Karte mit seperaten Video-Puffer und einer Tochterkarte als framegrabber. Der interne Video-Puffer kann von einer Videokamera beschrieben werden und wird für die Bildschirmanzeige ausgelesen. Es können deshalb in Echtzeit verschiedene Videoquellen auf dem Bildschirm angezeigt werden. Problematisch ist das schnelle Kopieren vom internen Video-Puffer in das CPU-RAM (grabben). Auf einer DEC 5000/240 in einer X-Windows-Umgebung konnten wir Grauwertbilder von einer Kamera mit einer Auflösung von 256x256 Pixel (8bit) in Echtzeit (30 frames/seconds) in das CPU-RAM kopieren. Die Karte unterstützt die sogenannte Xv-Erweiterung von X-Windows, die die nötigen Funktionen für Digitalisieren (grabbing) und Anzeigen anbietet.

Als weitere Hardware haben wir ein Stand-Mikrophon und eine NTSC[1]-Kamera verwendet.

## 3.2 Bimodale Daten Aquisitation

Der kritische Punkt bei der bimodalen Spracheingabe ist, die zeitliche Korrelation zwischen Ton und Bild nicht zu verlieren. Unser Aufnahmeprogramm speichert die genaue Systemzeit für jedes Bild und die Systemzeit zu der der akustische A/D-Wandler anfängt abzutasten.

Als Verifikation der zeitlichen Korrelation wurde ein Programm entwickelt, das mit derselben relativen Systemzeit die gespeicherten Bildfolgen und das abgetastete akustische Signal abspielt. Mit diesem Programm konnten wir bestätigen, daß die Methode der Systemzeit-Abspeicherung genügend zeitliche Auflösung beinhaltet. Es konnten keine zeitlichen Verzerrungen oder Verschiebungen festgestellt werden (speziell nicht im bimodalen Zeitlupen

[1]NTSC ist die amerikanische und japanische Fernsehnorm. Sie unterscheidet sich von der europäischen PAL/SECAM-Norm hauptsächlich durch die höhere Bildrate von 30 Bilder/Sek. und unterschiedliche Farbkodierung.

Modus). Im Einzelbild-Modus dieses Programmes (Bild 3.1) haben wir auch das FFT-Spektrum des akustischen Sprachsignals angezeigt und mittels eines Zeitzeigers auf die FFT-Koeffizienten nochmals sicher gehen können, daß die Lippenbewegungen synchron zu den ausgesprochenen Phonemen abgespeichert sind.

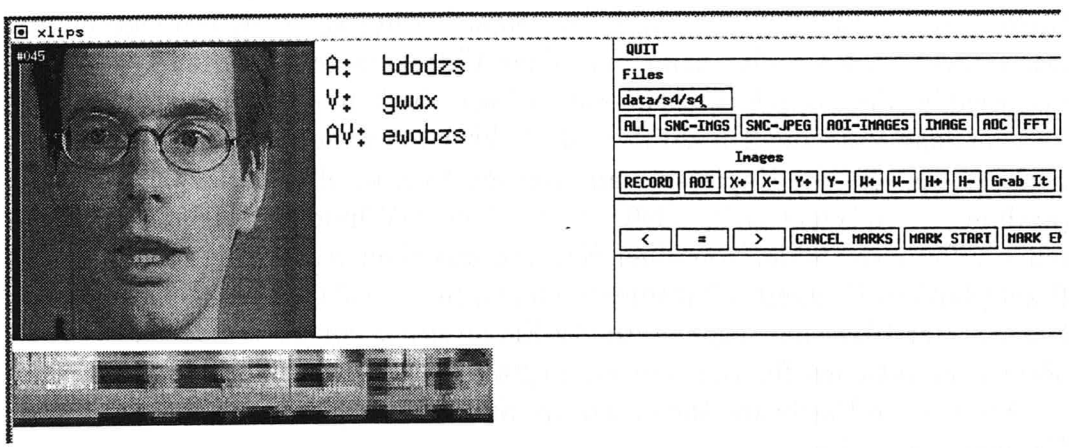


Abbildung 3.1: Xlips ist ein Tool, um die bimodale Datenbank abzuspielen, akustisch und visuell von Hand zu segmentieren und Vorverarbeitungs Algorithmen und Erkennungsergebnisse zu visualisieren.

Um die maximale Bildrate zu erreichen, haben wir zuerst 5-10 Sekunden eines gesprochenen Satzes im CPU-RAM abgespeichert (bei 256x256 Bilder = 10 - 20 MByte) und dann off-line als PGM [2] oder JPEG [3] Bilder abgespeichert.

Eine übliche Aufnahmesitzung spielt sich nun folgendermaßen ab: Eine Person setzt sich for die DEC-Workstation, mit Mikrophon und Kamera auf die Lippen gerichtet. Auf dem Bildschirm erscheint in großen Buchstaben der

[2] PGM steht für "Portabel Gray Map" und ist ein weitverbreiteter Standard für Grauwertbilder.

[3] JPEG ist ein Bild-Kompressions-Verfahren.

gewünschte Satz. Die Person drückt eine Taste, liest den Satz ab und drückt wieder eine Taste. Danach wird die Bildsequenz, das akustische Sprachsignal und die zeitliche Korrelation auf Platte abgespeichert.

# Kapitel 4

# Spracherkennungs-System

Unser Ziel ist es, bimodale kontinuierliche Spracherkennung zu betreiben. Alle bisher veröffentlichten Ansätze arbeiten entweder auf statischen Bildern und Vokal/Phonem-Klassifizierung, oder beschränken sich auf isolierte Buchstaben oder Ziffern. Da sehr wenig Erfahrung in der bimodalen Verarbeitung besteht, wurde in den Ansätzen nur an einfachen Problemen gezeigt, daß die Information über die Lippenbewegung erfolgreich zur akustischen Erkennung ausgenutzt werden kann.

Wir wollen die komplette Erfahrung eines "state-of-the-art" akustischen Ansatzes nutzen und auf diesem Niveau zeigen, daß bimodale Erkennung Sinn macht. Der hier vorgestellte Klassifikator ist ein sogenanntes Multi-State Time-Delay-Neural-Network (MS-TDNN). Er vereinigt konnektionistische Phonem-Klassifizierung und dynamisches Programmieren zur kontinuierlichen Wort-Klassifizierung. Im Folgenden werden die drei Stufen des Systems beschrieben: Zuerst die bimodale Vorverarbeitung, dann die Phonem-Klassifizierung und abschließend die Wort-Klassifizierung.

## 4.1 Akustische Vorverarbeitung

Das digitalisierte Sprachsignal ist immer noch zu hochdimensional, um effiziente Klassifizierung zu betreiben. Es muß in einer niederdimensionalen Form repräsentiert werden, ohne charakteristische Merkmale zu verlieren. Wie schon erwähnt, ist die wirkliche Sprachquelle der Vokaltrakt und die Stimmbandanregung. Die Stellung des Vokaltraktes und die Anregung durch

die Lungen gehorchen einer gewissen physikalischen Trägheit, die weit unter der Abtastrate von 16 KHz liegt. Die dadurch beeinflußte Frequenzverteilung des akustischen Signals ist also eine gewisse Zeit annähernd konstant. Heuristisch wurde eine Auflösung von 100 Hz als ausreichend ermittelt, auf der auch alle etablierten Ansätze basieren. Mit dieser "Abtastrate" kann nun direkt die Vokaltraktstellung beschrieben werden. Geht man davon aus, daß der Rachen, Mundhöle, Zähne, Zunge und Lippen wie ein linearer Filter auf den Grundton wirken (Faltung des Signals mit einer Maske), kann im 10 msec Takt diese Filterkoeffizienten (endliche Maske) bestimmt werden. Dieses Verfahren wird "Linear Predictive Coding" (LPC) genannt. Üblich ist, in einem 20 bis 40 msec "Hamming"-Fenster[1] 8 - 16 LPC Koeffizienten zu approximieren.

Ein anderes bekanntes Verfahren (das aber nicht mehr auf dieser Filter-Modellannahme basiert) ist die sogenannte Melscale Filterbank-Repräsentation. Sie basiert auf einem biologisch Model des Gehöhrs. Die für verschiedene Frequenzbereich unterschiedlich sensitiven Nervenzellen (choclear hair cells and auditory nerve-fibers) auf dem Trommelfeld werden als Bandpaßfilter beschrieben. In Wirklichkeit besitzt das Ohr über 100 solche Zellen, heuristisch wurde jedoch ermittelt, daß 16 Bandpaßfilter logarithmisch über das Frequenzspektrum verteilt ausreichen. Die Koeffizienten berechnet man mittels der schnellen Fouriertransformation (FFT) auf einem Hamming-Fenster von üblicherweiße 256 Signalwerten. Die Phaseninformation der Fouriertransformation wird dann eliminiert, da sie zur Spracherkennung nicht beiträgt. Die resultierenden Koeffizienten werden auf einer logarithmischen Skala zu den 16 Melscale-Koeffizienten zusammengefaßt[2].

Als letzte populäre akustische Vorverarbeitung sollte die sogenannte RASTA-PLP Kodierung erwähnt werden [13]. Sie macht besonderen Sinn in der robusten Spracherkennung. Oben wurde argumentiert, daß der Vokaltrakt einer gewissen Trägheit unterliegt und deshalb mit einer gewissen zeitlich lokalen Konstanz des Frequenzspektrums bei der Merkmalskodierung gerechnet werden kann. Es gibt jedoch noch andere Faktoren, die das Frequenzspektrum mit einer zeitlich deutlich globaleren Konstanz beeinflussen:

[1] Ein "Hamming"-Fenster ist ein Zeitfenster auf dem Sprachsignal, das nach außen abfällt, und mit den benachbarten Hamming-Fenstern überlappt.

[2] Die 16 Melsale-Koeffizienten sind die gewichteten Mittelwerte in 16 Bereichen auf dem Frequenzspektrum. Idealisiert haben die Gewichtungen jeweils Gaußverteilung um die logarithmisch verteilten Frequenzpunkte.

Änderung der Kanalcharakteristik und Rauschen. Die Charakteristik des Kanales wird von Mikrophon, Richtung des Sprechers zum Mikrophon, Raumakustik und Art des Sprechers bestimmt.

Das heißt, die Änderung des Frequenzspektrums in zeitlich zu kurzen und zu langen Abständen ist für die Erkennung der gesprochenen Sprache irrelevant. Die Idee ist nun, eine zusätzliche Bandpassfilterung über Merkmalsvektor-Folgen anzuwenden. Dies ist nicht zu verwechseln mit den Bandpassfilterungen auf dem digitalisierten Sprachsignal selbst. Bei RASTA-PLP wird eine Bandpassfilterung auf den Logarithmus des Frequenzspektrums angewandt, was die Kodierung invariant zu konvolutionärem Rauschen (Kanalcharakteristik) macht. Es sind Studien im Gange, RASTA-PLP auf additives Rauschen zu erweitern [18].

In dieser Arbeit wurden zwei unterschiedliche akustische Vorverarbeitungen untersucht: Die Melscale-Filterbank-Kodierung und die RASTA-PLP-Kodierung. Bei der Melscale-Kodierung basiert die Klassifizierung auf einer Folge von 16 dimensionalen Vektoren mit 10 msec Abstand und bei der RASTA-PLP-Kodierung auf einer Folge von 9 dimensionalen Vektoren mit 10 msec Abstand.

## 4.2 Visuelle Vorverarbeitung

Die visuelle Vorverarbeitung ist weitaus problematischer als die akustische Vorverarbeitung. Sie erfordert bei etwa 2 MByte/sec eine unvergleichbar größere Dimensionsreduktion. Das Gebiet der Bildauswertung ist noch nicht in dem selben "stabilen" Zustand wie die akustische Signalverarbeitung. Der richtig gewählte Ansatz hat hier äußerst starken Einfluß auf den Erfolg des kompletten Systems. Er ist der Flaschenhals hinsichtlich der Echtzeitfähigkeit und der visuellen Erkennungsleistung.

Wie in fast allen Bildauswertungs-Szenarios sind zwei Teilprobleme zu lösen: Segmentierung und Erkennung. In unserem Fall bedeuted Segmentierung, zunächst Gesicht und dann die Lippen im Bildbereich zu finden. Erkennung bedeutet, die spezielle Lippenstellung oder Bewegung in einer geeigneten Form zu repräsentieren und zu klassifizieren. Ideal wäre eine Lösung, in der beide Probleme eng gekoppelt gemeinsam gelöst werden (wie es z.B. schon in der akustischen Erkennung bezüglich der zeitlichen Segmentierung und der Phonem-Klassifizierung mit TDNN's gelöst ist).

Eine andere Fragestellung ist, in wieweit eine Informationsreduktion in der Vorverarbeitung geschehen soll und wieviel dem konnektionistischen Klassifikator überlassen werden soll. Der "trade-off" ist, bei zuviel Reduktion gehen signifikante Merkmale verloren, bei zu wenig Reduktion leidet die Generalisierungsfähigkeit des Lernverfahrens für den Klassifikator. Solange sehr allgemeines Modellwissen in der Vorverarbeitung verwendet wird, hilft es dem Klassifikator invariant zu verschiedenen Phänomenen zu bleiben. Bei zuviel Modellwissen muß das Modell selbst von den Daten gelernt werden und es ist unbedingt nötig, auch "Top-Down"-Informationsfluß in der Erkennung zu haben. Diese Diskussion ist aber auch sehr stark von "religiösen Glaubenskämpfen" geprägt, die meist von zwei extremen Lagern aus geführt werden: Die "Konnektionisten", die blind auf ihr Lernverfahren vertrauen, und dafür alle Daten der Welt sammeln, und die Verfechter der klassischen heuristischen Methoden, die alles über das spezielle Problem schon im Voraus wissen und nur noch in ihren Algorithmus einbauen müssen. Die Kunst ist, ein ausgewogenes Verhältnis zwischen beiden Extrema zu finden. Populär ist z.B. das Lernverfahren stark dem Problem angepaßt einzuschränken (In TDNNs ist dies z.B. durch sogenanntes "Weightsharing" erreicht. s.u.).

Für den ersten Prototyp wurde ein mehrstufiges System implementiert, das teilweise die klassische Idee verfolgt und teilweise stark auf dem konnektionistischen Klassifikator basiert. Der Ansatz setzt voraus, daß die Lippenposition schon gefunden ist. Ansätze zur Lippenfindung sind im Gange.

### 4.2.1 Face Tracking

Die im Moment verwendete Lippenfindung wird anhand eines Systems durchgeführt, das von Peter Rander in Takeo Kanades Bildauswertungslabor an der CMU entwickelt wurde. Es ist ein allgemeines Gesichtsverfolungs-System, daß kollaborativ auch in anderen multimodalen Projekten verwendet wird. Die Portierung und Erweiterung auf DEC Workstations wurde von Uwe Meier übernommen. Das System ist völlig abgekoppelt von der hier beschriebenen Arbeit, sollte aber trotzdem erwähnt werden, da es als Teil für ein geplantes bimodales Demo-System vorgesehen ist.

Es repräsentiert die erste Stufe der Vorverarbeitung und verfolgt einen klassischen hierarchischen "template-matching" Ansatz. Zuerst wird mittels einer "Gaussian Pyramid" in verschiedenen Auflösungen das Bild tiefpassgefiltert und hierarchisch mit verschieden genauen "Templates" verglichen.

Genauere Details können in [25] nachgelesen werden.

Bild 4.1 zeigt ein typisches Resultat des Face-Tracking-Systems. Uns interessiert in dieser Arbeit nur die Koordinaten der Lippen.

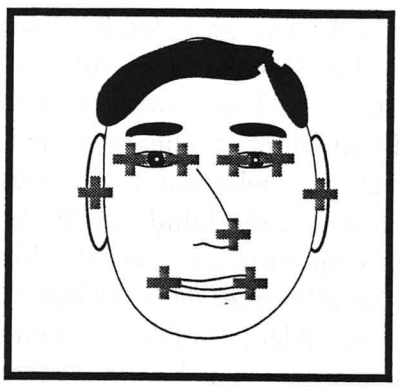

Abbildung 4.1: Veranschaulichung der mit Peter Randers Face-Tracker (RTFT) gefundenen Augen, Ohren, Nase und Mund

### 4.2.2 Grauwert Kodierung

Basierend auf den gefundenen Koordinaten wird ein Rechteck um die Lippen gelegt (Bild 4.2). Dieses Rechteck wird im Folgenden AOI (Area Of Interest) genannt. Der naheliegendste Ansatz ist die Grauwertmatrix des AOI direkt als Eingabe für den Klassifikator zu verwenden. Damit ist man auf der sichersten Seite bezüglich Informationsverlust (man verliert keine signifikanten Merkmale durch die Vorverarbeitung). Im Durchschnitt hat ein AOI die Größe von 80 Pixel Breite und 50 Pixel Höhe, was einem 4000 dimensionalen Merkmalsvektor entspräche. In Experimenten mit niedrigeren Auflösungen bis zu 24x16 Pixel AOI (384 dimensional) haben wir festgestellt, daß mit dem Auge die Lippenstellung immer noch eindeutig klassifiziert werden kann. Das sollte theoretisch den Klassifikator nicht hindern, auch auf dieser niedrigeren

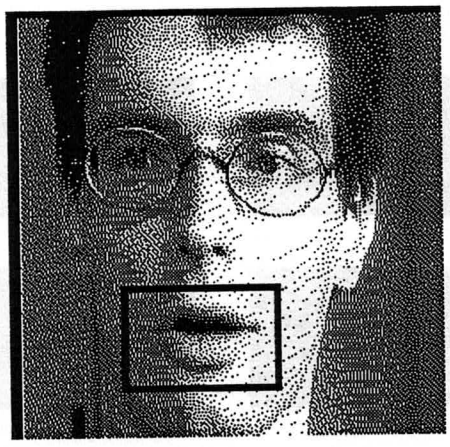

Abbildung 4.2: Von Hand segmentiertes oder durch RTFT gefundenes AOI

Auflösung gleiche Erkennungsleistung zu vollbringen. Niederdimensionalere Merkmalsvektoren mit annähernd gleichem Informationsgehalt sollten für das Lernverfahren des Klassifikators sogar von Vorteil sein.

Unabhängig von der wirklichen Größe des AOI skalieren wir nun die Matrix auf 24x16 Pixel. Jedes Element ist der Grauwert-Durchschnitt des entsprechend größeren Bereiches im original Bild. Dies entspricht einer Tiefpaß-Filterung (die nebenbei noch Rauschen vermindert). Die Grauwerte zwischen 0 und 255 werden linear zwischen -1 und 1 skaliert und als 384 dimensionaler Eingabevektor an den neuronalen Netzwerk-Klassifikator weitergereicht. Bild 4.3 zeigt eine typische Folge von AOIs.

Optional ist auch noch eine Grauwertnormalisierung implementiert. Basierend auf dem Histogramm aller im Bild vorkommenden Grauwerte, wird den obersten 5 % im Histogramm der Wert 1 zugeordnet, den untersten 5 % der Wert -1 und die restlichen 90 % im Grauwerthistogramm werden zwischen -1 und 1 skaliert. Diese Kodierung ist im Groben invariant zu verschiedenen Beleuchtungsstärken. Sie ist motiviert durch die Eingabekodierung im ALVINN-Projekt von Dean Pommerleau [24].

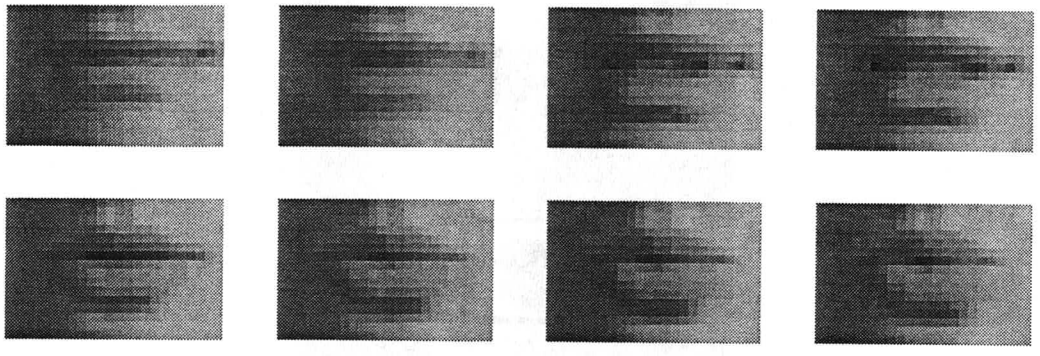

Abbildung 4.3: Tiefpass-gefilterte AOI Sequenz

### 4.2.3 Frequenz-Kodierung

Die Grauwert-Kodierung hängt sehr stark von einem gut positioniertem AOI ab. Der in Abschnitt 4.2.1 vorgestellte Lippenverfolger ist jedoch nicht sehr genau in der Positionierung. Dies war die Motivation für eine verschiebungsinvariante Vorverarbeitung, die sogenannte zwei-dimensionale Fouriertransformation. Ein weiterer Vorteil ist die beleuchtungsinvariante Repräsentation, wenn der "0te" Fourier-Koeffizient (Energie) weggelassen wird, und die Tiefpaßfilterung, wenn höhere Koeffizienten abgeschnitten werden. Grundsätzlich gibt es zwei Überlegungen: 1. Mit wieviel Koeffizienten lassen sich die Lippenstellungen ausreichend kodieren. 2. Reicht der Betrag der Fouriertransformation aus, oder sollte die komplexe Darstellung beibehalten werden (ist die Phaseninformation wichtig).

Wieder wurde empirisch mit dem Auge ermittelt, daß die ersten 15x15 2D-FFT Koeffizienten ausreichen, um die Lippen zu kodieren. Ein Lippen-AOI wurde fouriertransformiert, im Frequenzbereich wurden außerhalb eines Fensters alle Koeffizienten auf 0 gesetzt, und dann die inverse Fouriertransformation angewandt. Das resultierende Bild konnte mit einem 15x15 Fenster im Frequenzbereich mit dem Auge noch gut klassifiziert werden. Dieses Verfahren liefert nun zwei alternative Merkmalskodierungen: 1. Ein 225 dimensionaler Vektor bestehend aus den Beträgen der 2D-FFT. 2. Ein 450 dimensionaler Vektor bestehend aus den komplexen Zahlen der 2D-FFT. Für

die 2D-FFT wurden die AOIs auf 128x128 Pixel normalisiert.

### 4.2.4 Modelgestützte Parameter Extraktion & andere Arten der Dimensionsreduktion

Wie schon oben diskutiert, wäre es besser, Segmentierung und Erkennung zu vereinigen. Die naheliegendste Lösung wäre, dem neuronalen Netzwerk-Erkenner anstatt dem vorsegmentierten AOI das komplette Bild zu präsentieren. Der Erkenner könnte z.B. über das ganze Bild von links nach rechts und oben nach unten verschoben werden, oder im Erkenner selbst könnte ein geschicktes "weight-sharing" (s.u.) implementiert werden. In [17] und [21] wurden solche Ansätze schon erfolgreich auf Handschrifterkennung angwandt. All diese Ansätze sind jedoch immer noch sehr sensitiv zu Beleuchtung, Skalierung und Rotation. Auch könnte man dies bezüglich invariante Trainings- und Erkennungs-Methoden anwenden, das würde aber die Größe der Datenbank explosiv erhöhen.

Im Moment sind weitere Untersuchungen im Ansatz, das Modellwissen schon in der Vorverarbeitung zu benutzen. Aufbauend auf Energieminimierungs- Techniken wie "Snakes" [16] oder "deformable templates" [29] wird versucht, die Konturen oder die komplette Grauwertverteilung der Lippen zu lernen [6] und Dimensionsreduktion mittels Prinzipieller Komponenten Analyse schon in der Vorverarbeitung anzuwenden [5]. Für die Lippenverfolgung und Erkennung wird das gleiche Modellwissen verwendet. Alle Untersuchungen haben jedoch noch sehr starken Pilot-Experiment-Charakter und konkrete Aussagen über Erfolg und Mißerfolg zahlreicher Ansätze kann noch nicht gemacht werden.

Nichts desto trotz soll hier zumindest der klassische "Snake"-Ansatz und "deformable-template"-Ansatz beschrieben werden und in wieweit die aktuellen Algorithmen mit diesen Ansätzen verwandt sind.

#### Snakes

Klassische Bildverarbeitung besteht aus meheren Stufen: Zuerst Tiefpassfilterung des Bildes, dann Gradientenschätzung, dann Kantendetektion, dann Kantenverkettung. Diesem strikten "Einbahnstraßen"-Ansatz steht der Snake-Ansatz mit einer globalen dynamischen Optimierung über alle Stufen entgegen.

Es wird eine Energiefunktion definiert und mittels Gradientenabstieg wird nach einem lokalen Minimum gesucht. Das Resultat ist eine Konturkurve (geschlossen oder offen). Entlang dieser Kurve sind bestimmte Bildmerkmale maximal. Die Energie dieser Kurve (Snake) wird folgendermaßen definiert:

$$E_{snake} = \int_0^1 E_{int}(v(s)) + E_{image}(v(s))ds \tag{4.1}$$

$v(s) = (x(s), y(s))$ repräsentiert die Pixel-Positionen entlang der Snake ($s = 0..1$). Die interne Energie $E_{int}$ wird durch die folgenden ersten und zweiten Ableitungen definiert:

$$E_{int} = \frac{\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2}{2} \tag{4.2}$$

$\alpha(s)$ und $\beta(s)$ gewichten die zwei Terme unterschiedlich. Meistens sind beide Werte konstant ($\alpha(s) = a, \beta(s) = b$). Ist $v_s(s)$ klein, macht die Kurve keine großen Sprünge, ist $v_{ss}(s)$ klein, gibt es keine "Knicke" (die Kurve ist differenzierbar).

$E_{image}(v(s))$ ist minimal, wenn bestimmte Merkmale im Bild an der Stelle $v(s)$ maximal sind. Um Kanten zu finden, würde für $E_{image}(v(s))$ ein Gradientenoperator gewählt.

Um die Umrandungen der Lippen zu finden (boundary detection), wird mit einer initialen Schätzung der Lippenpostion angefangen (Bild 4.4). Basierend auf einer sehr großen Gradienten-Maske und großer Schrittweite wird $E_{snake}$ z.B. mit Gradientenabstieg minimiert. Die Maske und Schrittweite wird graduell verkleinert.

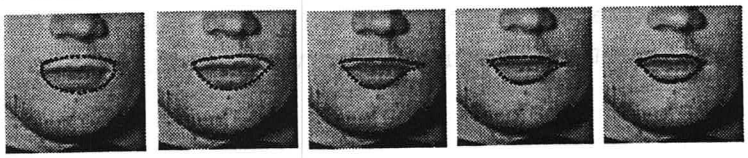

Abbildung 4.4: Eine typische Snake-Minimisierung

Dies geschieht über eine komplette Bildfolge. Es kann zusätzlich noch ein bestimmter zeitlicher Trägheitskoeffizient verwendet werden.

### Deformable Templates

Flexible Masken (deformable templates [29]) sind sehr verwandt zu Snakes. Sie basieren genauso auf einer Energieminimisierung von $E_{deftmpl}$, die in eine interne Energie $E_{int}$ und eine Merkmalsenergie $E_{image}$ aufgespalten ist. $E_{int}$ basiert jedoch auf einem niederdimensionalerem Modell. Für Lippenkonturen kann es z.B. aus Polynomen bestehen. Die zu minimierende Werte sind Abstände zwischen den Polynomendungen, Abstände der Polynomparameter zu gewissen Mittelwerten, usw. Bild 4.5 zeigt ein Beispiel-Modell. $E_{image}$ können wieder beliebige Bildmerkmale sein. Entlang der Lippenpolynome könnte es ein Gradientenoperator sein, und innerhalb der oberen und unteren Lippengrenze könnte es das Integral eines gewissen Hochpaßfilters sein (niedrige Energie entspricht konstanter Grauwertverlauf). Genauere Details können in [29] nachgelesen werden. Die Energie wird wieder durch Gradientenabstieg minimisiert.

### Gelernte Konturmodelle

Die beiden geschilderten Ansätze, Snakes und deformable templates sind für Objektverfolgung sehr populär. Snakes haben eine sehr allgemeine Annahme über die Konturform und deformable templates haben eine sehr spezifische Annahme über die Form.

Eine zu allgemeine Annahme macht das Lippenverfolgen instabil. Die Snake kann sehr leicht in ein anderes lokales Minimum "abgleiten". Benachbarte Konturen mit unterschiedlichen Formen sind übliche Fehlerquellen. Eine zu grobe Festlegung der Lippenform durch deformable templates könnte wichtige Formeigenschaften ignorieren. Die templates müssen von Hand parametrisiert werden.

Wir untersuchen im Moment verschieden Ansätze, in denen der interne Energieterm $E_{int}$ durch eine gelernte Funktion ersetzt wird. Im einfachsten Falle wird ein "Kontur-Raum" gelernt, der durch alle "legalen" Konturen definiert ist. Anhand von normalisierten Beispielkonturen kann z.B. eine prinzipielle Komponenten Analyse durchgeführt werden. Der Kontur-Raum ist durch die ersten $n$ prinzipiellen Achsen aufgespannt. Weitere Untersuchungen sind im Gange, den Raum durch ein Manigfalt zu repräsentieren (surface learning [6]).

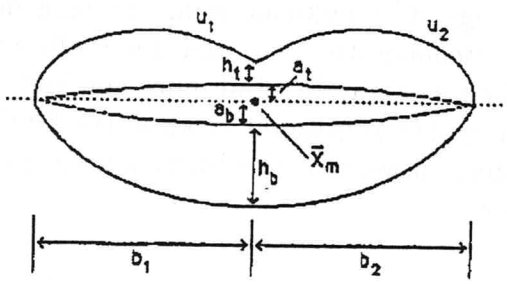


Abbildung 4.5: Aus: Facial feature extraction by deformable templates by Alan Yuille, David Cohen and Peter Hallian, Havard Robotics Laboratory TR 88-2

### 4.2.5 Bimodale zeitliche Synchronisation

Wie schon in Abschnitt 3.2 erläutert, ist es sehr wichtig, nicht die zeitliche Korrelation zwischen Ton und Bild bei der Klassifizierung zu verlieren. Die akustische Vorverarbeitung liefert genau jede 10 msec einen n-dimensionalen Merkmalsvektor. Die visuelle Vorverarbeitung liefert im Durchschnitt etwa alle 33 msec einen m-dimensionalen Merkmalsvektor. ( Die visuellen Algorithmen beschränken sich jeweils genau auf ein Bild. Wann dieses Bild aufgenommen wurde ist für den Algorithmus irrelevant.) Wir wollen nun auch auf der visuellen Seite genau jede 10 msec einen Merkmalsvektor produzieren. Da wir den genauen Aufnahmezeitpunk jedes Bildes gespeichert haben, können wir die visuellen Merkmalsvektoren resynchronisieren. Zwei sehr einfache Algorithmen werden alternativ dafür verwendet:

1. Für jeden akustischen Merkmalsvektor berechnen wir die originale Aufnahmezeit. (Kurz bevor wir den Aufnahme-Befehl an den akustischen A/D Wandler geschickt haben, wurde die genaue Systemzeit abgespeichert. Wir gehen davon aus, daß die Taktrate des A/D Wandlers von 16 KHz sehr konstant ist. Ein zusätzlicher zeitlicher "Offset" wird zur akustischen Zeit addiert, da der A/D-Wandler eine gewisse "Vorlauf-Zeit" benötigt.) Zu diesem akustischen Merkmalsvektor ordnen wir genau den visuellen Merkalsvektor zu, der zeitlich am nächsten zur akustischen Aufnahmezeit liegt. Dies führt dazu, daß derselbe visuelle Vektor im Durchscnitt zu drei verschiedenen akustischen Vektoren zugeordnet wird.

2. Als Alternative linear-interpolieren wir die visuellen Vektoren zu Merkmalsvektoren im 10 msec Takt. Wir nehmen dabei an, daß die Merkmalsraum-Gerade zwischen zwei legalen Vektoren nur aus legalen Merkmalsvektoren besteht. Diese Annahme kann falsch sein, aber als Approximation reicht uns dieses Verfahren aus. ( In [6] wird als "Abfallsprodukt" eines Lernverfahrens ein anderes Interpolationsverfahren vorgeschlagen.)

## 4.3 Klassifizierung

Unser Signal ist nun vorverarbeitet und liefert im 10 msec Takt einen n-dimensionalen akustischen Merkmalsvektor und einen m-dimensionalen visuellen Merkmalsvektor. Von nun an betrachten wir diese Merkmalsvektoren ganz allgemein als Sprach-Merkmale und wenden darauf Verfahren an, die

sich in der rein akustischen Sprachverarbeitung sehr gut bewährt haben.

Als eine der erfolgversprechensden Ansätze gelten sogenannte hybride Ansätze, in denen die Phonem-Klassifizierung "konnektionistisch" realisiert ist, und die Wort/Satz-Klassifizierung mittels dynamischer Programmierung gelöst wird. In den letzten Jahren haben sich zwei sehr ähnliche Ansätze als sinnvoll herauskristallisiert: Die sogenannten Multi-State Time-Delay-Neural-Networks (MS-TDNN [11]) und die Multi-Layer-Perceptron/Hidden-Markov-Model Systeme (MLP/HMM [1]). Ganz grob unterscheiden sich beide Ansätze in zwei Punkten (auf die Einzelheiten wird in den folgenden Abschnitten detailierter eingegangen): Das MS-TDNN verwendet in der konnektionistischen Phonem-Klassifizierung ein sogenanntes "weight-sharing". Das MLP/HMM hat mächtigere Wort-Modelle (zusätzlich sind "transition-probabilities" möglich). Es gibt aber nicht genau einen MS-TDNN Ansatz und genau einen MLP/HMM Ansatz. Die berühmten Prozente für "State-of-the-art" Erkennungsraten hängen zusätzlich von verschiedenen Vorverarbeitungen, Variationen in dem Lernverfahren und Phonem-Kodierung, Anzahl freier Parameter und von zusätzlichen Heuristiken in der Wort-Klassifizerung ab.

Wir beschränken uns in dieser Arbeit auf den MS-TDNN Ansatz, erwähnen abschließend aber auch einige Experimente mit dem MLP/HMM Ansatz.

### 4.3.1 Time Delay Neural Network

Ein Time Delay Neural Network [28] (auf die Eindeutschung in "Zeitverzögerungs Neuronales Netz" wird verzichtet; im Folgenden wird es als TDNN bezeichnet) ist ein mehrschichtiges Perzeptron (MLP), das zusätzlich in der Lage ist, zeitliche Phänomene zu lernen.

Als sehr kurzer Exkurs (Genaue Definitionen und Motivationen können in [14] nachgelesen werden): Ein "klassisches" mehrschichtiges Perzeptron (MLP) besteht aus einer Eingabeschicht, mehrere "verdeckte"[3] Schichten und einer Ausgabeschicht. Jede Schicht besteht aus einer bestimmten Anzahl von analogen Einheiten. Alle Einheiten der Eingabeschicht sind mit allen Einheiten der ersten verdeckten Schicht verbunden. Dasselbe gilt zwi-

[3]Verdeckt (hidden) deshalb, da diese Schichten von "Außen" (Eingabe/Ausgabe) nicht gesehen werden.

schen der ersten und zweiten Schicht u.s.w. Die letzte Schicht ist mit der Ausgabeschicht in dieser Weise verbunden. Die Verbindungen sind von "Unten" (Eingabe) nach "Oben" (Ausgabe) gerichtet und unterschiedlich gewichtet. Jede Einheit einer Schicht hat ein analogen Ausgabewert, der eine Funktion der Eingabegewichte von der unteren Schicht und damit verbundenen Ausgabewerten der "unteren" Schicht ist (Z.B. "Sigmoidale Funktion"[4] angewandt auf die gewichtete Summe der Werte der verbundenen Einheiten.)

Im Kontext der Phonem-Klassifizierung enthalten die Einheiten der Eingabeschicht die normalisierten Koeffizienten der Sprachvorverarbeitung. Die Werte der Ausgabeschicht sollen dann Hypothesen für verschieden Phoneme berechnen (pro Phonem genau eine Einheit). Um dies zu erreichen, wird ein überwachtes Lernverfahren namens "Backpropagation" [26] auf die Gewichte angewandt. In der Eingabeschicht werden die Merkmalsvektoren von Beispiel-Phonemen angelegt, und in der Ausgabeschicht wird verlangt, daß die entsprechende Phonem-Ausgabeeinheit den Wert 1 hat und alle restlichen Ausgabeeinheiten den Wert 0 haben. Ziel des Backpropagation-Algorithmus ist es, iterativ den Fehler an der Ausgabeeinheit zu vermindern. Die anfänglich zufälligen Gewichte werden mittels Gradientenabstieg im Gewichtsraum (bezüglich einer Fehlerfunktion) "justiert". Prinzipiell ist dieses Verfahren eine Maximum-Likelihood Schätzung.

Soweit der kurze (unvollständige) Exkurs, wie man ein einfaches MLP auf Phonem-Klassifikation anwenden kann.

Als Unterschied zum konventionellen MLP hat im TDNN-Ansatz jede Verbindung zusätzliche Zeitverzögerungen mit verschiedenen Gewichten.

Ein TDNN benutzt üblicherweise eine verdeckte Schicht zur Phonem-Erkennung. Bild 4.6 zeigt das TDNN als Phonemklassifikator mit allen Aktivierungen über die Zeit verteilt. Man kann sich hier das TDNN auch als ein größeres Netz vorstellen, das auf die gesamte Merkmalsvektoren-Folge eines Wortes oder Satzes schaut. Die Ausgabe ist dann wieder eine zeitliche Folge von Phonem-Hypothesen-Vektoren. Jeder Ausgabe-Vektor wird mit den gleichen Gewichts-Kopien berechnet, was man als sogenanntes "weight-sharing" bezeichnet. (Würden die Zeitverzögerungen (time delays) nur am

[4]Eine sigmoidale Funktion ist eine differenzierbare symmetrische Funktion, die um den 0-Punkt annähernd linear ist, sich im positiven Bereich 1 und im negativen Bereich -1 annähert. Sie kann auch auf den Bildbereich 0-1 skaliert werden. Die Standard Sigmoid-Funktion ist: $\frac{1}{1+exp(-x)}$ mit $x = \sum_i w_i o_i$ wobei $w_i$ das enstprechende Gewicht zur Einheit $o_i$ in der unteren Schicht ist.

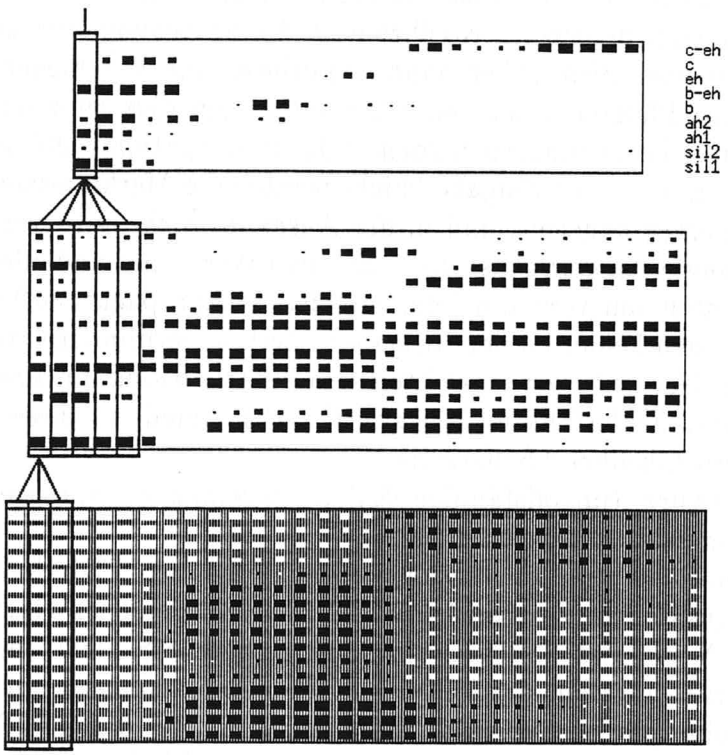


Abbildung 4.6: Aktivierungs-Szenario eines TDNN bei der Phonem-Erkennung

Eingang vorkommen, wäre es ein normales MLP, dessen Eingabeschicht auf ein Zeitfenster der Mermalsvektoren-Folge schaut und während Training und Erkennung über die gesamte Folge geschoben wird.)

In einer bahnbrechenden Arbeit von Waibel et al. [28] wurde gezeigt, daß die TDNN-Architektur sich hervoragend zur Phonem-Erkennung eignet. Durch das "weight-sharing" ist Segmentierung und Erkennung vereinigt. In der ursprünglichen Form des TDNN werden in der obersten Schicht alle Ausgabewerte zeitlich in entsprecheneede Phonem-Einheiten akkumuliert. Jede Phonem-Einheit hat also genau einen Wert für den gesamten Zeitraum im Eingabefenster. Für das Training ist keine zeitliche Segmentierung nötig und in der Erkennung ist dieser Ansatz relativ invariant zu zeitlichen Verschiebungen im Eingabefenster. In der MS-TDNN Erweiterung werden diese "Phonem-Akkumulatoren" durch dynamische Programmierung (oder Multi-State Einheiten) ersetzt (siehe unten).

Beste Ergebnisse wurden mit 3 Zeitverzögerungen a 10 msec in der Eingabeschicht und 5 Zeitverzögerungen a 10 msec in der verdeckten Schicht erreicht.

### 4.3.2 Phonem/Visem Klassen

Wir haben gesagt, daß wir die gleichen Klassifikations-Mechanismen auch auf die visuellen Merkmale anwenden. (Wie schon argumentiert, ist die gesprochene Sprache nicht durch das akustische Signal, sondern durch den Vokaltrakt definiert). Im Falle der rein visuellen Phonem-Klassifizierung würde das TDNN anstatt auf die akustischen Melscale Koeffizienten auf die Grauwert-Matrix oder 2D-FFT Koeffizienten schauen.

Die Frage ist nun, was wir genau unter visueller Phonem-Klassifizierung verstehen. Laut Lexikon sind Phoneme als Elemente einer Menge von kleinsten Spracheinheiten definiert, die es ermöglichen in einer Sprache oder einem Dialekt eine Äußerung von einer anderen Äußerung zu unterscheiden. (Webster: Member of the set of the smallest units of speech that serve to distinguish one utterance from another in a language or dialect.) Die Äußerung (utterance) muß in diesem Kontext akustisch sein (vocal expression, uttered by the voice).

Laut dieser Definition dürften wir nicht von visueller Phonem-Klassifizierung sprechen, was uns aber ziemlich gleichgültig ist. Das wirkliche Problem liegt in der aktuellen Wahl einer Phonemmenge. Es gibt in keiner

Sprache *die* Standard-Phonemmenge. Die Unterteilung hängt meistens von dem verwendeten Wortschatz und der Klassifikationstechnik ab. Variationen wie kontextabhängige Diphone, Triphone, "senones" u.s.w. bereichern zusätzlich die Auswahlmöglichkeiten.

In unserer Anwendung macht es jedoch Sinn die Menge der Klassen zu beschneiden. Die visuellen Sprachmerkmale geben nur über den sichtbaren Teil des Vokaltraktes Auskunft. Der "unsichtbare" Teil des Vokaltraktes wirkt jedoch in vielen Fällen stark unterschiedlich auf das Sprachsignal. Die visuellen Merkmale können deshalb nicht diskriminant genug sein, um eine gebräuchliche Phonemmenge erfolgreich zu klassifizieren. Als Beispiel: Die Phoneme /b/ und /p/ werden durch dieselben LippenbewegungenWort erzeugt. Der Unterschied ist die stimmhafte Anregung des Vokaltraktes für das /b/ und die stimmlose Anregung des Vokaltraktes für das /p/. Ein Erkenner kann als bestes Ergebnis 50 % Erkennungsrate auf der /b/ - /p/ Klassifikation erreichen. In unserem Falle würde im optimalen Fall das TDNN gleiche Ausgabeaktivität für beide Phonem-Einheiten erzeugen. Dies kann trotzdem sinnvoll sein, wenn das Netz auf eine komplette Phonem-Menge trainiert ist. Die Ausgabe kann dann im optimalen Fall folgendermaßen interpretiert werden: Mit aller Wahrscheinlichkeit ist es ein /p/ oder /b/, aber auf keinen Fall eines der anderen Phoneme. Der akustische Klassifikator kann dann unterscheiden, ob es wirklich ein /p/ oder ein /b/ ist.

Als alternative Phonemmenge definieren wir die sogenannten "Viseme". Das Design-Ziel für diese Menge ist es, dem visuellen Klassifikator die Chance zu geben, im theoretischen Fall an 100 % Erkennungsrate zu gelangen. Das heißt, Viseme sollen eine Menge von kleinsten Spracheinheiten sein, die es ermöglichen, eine Äußerung von einer anderen Äußerung auf Grund der visuellen Sprachmerkmale zu unterscheiden. Der Ausdruck "Viseme" ist in der Lippenlese-Literatur gebräuchlich. Im Falle des /p/- und /b/-Phonems gäbe es für beide Klassen genau ein /p_or_b/-Visem. Dem Trainingsalgorithmus für den Klassifikator hilft es in der Hinsicht, daß pro Klasse mehr Trainingsdaten vorhanden sind (/p_or_b/ hat bei gleicher Datenbank doppelt soviel Daten als /b/ oder /p/ getrennt).

Tabelle 4.1 zeigt die in dieser Arbeit verwendete Phonem-Kodierung bezüglich des deutschen Buchstabierproblems und die entprechende Visem-Kodierung.

| Wort | Phonem-Kodierung | Visem-Kodierung |
|---|---|---|
| sil | /si1/ - /si2/ | (si1) - (si2) |
| A | /#/ - /ahI/ - /ahF/ | (ahI) - (ahF) |
| B | /bI/ - /b-eh/ - /ehF/ | (b) - (b-eh) - (ehF) |
| C | /tI/ - /s/ - /s-eh/ - /ehF/ | (s) - (s-eh) - (ehF) |
| D | /dI/ - /d-eh/ - /ehF/ | (d) - (d-eh) - (ehF) |
| E | /#/ - /ehI/ - /ehF/ | (ehI) - (ehF) |
| F | /#/ - /aeI/ - /ae-f/ - /fF/ | (ehI) - (e-f) (f) |
| G | /gI/ - /g-eh/ - /ehF/ | (dI) - (d-eh) - (ehF) |
| H | /hI/ - /h-ah/ - /ahF/ | (hI) - (h-ah) - (ahF) |
| I | /#/ - /ieI/ - /ieF/ | (ieI) - (eiF) |
| J | /jI/ - /j-o/ - /o-t/ - /tF/ | (ieI) - (ie-o) - (d) |
| K | /kI/ - /k-ah/ - /ahF/ | (d) - (d-ah) - (ahF) |
| L | /#/ - /aeI/ - /ae-l/ - /lF/ | (ehI) - (e-l) - (l) |
| M | /#/ - /aeI/ - /ae-m/ - /mF/ | (ehI) - (e-m) - (b) |
| N | /#/ - /aeI/ - /ae-n/ - /nF/ | (ehI) - (e-d) - (d) |
| O | /#/ - /ohI/ - /ohF/ | (ohI) - (odF) |
| P | /pI/ - /p-eh/ - /ehF/ | (b) - (b-eh) - (ehF) |
| Q | /kI/ - /k-uh/ - /uhF/ | (d) - (d-uh) - (uhF) |
| R | /#/ - /aeI/ - /ae-r/ - /rF/ | (eI) - (ae-d) - (dF) |
| S | /#/ - /aeI/ - /ae-s/ - /sF/ | (eI) - (e-s) - (s) |
| T | /tI/ - /t-eh/ - /ehF/ | (d) - (d-eh) - (ehF) |
| U | /#/ - /uhI/ - /uhF/ | (uhI) - (uhF) |
| V | /fI/ - /f-au/ - /auF/ | (f) - (f-au) - (uF) |
| W | /vI/ - /v-eh/ - /ehF/ | (f) - (f-eh) - (ehF) |
| X | /#/ - /iI/ - /i-k/ - /k-s/ - /sF/ | (ieI) - (i-k) - (s) |
| Y | /#/ - /ueI/ - /p/ - /s/ - /i/ - /l/ - /o/ - /nF/ | (ueI) - (b) - (s) - (ie) - (l) - (oh) - (d) |
| Z | /tI/ - /s-t/ - /t-ae/ - /ae-t/ - /tF/ | (s) - (s-eh) - (eh-d) - (d) |

Tabelle 4.1: Visem-Kodierung

## 4.4 Wort-Klassifizierung mittels dynamischer Programmierung, MS-TDNN

Basierend auf den geschätzten TDNN-Phonem-Hypothesen wird auf der nächsten Ebene Wortklassifikation betrieben. Auf der Phonem-Ebene haben konstante zeitverzögerte Gewichte zur Repräsentation zeitlicher Korrellationen ausgereicht. Auf der Wortebene haben Faktoren wie Geschwindigkeitsänderungen in der Aussprache einen weitaus größeren Einfluß als auf der Phonemebene, was ein sogenanntes Zeitangleichen (Dynamic Time Warping) erfordert. Die Multi-State TDNN Architektur [12] erweitert die TDNNs um sogenannte "Multi-State Units". Für jedes zu erkennende Wort existiert eine MS-Einheit, die ein sogenanntes Wort-Modell repräsentiert. Das Wort-Modell besteht aus einer Folge von Phonemen. In der Erkennung wird mittels dynamischer Programmierung für jedes Wortmodell eine optimale Zuordnung von zeitlich hintereinander folgenden TDNN-Phonem-Hypothesen zum Wortmodell gefunden. In unserem Ansatz entspricht die Wortklassifikation genau dem Ansatz von Hermann Ney [23], dem sogenannten "One Stage Dynamic Time Warping" Algorithmus für kontinuierlich gesprochene Sprache. Bild 4.7 zeigt die Wort-Klassifikations-Schicht eines MS-TDNNs mit typischen Pfaden. Die "Distanzen" für das Dynamic Time Warping werden durch die Aktivierungen der zugehörigen Phonem-Hypothesen ersetzt. Die MS-Einheit besitzt als Aktivität die akkumulierten Phonem-Hypothesen. Der Pfad selbst wird so gewählt, daß die Akkumulation optimal ist.

Würde man die Ausgaben des TDNN zu Wahrscheinlichkeiten normieren und den Logarithmus bilden, entspräche eine MS-Einheit genau einem Spezialfall eines Hidden Markov Model (HMM), mit der Einschränkung, daß jede Statuseinheit des HMMs einen Übergang zu sich selbst mit konstanter Wahrscheinlichkeit von 0.5 hat und eine Übergang zur nächsten Statuseinheit mit konstanter Wahrscheinlichkeit von 0.5 hat. Das Dynamic Time Warping entspräche dem Viterbi-Algorithmus, da der Logarithmus der Hypothesen gebildet wird (die DTW-Summation entspricht einer Multiplikation auf den originalen Werten).

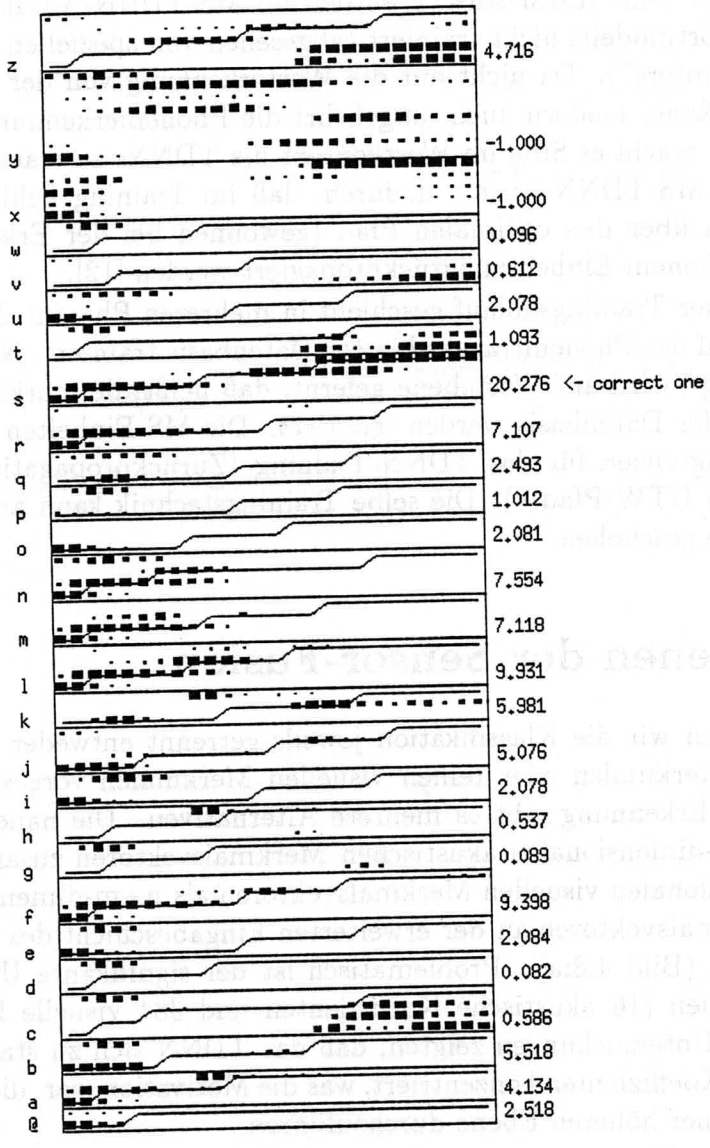


Abbildung 4.7: Dynamic Time Warping in der MS-TDNN Architektur

### 4.4.1 Integriertes globales Training

Im Gegensatzt zum HMM-Ansatz werden im MS-TDNN-Ansatz die Parameter der Wortmodelle nicht trainiert (abgesehen von speziellen "Duration-Control-Paramters"). Da nicht nur die Worterkennung von der Phonemerkennung abhängt, sondern auch umgekehrt die Phonemerkennung von dem Wortkontext, macht es Sinn im Wortkontext die TDNNs zu trainieren. Dies geschieht im MS-TDNN-Ansatz dadurch, daß im Training Fehler von den MS-Einheiten über den optimalen Pfad (gewonnen bei der Erkennung) in die TDNN-Phonem-Einheiten zurückpropagiert werden [12].

Ein üblicher Trainingsablauf geschieht in mehreren Phasen: Zuerst wird das TDNN auf der Phonemtranskribierten Datenbasis trainiert. Nach diesem "Bootstraping" wird auf Wortebene gelernt, daß heißt, die markierten Phonemgrenzen der Datenbasis werden ignoriert. Die MS-Einheiten generieren neue Phonemgrenzen für das TDNN-Training (Zurückpropagation entlang des optimalen DTW-Pfades). Die selbe Trainingstechnik kann anschließend auf Satzebene geschehen.

## 4.5 Ebenen der Sensor-Fusion

Bis jetzt haben wir die Klassifikation jeweils getrennt entweder auf reinen akustischen Merkmalen oder reinen visuellen Merkmalen vorgestellt. Für die bimodale Erkennung gibt es mehrere Alternativen. Die naheliegendste Idee ist, die n-dimensionalen akustischen Merkmalsvektoren zusammen mit den m-dimensionalen visuellen Merkmalsvektoren als n+m-dimensionale bimodale Merkmalsvektoren an der erweiterten Eingabeschicht des TDNN zu repräsentieren (Bild 4.8:a). Problematisch ist der signifikante Unterschied der Dimensionen (16 akustische Koeffizienten und 384 visuelle Koeffizienten). Unsere Untersuchungen zeigten, daß das TDNN sich zu stark auf die 384 visuellen Koeffizienten konzentriert, was die Motivation war, die "Sensor-Fusion" auf einer höheren Ebene durchzuführen.

Die nächste Ebene wäre auf der verdeckten Schicht. Das TDNN hat zwei verschiedene verdeckte Schichten (Bild 4.8:b). Eine Schicht ist nur mit den akustischen Mermalen verbunden und die andere nur mit den visuellen Merkmalen. Die beiden verdeckten Schichten werden dann in einer nächsten verdeckten Schicht oder in der Phonem-Schicht vereinigt. Die zwei getrenn-

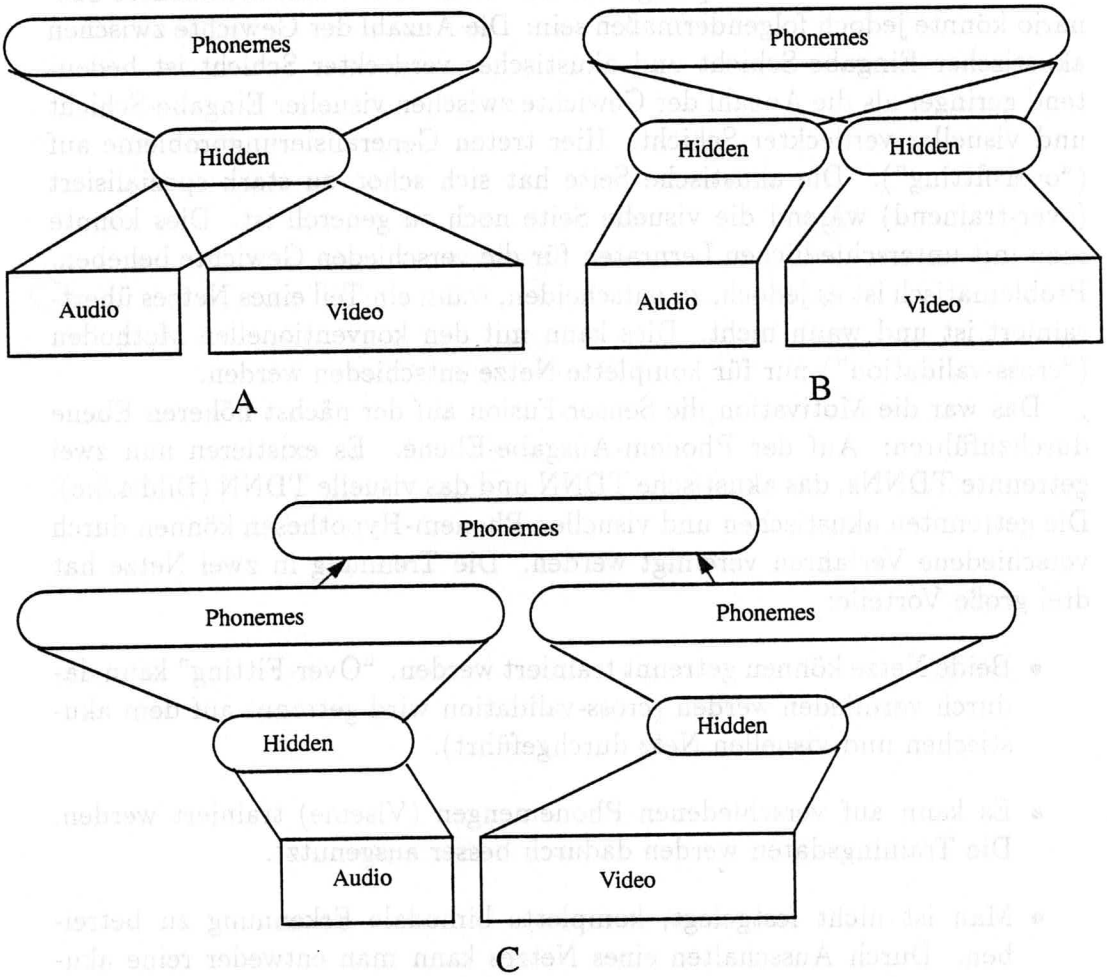


Abbildung 4.8: a: Sensorfusion in der Eingabeschicht, b: Sensorfusion in der verdeckten Schicht, c: Sensorfusion in der Ausgabeschicht

ten verdeckten Schichten sollten etwa gleiche Anzahl von Einheiten haben, um in der Fusion eine bessere Balance zu erreichen. Experimente mit dieser Topologie waren auch nicht erfolgreich. Ein Grund könnte eine nicht ausreichende Suche im "Topologie-Raum" sein. Das wahrscheinlichere Szenario könnte jedoch folgendermaßen sein: Die Anzahl der Gewichte zwischen akustischer Eingabe-Schicht und akustischer verdeckter Schicht ist bedeutend geringer als die Anzahl der Gewichte zwischen visueller Eingabe-Schicht und visueller verdeckter Schicht. Hier treten Generalisierungprobleme auf ("over-fitting"). Die akustische Seite hat sich schon zu stark spezialisiert (over-trainend) wärend die visuelle Seite noch zu generell ist. Dies könnte man mit unterschiedlichen Lernraten für die verschieden Gewichte beheben. Problematisch ist es jedoch, zu entscheiden, wann ein Teil eines Netzes übertrainiert ist und wann nicht. Dies kann mit den konventionellen Methoden ("cross-validation") nur für komplette Netze entschieden werden.

Das war die Motivation, die Sensor-Fusion auf der nächst höheren Ebene durchzuführen: Auf der Phonem-Ausgabe-Ebene. Es existieren nun zwei getrennte TDNNs, das akustische TDNN und das visuelle TDNN (Bild 4.8:c). Die getrennten akustischen und visuellen Phonem-Hypothesen können durch verschiedene Verfahren vereinigt werden. Die Trennung in zwei Netze hat drei große Vorteile:

- Beide Netze können getrennt trainiert werden. "Over-Fitting" kann dadurch vermieden werden (cross-validation wird getrennt auf dem akustischen und visuellen Netz durchgeführt).

- Es kann auf verschiedenen Phonemengen (Viseme) trainiert werden. Die Trainingsdaten werden dadurch besser ausgenutzt.

- Man ist nicht festgelegt, komplette bimodale Erkennung zu betreiben. Durch Ausschalten eines Netzes kann man entweder reine akustische Erkennung oder reine visuelle Erkennung betreiben, ohne neu gelernte Gewichtsmatrizen benutzen zu müssen. Abhängend von der Zuverlässigkeit der beiden Modalitäten können die akustischen und visuellen Hypothesen unterschiedlich gewichtet werden.

Die Fusion selbst kann unterschiedlich implementiert werden. In den ersten beiden Ansätzen (Eingabe-Schicht, verdeckte Schicht) ist sie implizit in den vollvernetzten Verbindungen zwischen zwei Schichten vorhanden.

Im letzten Ansatz (zwei getrennte TDNNs) geschieht die Fusion auch durch Gewichte. Eine zusätzliche bimodale Phonem-Hypothesen-Schicht ist über Gewichte mit den beiden TDNN Ausgabeschichten verbunden (Bild 4.8:c). Es darf sich aber nicht um eine Vollvernetzung handeln. Jede Einheit der bimodalen Phonem-Hyptothesen-Schicht hat genau zwei Eingabegewichte. Ein Gewicht von der entsprechenden akustischen Hypothese und ein Gewicht von der entsprechenden visuellen Hypothese. Falls das visuelle Netz auf Viseme trainiert worden ist, besteht die bimodale Phonem-Schicht aus der vollen akustischen Phonemmenge und einige unterschiedliche Phonemeinheiten haben eine Verbindung zur gleichen Visemeinheit. (Z.B. das bimodale /b/ ist mit dem akustischen /b/ und mit den visuellen /b_or_p/ verbunden, das bimodale /p/ ist mit dem akustischen /p/ und mit dem gleichen visuellen /b_or_p/ verbunden).

Als Alternative zu der gewichteten Verbindung könnte man für die bimodale Hypothese die akustische Hypothese mit der visuellen Hypothese multiplizieren [8]. Die Phonemausgaben müssen in diesem Falle Wahrscheinlichkeiten sein. Dafür könnte man die sogenannte "softmax"-Aktivierungsfunktion [7] verwenden. Unsere Experimente mit der Multiplikation führten jedoch zu keinen besseren Ergebnissen. In dieser Arbeit beschränken wir uns auf die gewichtete Verbindung.

### 4.5.1 Entropie Gewichte

Wie schon argumentiert sollte die Gewichtung in einer gewissen Weise dynamisch geschehen. Sie sollte von der Zuverlässigkeit der entsprechenden Modalität abhängen. Würde der Sprecher nicht zur Kamera schauen, oder seine Lippen wären verdeckt, oder er spricht ein Phonem aus, daß visuell sehr schwer zu schätzen ist, ist die visuelle Zuverlässigekeit sehr gering. Die Erkennung sollte in diesem Falle mehr auf der akustischen Schätzung basieren. Existiert jedoch sehr starkes Hintergrundrauschen, mehrere Sprecher sind "vermischt" (crosstalk), oder das ausgesprochene Phonem ist akustisch mehrdeutig, nimmt die akustische Zuverlässigkeit ab, und der Erkenner sollte mehr die visuelle Schätzung benutzen.

Wir wollen eine bimodale Gewichtung basierend auf der Zuverlässigkeit der beiden TDNN Hypothesen berechnen. Eine Schätzung hat maximale Zuverlässigkeit, wenn das Netz für genau ein Phonem die Ausgabe 1 erzeugt und für alle anderen Phoneme die Ausgabe 0 erzeugt. Bei minimaler Zu-

verlässigkeit würde das Netz für alle Phoneme die gleiche Ausgabe erzeugen. Ein gebräuchliches Maß ist die sogenannte Entropie. Sie basiert auf einem Satz von Wahrscheinlichkeiten und gibt Auskunft über den Informationsgehalt dieser Wahrscheinlichkeiten (hoher Informationsgehalt entspricht hoher Zuverlässigkeit). Wir normalisieren die Ausgabehypothesen der TDNNs zu Wahrscheinlichkeiten und berechnen die Entropie der akustischen TDNN-Ausgabe und die Entropie der visuellen TDNN-Ausgabe. Für die Normalisierung dividieren wir jede Hypothese mit der Summe aller Hypothesen. Die resultierenden "Wahrscheinlichkeiten" summieren zu 1. Die Entropie wird durch folgende Formel berechnet:

$$entropy(TDNN) = -\sum_{i=1}^{n} \frac{hyp_i}{\sum_{j=1}^{n} hyp_j} log \frac{hyp_i}{\sum_{j=1}^{n} hyp_j} \tag{4.3}$$

Die Gewichtung für jede bimodale Phonemhypothese ist gleich:

$$hyp_i^b = w_a hyp_i^a + w_v hyp_i^v \tag{4.4}$$

$$w_a = \frac{1}{2} + \frac{\text{entropy(visual-TDNN) - entropy(acoustic-TDNN)}}{2K} \tag{4.5}$$

$$w_v = 1 - w_a \tag{4.6}$$

$$\tag{4.7}$$

$K$ entspricht der maximalen Entropie in der Trainingsmenge oder im aktuellen Satz.

Die Gewichte $w_a$ und $w_v$ nennen wir sogenannte Entropie-Gewichte, da sie nicht durch Backpropagation gelernt werden, sondern dynamisch während der Erkennung basierend auf der Entropie bestimmt werden. Bild 4.10 zeigt einen typischen zeitlichen Verlauf der Gewichte wärend der Ausprache der Phonemsequenz /ae/ - /m/ - /#/ - /ie/ - /#/ - /eh/. Weiß bedeuted größer als $\frac{1}{2}$, schwarz bedeuted kleiner als $\frac{1}{2}$. Die Größe der Quadrate gibt Auskunft wie nahe das Gewicht an 1 oder 0 ist. Wie man sieht ist die "akustische Zuverlässigkeit" während dem /ae/-Phonem größer, aber die "visuelle Zuverlässigkeit" während dem /m/-Phonem größer.

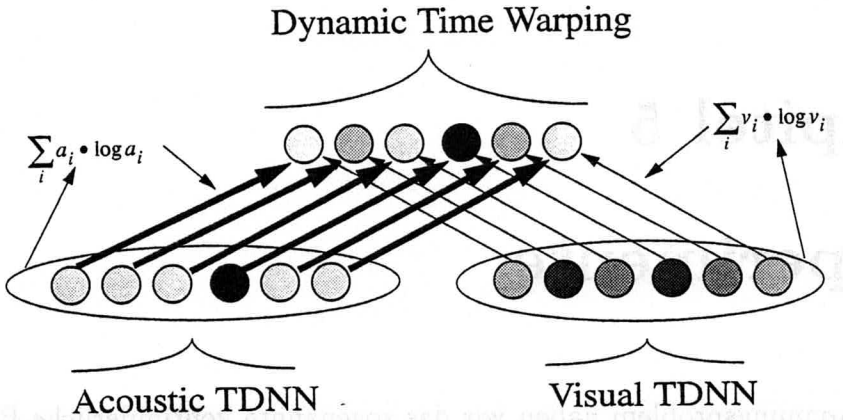


Abbildung 4.9: Bildliche Darstellung der Entropie-Gewichte

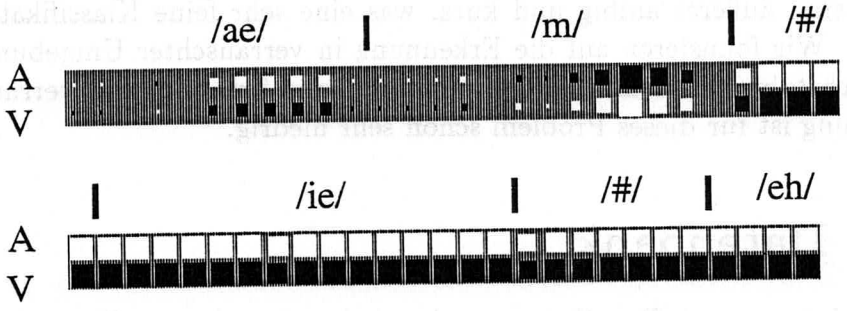


Abbildung 4.10: Typischer zeitlicher Verlauf der Entropie-Gewichte

# Kapitel 5

# Experimente

Als Erkennungsproblem haben wir das sogenannte kontinuierliche Buchstabieren gewählt. Das Alphabet besteht aus den 26 deutschen Buchstaben. Es kann jede beliebige Zufallssequenz buchstabiert werden. Wir können also keine linguistische Information in der Erkennung benutzen. Die Buchstaben selbst werden möglichst ohne Pause ausgesprochen. Wir sehen dieses Problem als allgemeines kontinuierliches Worterkennungsproblem. Die Wörter sind die Buchstaben (Y ist z.B. ein relativ langes Wort) und ein Satz ist die Buchstabensequenz. Verglichen zu einem Wortschatz mit längeren Wörtern und gleicher Perplexität ist das Buchstabierproblem sehr schwer. Die Buchstaben sind äußerst ambig und kurz, was eine sehr feine Klassifikation erfordert. Wir fokusieren auf die Erkennung in verrauschter Umgebung, was das Buchstabierproblem erschwert. Menschliche Erkennung in verrauschter Umgebung ist für dieses Problem schon sehr niedrig.

## 5.1 Datenbank

Die Trainings- und Test-Daten werden (wie schon kurz erläutert) folgendermaßen aufgenommen: Eine Versuchsperson setzt sich vor das Stand-Mikrophon und Kamera. Auf dem Computerbildschirm erscheinen in zeitlicher Abfolge deutsche Vornamen, Städtenamen oder Zufallssequenzen. Die Person drückt eine Taste, buchstabiert die Sequenz und drückt wieder eine Taste. Die Kamera ist auf die Lippen gerichtet und es wird wahlweise ein 256x256 Pixel Ausschnitt des gesamten Gesichtes aufgenommen oder ein

128x128 Pixel Ausschnitt der gesamten Lippen (die Aufnahmefrequenz ist ca. 30 Bilder pro Sekunde). Bild 5.1 zeigt die Lippen der in dieser Studie verwendeten zwei weiblichen und vier männlichen Versuchspersonen.

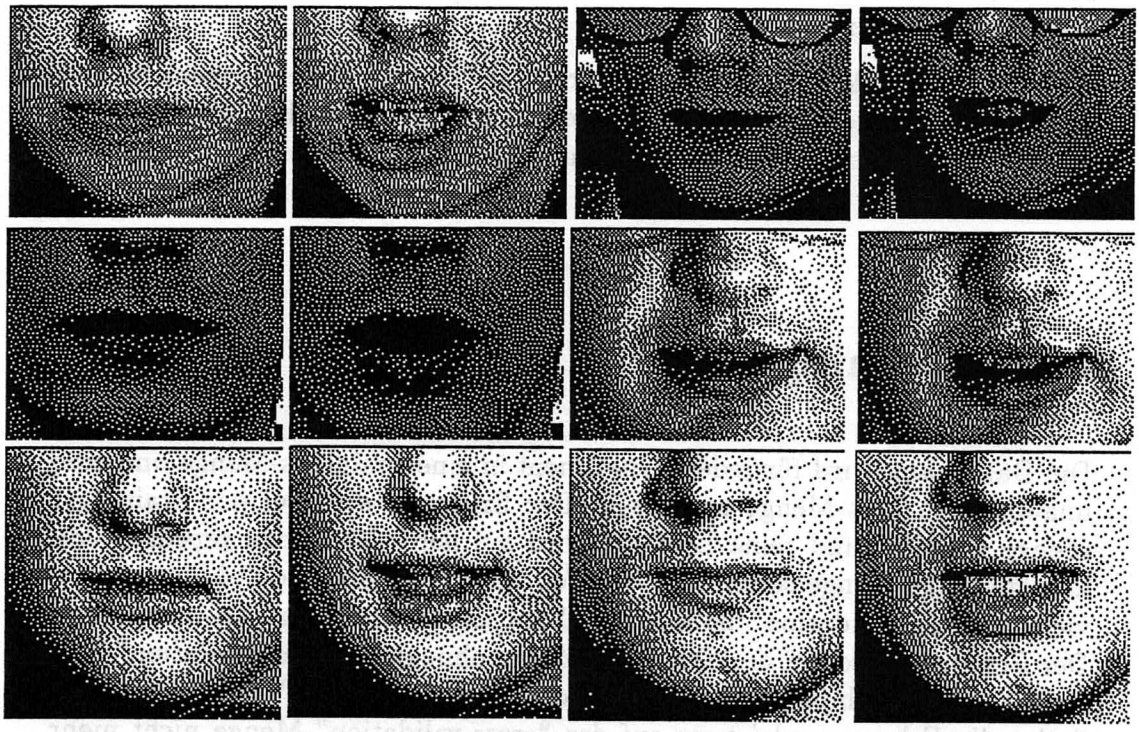

Abbildung 5.1: Typische Lippen der Datenbank

Experimente sind auf zwei sprecherabhängigen Datenbanken (msm, mcb) und einer "mehrsprecher" (multi-speaker) Datenbank (2f4m) durchgeführt worden. Die msm-Datenbank besteht aus 114 buchstabierten Sequenzen (859 Buchstaben) und die mcb-Datenbank besteht aus 350 buchstabierten Sequenzen (2549 Buchstaben). msm und mcb sind männliche Sprecher. Die 2f4m-Datenbank besteht aus sechs Sprechern (zwei weibliche und vier männliche Sprecher). Jeder Sprecher hat 100 Sequenzen buchstabiert (insgesammt 5695 Buchstaben). Verglichen mit Standard-Datenbanken wie z.B. der "Resource-Management-Database" erscheint diese Datenbank sehr klein. Der Grund für

diese "Bescheidenheit" liegt in der imensen Fülle von visuellen Daten. Die 2f4m-Datenbank belegt mehr als 1 GigaByte unserer Plattenkapazität. Die Aufnahme selbst ist schon sehr zeitraubend, da nach jeder buchstabierten Sequenz ein vielfaches der gesprochenen Zeit vergeht, bis alle Bilder abgespeichert sind. Üblicherweise dauert es über $1\frac{1}{2}$ Stunden, bis 100 Sequenzen aufgenommen wurden.

Wir beschränken uns deshalb zuerst auf "kleine" Datenbanken, bis wir zu einem sinnvollen Ansatz "konvergiert" sind und dann detaillierte Verfeinerungen auf einer größeren Datenbank untersuchen können.

## 5.2 Training

Die Datenbank wird für jedes Experiment in eine Trainings-Menge und eine unabhängige Test-Menge unterteilt. Der Erkenner wird auf der Trainings-Menge trainiert und seine Leistung wird auf der Test-Menge geprüft. Von der Trainings-Menge selbst wird ein Teil als "cross-validation"-Menge reserviert. "Cross-validation" ist ein Verfahren um "over-fitting" zu verhindern. Der Erkenner wird auf der Trainings-Menge ohne den "cross-validation"-Teil trainiert und nach jeder Iteration auf der "cross-validation"-Menge getestet. Steigt die Erkennungsleistung auf der "cross-validation"-Menge nicht mehr an, wird das Training beendet. (Würde das Training nicht an dieser Stelle beendet werden, würde der Erkenner zwar immer besser auf den Trainingsdaten werden, hat aber keine verbesserte Leistung auf allen anderen Daten. Er würde sich auf die Trainingsdaten zu stark spezialisieren und schlechter generalisieren.)

Das Training selbst ist in zwei Phasen unterteilt. Zuerst wird das TDNN direkt auf Phonem-Ziele trainiert ("bootstrapping"). Danach wird das MS-TDNN auf Wort-Ziele trainiert, d.h. das Zurückpropagieren geschieht von den MS-Einheiten über den optimalen Pfad (siehe 4.4.1) zurück in das TDNN. Beste Ergebnisse wurden mit dem McClelland-Fehlermaß für Phonem-Ziele erreicht und dem Classification Figure of Merrit für Wort-Ziele erreicht ([15]).

## 5.3 Verauschte Umgebung

Die Test-Menge wurde in verschiedenster Weiße künstlich "verschlechtert", um den Einsatz in verrauschter Umgebung zu demonstrieren. Wir haben dazu drei verschiedene Rauschquellen benutzt: Künstliches "weißes" Rauschen, originale Aufnahmen im Innenraum eines PKWs (VOLVO) in zwei verschiedenen Lautstärken, und Hintergrundsprechen, um den sogenannten "Cocktail-Party-Effekt" zu modellieren. Die Rauschquelle wurde mit dem reinen Sprachsignal additiv überlagert. Dies hat zwei Vorteile und einen Nachteil. Der erste Vorteil ist, wir können sehr exakt beliebige Signal-Rausch-Verhältnisse (SNR) generieren. Dies läßt sinnvolle Vergleichtests zu. Wir müssen nur einmal in einer rauscharmen Umgebung die Test-Menge aufnehmen und können sie nachträglich mit verschiedenen SNRs verrauschen. Der Nachteil ist jedoch, das Szenario ist nicht komplett realistisch. Hintergrundrauschen ist in der Realität zwar auch additiv, beeinflußt jedoch leicht die Aussprache des Sprechers. In stark verrauschter Umgebung wird meistens mit höherer Lautstärke und verschiedenem Grundton gesprochen, was die "Kanal-Charakteristik" des Sprechers leicht modifiziert.

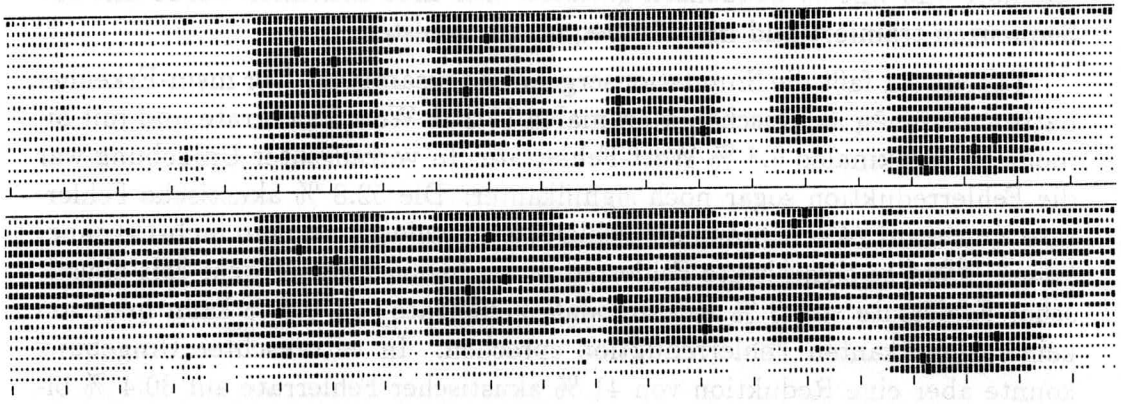

Abbildung 5.2: Melscale Koeffizienten in rauschfreier und verrauschter Umgebung

| Experiment | Akustisch | Visuell | Bimodal | Fehlerreduktion |
|---|---|---|---|---|
| msm/rauschfrei | 88.8 % | 31.6 % | 93.2 % | -39.3 % |
| msm/verrauscht | 47.2 % | 31.6 % | 75.6 % | -53.8 % |
| mcb/rauschfrei | 97.0 % | 46.9 % | 97.2 % | -6.7 % |
| mcb/verrauscht | 59.0 % | 46.9 % | 69.6 % | -25.8 % |

Tabelle 5.1: Worterkennungsraten (minus "Insertion" und "Deletion"-Fehler)

## 5.4 Sprecherabhängige Erkennung

In ersten Experimenten [3, 4] haben wir uns auf sprecherabhängige Erkennung beschränkt. Anfänglich stand kein gutes Transkribierprogramm und kein automatischer Lippenverfolger zur Verfügung, was das Hand-Segmentieren der Daten zu einem sehr zeitaufwendigen Prozess machte. Es wurden zwei sprecherabhängige Systeme trainiert (mcb-net und msm-net) und beide Systeme wurden auf rauschfreien und verrauschten Test-Daten geprüft ("weißes Rauschen", ohne SNR-Messung). Beste Ergebnisse wurden mit 15 verdeckten Einheiten im akustischen TDNN und 7 verdeckten Einheiten im visuellen TDNN erreicht. Der msm-Erkenner wurde mit 75 Sequenzen trainiert und mit 39 Sequenzen getestet. Der mcb-Erkenner wurde mit 200 Sequenzen trainiert und mit 150 Sequenzen getestet.

Tabelle 5.1 faßt die Simulationsergebnisse zusammen. Der msm-Erkenner hatte mit reinen akustischen Merkmalen 11.2 % Wort-Fehlerrate und mit bimodalen Merkmalen 6.8 % Wort-Fehlerrate. In verrauschter Umgebung war die Fehlerreduktion sogar noch signifikanter: Die 52.8 % akustische Fehlerrate wurde auf 24.4 % bimodale Fehlerrate reduziert. Bei dem mcb-Erkenner konnte nicht so signifikante Fehlerreduktion beobachtet werden: Mit akustischer Fehlerrate von 3 % auf bimodale Fehlerrate von 2.8 % kann man von keiner signifikanten Fehlerreduktion sprechen. In verrauschter Umgebung konnte aber eine Reduktion von 41 % akustischer Fehlerrate auf 30.4 % bimodale Fehlerrate beobachtet werden. Die Ergebnisse sind jedoch mit sehr großer Vorsicht zu betrachten. Die Datenbank ist sehr klein und die Varianz in Fehlerraten deshalb sehr groß. Die Ergebnisse sind jedoch trotzdem sehr ermutigend, speziell der Gewinn in verrauschter Umgebung.

| Experiment | Akustisch | Visuell | Bimodal | Fehlerreduktion |
|---|---|---|---|---|
| 2f4m/rauschfrei | 82.5 % | 12.2 % | 84.0 % | -8.6 % |
| 2f4m/30db SNR | 66.8 % | 12.2 % | 76.4 % | -28.9 % |
| 2f4m/20db SNR | 41.1 % | 12.2 % | 48.1 % | -11.9 % |
| 2f4m/15db crosstalk | 53.1 % | 12.2 % | 62.7 % | -20.5 % |

Tabelle 5.2: Mehrsprecher-Worterkennungsraten

## 5.5 Mehrsprecher Erkennung (multispeaker task)

Eine schwierigere Aufgabe ist die Mehrsprechererkennung. Wir benutzen zwei weibliche Sprecher und vier männliche Sprecher (2f4m). Die Trainingsmenge besteht aus 80 Sätzen pro Sprecher, die cross-validation-Menge besteht aus 10 Sätzen pro Sprecher und die Testmenge besteht aus 10 Sätzen pro Sprecher. Die Testmenge wurde diesmal mit unterschiedlichen Signal-Rausch-Verhältnissen erzeugt. Die Rauschquelle war das Innenraumrauschen eines Autos. In einem zusätzlichen Experiment war Hintergrundsprechen (crosstalk) vorhanden.

Tabelle 5.2 faßt die Simulationsergebnisse zusammen. Obwohl hier die reine visuelle Erkennung sehr schlecht ist, wurde trotzdem eine Fehlerreduktion mit der bimodalen Erkennung beobachtet.

Ähnliche Phänomene wurden mit einem anderen unabhängigen MLP/HMM-Ansatz beobachtet [5]. Dies bestärkt unseren Glauben, daß bimodale Erkennung speziell in verrauschter Umgebung sehr viel Sinn macht.

## 5.6 Analyse der Gewichtsmatrix

Abschließend wollen wir noch einen Blick auf die gelernten Gewichte werfen. In [28] wurde dies schon mit einem akustischen TDNN durchgeführt und in [24] wurde dies an einem MLP zur Fahrzeugsteuerung illustriert. In vielen Fällen ist es fast unmöglich isoliert die Funktion einer verdeckten Einheit zu interpretieren. Wir haben aber in manchen Fällen trotzdem einige interessante Gewichtskonstellationen bezüglich einer verdeckten Einheit beobachten

können, die es Wert sind, hier zu veranschaulicht zu werden.

Eine der signifikantesten visuellen Änderungen ist das Öffnen oder Zusammenpressen der Lippen. In dem Phonem /b/ oder /m/ geschieht dies am Anfang bzw. Ende.

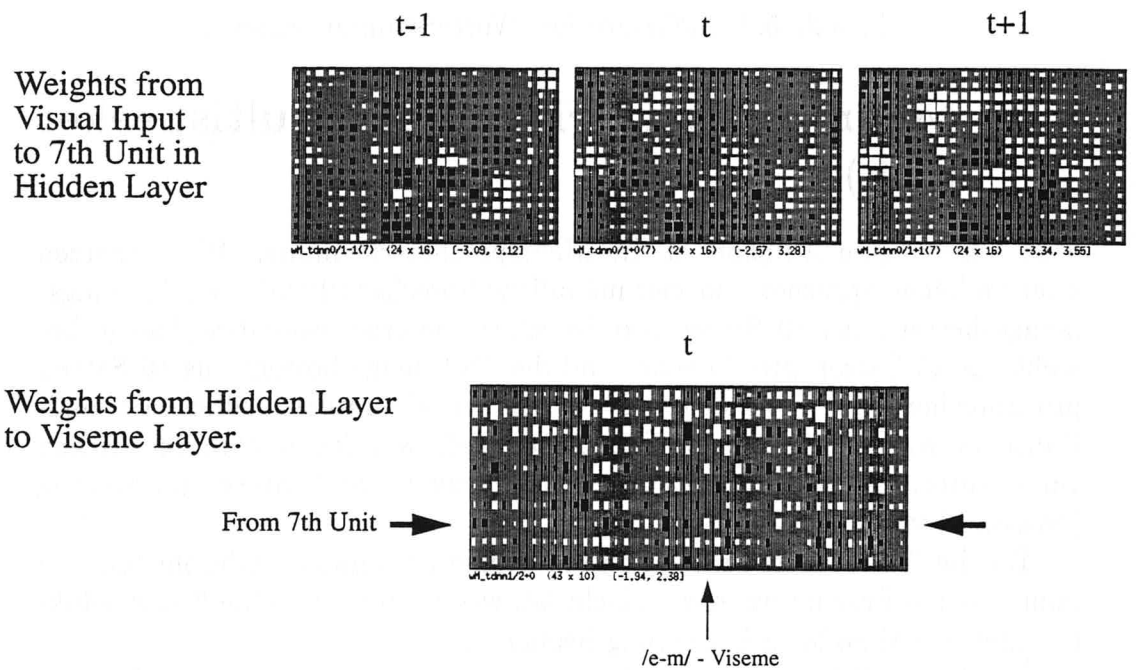


Abbildung 5.3: Gewichtsmatrix von der visuellen Eingabeschicht zur 7. verdeckten Einheit und Gewichtsmatrix von der 7. verdeckten Einheit zur Ausgabeschicht des TDNN

Bild 5.3 zeigt die Gewichtsmatrix der visuellen Eingabeschicht zu einer verdeckten Einheit und alle Gewichte dieser Einheit zu den Phonem-Einheiten. Schwarze Punkte entsprechen negativen Gewichten und weiße Punkte positiven Gewichten. Die Eingabegewichte sind geometrisch in gleicher Weise angeordnet wie sie zum Eingabe-AOI verbunden sind. Die drei Matrizen entsprechen jeweils den drei Zeitverzögerungen zu einer speziellen verdeckten Einheit. Es handelt sich hier um sprecherabhängige Gewichte.

Die Lippen des Sprechers haben niedrige Grauwerte und die Umgebung außerhalb der Lippen haben hohe Grauwerte. Die überwiegend negativen Gewichte in der ersten Zeitverzögerung erzeugen deshalb hohe Aktivierung für einen geöffneten Mund und die überwiegend positiven Gewichte in der dritten Zeitverzögerung erzeugen hohe Aktivierung für einen geschlossenen Mund. D.h. diese spezielle verdeckte Einheit hat hohe Aktivierung für Mundschließen. Dies wird auch durch ein hohes positives Gewicht von dieser Einheit zur /e-m/ Visem-Einheit bestätigt (/e-m/ ist Mundschließen). Zum /b_or_p/-Visem konnte ein negatives Gewicht beobachtet werden, andere Gewichte hatten jedoch auch starken Einfluß auf die Gesamtaktivierung.

# Kapitel 6

# Diskussion und Aussicht

In dieser Arbeit wurde gezeigt, wie visuelle Sprachinformation genutzt werden kann, um Spracherkennung robuster zu machen. Es wurde eine "state-of-the-art" Spracherkennungsarchitektur benutzt, um kontinuierlich gesprochenes Buchstabieren zu klassifizieren. Die visuelle Information der Lippenbewegungen wurde unterstützend zum akustischen Sprachsignal benutzt, um speziell in akustisch verrauschter Umgebung die Erkennung zu verbessern.

Im Gegensatz zu anderen Arbeiten, die entweder auf statischen Bildern, isolierten Phonemen, oder unrealistisch geringem Wortschatz Lippenlesen untersucht haben, wurde hier zum ersten mal demonstriert, daß kombinierte akustische und visuelle Erkennung für die reale Anwendung des kontinuirlichen Buchstabierens sehr sinnvoll ist. Es wurden verschiedene Alternativen für die visuelle Vorverarbeitung und die bimodale Sensorfusion illustriert. Als Klassifikator wurde ein modularer MS-TDNN Ansatz auf bimodale Erkennung modifiziert. Es wurden verschiedene Phonem-Klassen (Viseme) und dynamische Entropie-Gewichte untersucht.

Unsere Simulationen mit dem MS-TDNN Ansatz, MLP/HMM Ansatz und Experimente von anderen Gruppen charakterisiert folgendes grundlegendes Phänomen: Falls das akustisches Sprachsignal hohe Qualität hat, kann die Erkennungsleistung mittels Lippenlesen nicht signifikant verbessert werden. Sobald jedoch Qualitätsminderungen im akustischen Sprachsignal auftreten, trägt die visuelle Information sehr stark zur Leistungsverbesserung bei. In realistischen Anwendungen ist nie ein hochqualitatives akustischen Sprachsignal vorhanden. Störungen, wie z.B. Hintergrundrauschen, Telefonklingeln, Tastaturgeräusche, Ventilatoren, offenes Fenster, oder andere Stim-

men im Raum charakterisieren eine realistische Umgebung. Noch schwieriger wird es mit Anwendungen in Cockpits, PKWs, Werkshallen, oder allgemein in freier Natur. Existierende konventionelle Erkennungssysteme funktionieren meißt nur in rauscharmer Umgebung und sind deshalb nutzlos für all diese realistischen Anwendungen. Im Gegensatz zum akustischen Sprachsignal kann das visuelle Sprachsignal oft störfrei in hoher Qualität ermittelt werden. Ist der Sprecher z.B. vom Mikrophon weit entfernt, läßt die akustische Qualität drastisch nach, die Video-Kamera kann jedoch beliebig nahe an die Lippen "zoomen".

Obwohl wir schon sehr gute Erkennungsraten demonstrieren konnten, besitzt das hier vorgestellte System noch starken Prototyp-Charakter. Das gewählte Buchstabierproblem ist nur ein Teilproblem, die Datenbank selbst ist noch zu klein, um vernünftiges sprecherunabhängiges Training durchzuführen und der kritischste Teil des kompletten Systems, die visuelle Vorverarbeitung ist noch nicht sehr ausgereift. Andere Projekte befassen sich im Moment mit robuster Gesichtsverfolgung, Realzeit-Implementationen des kompletten Systems und Erweiterung der Datenbasis. In Planung ist eine größere Datensammelaktion für spontan gesprochene Dialoge mit offenem englischem Wortschatz und die Integration neuer visueller Klassifikatoren in ein komplettes Dialogsystem (Berkeley Restaurant Project BeRP). Untersuchungen sind im Gange, in wie weit existierende und zukünftige massiv parallele Systeme zur Realzeitimplementationen eines Lippenlesesystems verwendet werden können (CM-5, SPERT, CNS-1).

Das anfänglich sehr kleine und spezielle Gebiet der "Automatischen-Lippenlese-Forschung" hat inzwischen überraschend viele Anhänger gewonnen. Als ich zum ersten mal vor zwei Jahren in einem Gespräch mit David Stork von dieser Idee gehört hatte, sah die Sache sehr unrealistisch aus. Anfang 1992 wurde jedoch in großem Umfang durch das Land Baden Württemberg speziell dieses Problem in verschiedenen Instituten gefördert. Inzwischen sind zahlreiche Leute and der Universität Karlsruhe, an der University of California Berkeley und ICSI, und am Ricoh California Research Center mit ähnlichen Ansätzen beschäftigt. Es sind sogar Planungen im Gange einen internationalen Workshop über Lippenlesen zu organisieren, der die zahlreichen Gruppen aus dem Feld der psychologischen Studien, Animation und Erkennungssysteme aus Europa, USA und Japan zusammenbringen soll.

Multimodale System wie die hier vorgestellte bimodale Erkennung gewinnen immer mehr an Bedeutung. Neben robusteren Systemen entstehen völlig

neue Benutzerschnittstellen, die den Umgang mit der Maschine natürlicher gestalten und neue Dimensionen der Kommunikation eröffnen. Lippenlesen ist nur ein Teil davon. Andere Ansätze wie Gestikerkennung mit Stift oder Handzeichen, "eye-tracking", oder Gesichtsidentifikation sind andere Beispiele. Das Aufkommen neuer leistungsvollerer Systeme und Fortschritte in anderen Forschungsbereichen gibt diesem Bereich zusätzlichen Antrieb.

# Literaturverzeichnis


[1] H. Bourlard and N. Morgan. Merging multilayer perceptrons and hidden markov models: some experiments in continuous speech recognition. *Neural Networks: Advances and Applications*, 1991.

[2] M.H. Goldstein B.P. Yuhas and T.J. Sejnowski. Integration of Acoustic and Visual Speech Signals using Neural Networks. *IEEE Communications Magazine*, Nov. 1989.

[3] C. Bregler, H. Hild, S. Manke, and A. Waibel. Bimodal sensorfusion on the example of speechreading. In *Proc. ICNN*, 1993.

[4] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. In *Proc. ICASSP*, 1993.

[5] C. Bregler and Y. Konig. Eigenlips for robust speechrecognition. Submitted to ICASSP 94.

[6] C. Bregler and S. Omohundro. Surface learning with applications to lip-reading. To appear in NIPS 6.

[7] J. S. Bridle and S. J. Cox. Recnorm: Simultaneous normalisation and classification applied to speech recognition. In *Neural Information Processing Systems (NIPS 3)*, 1991.

[8] Greg Wolff David G. Stork and Earl Levine. Neural Network Lipreading System for Improved Speech Recognition. In *IJCNN*, June 1992.

[9] Barbara Dodd and Ruth Campbell. *Hearing by Eye: The Psychology of Lip-Reading.* Lawrence Erlbaum Associates, Publishers, 1987.

[10] D. Bodoff E. Petajan, B. Bischoff and N.M. Brooke. An Improved Automatic Lipreading System to enhance Speech Recognition. In *ACM SIGCHI*, 1988.

[11] P. Haffner, M. Franzini, and A. Waibel. Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, May 1991.

[12] P. Haffner and A. Waibel. Multi-State Time Delay Neural Networks for Continuous Speech Recogniton. In *Neural Information Processing Systems (NIPS 4)*. Morgan Kaufmann, April 1992.

[13] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. Rasta-plp speech analysis technique. In *Proc. ICASSP*, 1992.

[14] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Addison Wesley, 1991. Lecture Notes Volume I, Santa Fe Institute, Studies in the Science of Complexity.

[15] J.B. Hampshire II and A.H. Waibel. A novel objective function for improved phoneme recognition using time-delay neural networks. *IEEE Transactions on Neural Networks*, 1(2), 1990.

[16] M. Kaas, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. of the First Int. Conf. on Computer Vision*, 1987.

[17] Jim Keeler and David E. Rumelhart. A self-organizing integrated segmentation and recognition neural net. In *Neural Information Processing Systems (NIPS 4)*, 1992.

[18] J. Koehler, N. Morgan, H. Hermansky, and H. Hirsch. Integrating rasta-plp into speech recognition. In *Submitted to ICASSP*, 1994.

[19] K. Mase and A. Pentland. Lip reading: Automatic visual recognition of spoken words. In *Proc. Image Understanding and Machine Vision*. Optical Society of America, June 1989.

[20] D.W. Massaro and M.M. Cohen. Evaluation and integration of visual and auditory information in speech percpetion. *Journal of Experimentaal Psychology: Human Perceptionand Performance*, 9, 1983.

[21] Ofer Matan, Christopher J.C. Burgers, Yann Le Cun, and John S. Denker. Multi-digit recognition using a space displacement neural network. In *Neural Information Processing Systems (NIPS 4)*, 1992.

[22] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264, 1976.

[23] H. Ney. The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1984.

[24] Dean A. Pomerleau. Neural Network Perception for Mobile Robot Guidance. Technical Report CMU-CS-92-115 (Ph.D. Thesis), Carbegie Mellon University, 1992.

[25] Peter Rander. Facetracking using a template based pyramid approach. Personal Communications.

[26] D. E. Rummelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing*, chapter 8, vol. 1. MIT Press, Cambridge, 1986.

[27] A. Q. Summerfield. Audio-visual speech perception, lipreading and artificial stimulation. In M.E. Lutman and M.P. Haggard, editors, *Hearing Science and hearing disorders*. Academic Press, 1983.

[28] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, March 1989.

[29] A. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.