Universität Karlsruhe
Institut für Logik, Komplexität
und Deduktionssysteme
Am Fasanengarten 5
D-76128 Karlsruhe

Robuste Systemarchitekturen für maschinelles Lippenlesen

Diplomarbeit von

Uwe Meier

Betreuer:

Prof. Dr. Alex Waibel
Dr. Paul Duchnowski

angefertigt am

Computer Science Department
Interactive Systems Laboratories
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.

uwem@cs.cmu.edu
uwe@ira.uka.de

März 95

Zusammenfassung

In dieser Diplom-Arbeit werden zwei Systemarchitekturen zum robusten, automatischen Lippenlesen dargestellt und miteinander verglichen. Unter robustem Lippenlesen hat man sich dabei Systeme vorzustellen, die auch On-Line-Bedingungen wie unterschiedliche Lippengrößen, unterschiedliche Beleuchtungen und auch unterschiedliche Lippenpositionen innerhalb des Bildes bewältigen. Für das robuste Lippenlesen werden zwei unterschiedliche Ansätze untersucht. Zum einen kann dieses Problem durch eine entsprechende Vorverarbeitung der Bilddaten gelöst werden. Eine andere Möglichkeit besteht darin, eine Netzarchitektur zu entwerfen, in der diese Problematik direkt eingeht.

Kapitel 1 gibt eine Motivation für die Verwendung von Multimodalen Systemen. In Kapitel 2 wird ein Überblick über andere Arbeiten aus dem Gebiet Lippenlesen gegeben. Kapitel 3 beschreibt die dieser Arbeit zugrundeliegende Datenbasis, die Aufnahmeverfahren und verwendete Hardware. In Kapitel 4 wird die Integration eines Face Trackers und Lippenfinders in einen MS-TDNN basierten Lippenleser beschrieben und kurz auf die Kombinationsmöglichkeiten mit akustischen Erkennern eingegangen. Kapitel 5 beschreibt eine neue zeit- und positionsinvariante Netzarchitektur.

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Pittsburgh, den 06.03.1995

Uwe Meier

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Durch die immer schneller fortschreitende Entwicklung der Computer bezüglich Rechengeschwindigkeit, Speicher und der Anbindung an multimedia Geräte sind in den letzten Jahren auch die multimodalen Rechnerumgebungen [49] immer mehr ins Zentrum des Interesses gerückt. Die Kombination von Ton, Bild und Text zur Ausgabe ist bereits auf Multimedia-PCs erhältlich, über den sogenannten Daten-Highway sind multimedia Dokumente in unüberschaubarer Zahl zugänglich. Lediglich beim Zugang zu diesen Informationen hat sich nicht viel getan, oft sind die Datenbanken nur mit speziellem Wissen einer Datenbanksprache zugänglich, als einziges Eingabemedium steht nur die Tastatur zur Verfügung. Dieser Zugang soll durch multimodale Benutzerschnittstellen vereinfacht und jedermann zugänglich gemacht werden.

Ziel dabei ist es, die Einschränkungen, die die Tastatur als Haupteingabegerät für den Computer mit sich bringt, zu überwinden. Während die Vielfalt der Art und die Menge der Informationen immer größer wird, ist die Art und Weise des Zugriffs auf diese Informationen noch sehr beschränkt. Haupteingabemedien sind Tastatur und Maus. Informationen, die bei der Mensch-zu-Mensch Kommunikation genutzt werden, gehen dabei vollständig verloren. Es liegt allein in der Verantworung des Anwenders, alle zusätzlichen Informationen explizit per Tastatur einzugeben. Deshalb soll eine multimodale Rechnerumgebung geschaffen werden, in der ein natürliches Arbeiten mit dem Rechner möglich ist, d.h. es sollen die selben Kommunikationsmechanismen verwendet werden, die auch in der zwischenmenschlichen Kommu-

nikation verwendet werden: Sprachverarbeitung, Handschriftenerkennung, Gestikanalyse, Gesichts- und Augenverfolgung oder auch Lippenlesen.

In dieser Arbeit wird schwerpunktmäßig das maschinelle Lippenlesen betrachtet. Dabei sollen so wenig störende technische Hilfsmittel wie möglich eingesetzt werden, damit sich der Sprecher frei im Raum bewegen kann. Dadurch ergeben sich eine Reihe von Aufgabenstellungen, die durch eine geeignete Vorverarbeitung oder auch durch eine spezielle Systemarchitektur gelöst werden müssen, da das Gesicht und die Lippen der Sprecher im Videobild gesucht und verfolgt werden müssen.

Eine Anwendung für das maschinelle Lippenlesen findet sich zum Beispiel in Arbeitsumgebungen, in denen ein hoher Anteil von Hintergrundgeräuschen vorhanden ist oder auch mehrere Sprecher durcheinander reden (*cocktail-party-effect, cross-talking*). Diese Arbeitsumgebungen sind für einen rein akustischen Erkenner nicht optimal und die Erkennungsleistung der einzelnen Systeme kann durch zusätzliche visuelle Informationen verbessert werden. Das Problem des *cross-talking* löst der Mensch durch weitere Informationsquellen, wie z.B. der Lippenbewegung.

Daß der Mensch zusätzlich zur akustischen Information auch visuelle Informationen nutzt, läßt sich am *McGurk-Effect* [30] zeigen: Bei akustisch leicht verwechselbaren Signalen kann die Lippenbewegung zur Auflösung dieser Mehrdeutigkeit genutzt werden. Das Auftreten dieses Effekts wurde an folgendem Beispiel gezeigt: Auf einem Videoband wird das akustische Signal für das amerikanische 'ba' und das visuelle Signal für 'ga' vorgespielt. Der Zuhörer/Zuschauer versteht dabei in den meisten Fällen das Signal 'da'. Dieser Versuch zeigt, daß der Mensch, meist unbewußt, zusätzlich zum akustischen Signal auch das visuelle Signal nutzt, um leicht verwechselbare Signale zu unterscheiden.

Kapitel 2

Verwandte Arbeiten

Hier soll ein kurzer Überblick über verschiedene Arbeiten zum Thema maschinelles Lippenlesen gegeben werden.

Petajan [37]

Der erste bedeutende Versuch, die automatische akustische Spracherkennung mit einem Lippenleser zu kombinieren, wurde 1984 von Petajan unternommen. Als Task lag dabei ein sprecherabhängiger Wortschatz von 100 Worten zugrunde. Vier statische Merkmale wurden von jedem Frame extrahiert und durch eine *linear time-warping* Prozedur wurde das wahrscheinlichste Wort bestimmt. Durch die Kombinierung mit einem akustischen Einzelworterkenner wurde eine Verbesserung von 65 auf 75% erreicht.

Pentland und Mase [28]

parametrisierten das Bild der Lippenregion durch Berechnung des optischen Flußes in 4 Bereichen des Bildes, um die Bewegung von Gesichtsmuskeln zu berechnen. Diese Regionen wurden von Hand ausgewählt. Für die visuelle Erkennung wurde ein *Template Matcher* verwendet. Als Task wurden dabei 5 Ziffern verwendet. Bei der Erkennung von Sequenzen von 3 bis 5 Ziffern wurde bei 3 Sprechern eine durchschnittliche Erkennungsrate von 75% erreicht.

David Stork, Greg Wolff und Earl Levin [46]

entwickelten am Ricoh California Research Center einen maschinellen Lippenleser auf der Basis eines *time-delay neural network* (TDNN).

Dieses Netz wird, leicht modifiziert, sowohl zur akustischen wie auch zur visuellen Erkennung genutzt. In dem visuellen Netz werden 5 Merkmale pro Frame verwendet. Diese Merkmale werden aus 10 Reflektionspunkten gewonnen, die dem Sprecher ins Gesicht geklebt wurden. Zur akustischen Erkennung werden 14 MelScale Koeffizienten verwendet.

Zur Kombination der Ergebnisse des visuellen Netzes und des akustischen Netzes existieren 2 Ansätze:

1. Nach der Formel

$$p(C|A,V) = kp(C|A)p(C|V)$$

   Dabei ist k ein konstanter Normalisierungsfaktor, $p(C|A)$ ist die Wahrscheinlichkeit, daß akustisch C erkannt wurde, $p(C|V)$ daß visuell C und $p(C|A,V)$ daß akustisch und visuell kombiniert C erkannt wurde.

2. Mit einem weiteren neuronalen Netzwerk, das die Ausgaben des akustischen und des visuellen Netzwerkes als Eingabe erhält.

Trainiert wurde das sprecherunabhängige Netz auf den 10 amerikanischen Buchstaben b, d, f, m, n, p, s, t, v und z. Dabei wurde eine rein visuelle Erkennungsrate von 51% und eine kombinierte Erkennungsrate von 91% erreicht. Bei verrauschten Daten wurde eine Verbesserung von rein akustisch 43% auf 75% akustisch und visuell kombiniert erreicht. Dabei wurde die Methode (1) zur Kombinierung verwendet.

Ben P. Yuhas, Moise H. Goldstein Jr., Terrence J. Sejnowski [53]

untersuchten in einem sprecherabhängigen Ansatz die Integration des visuellen und akustischen Signals zur Erkennung von 9 Vokalen. Dabei wurde zur visuellen Erkennung ein handsegmentiertes Einzelbild verwendet. Aus diesem Bild wird dann durch ein *multi layer perceptron* (MLP) das *short time spectrum* geschätzt.

Die Kombination der visuellen und akustischen Ergebnisse erfolgt nach folgender Formel:

$$\alpha S_{visuell} + (1-\alpha) S_{akustisch}$$

$S_{visuell}$ und $S_{akustisch}$ sind dabei die Aktivierungen der Ausgabeneuronen des Netzes. Für α wurde bei einer SNR[1] im Bereich von -12 dB bis 24 dB empirisch folgender Wert ermittelt:

$$\alpha = 0.535 - 0.22S/N$$

Bei der Erkennung auf verrauschten Daten wurde dabei eine Verbesserung von 11% rein akustisch auf bis zu 55% kombiniert erreicht.

Alan J. Goldschen [14]

beschreibt in seiner Dissertation einen *Hidden Markov Model* (HMM) basierten sprecherabhängigen Lippenleser. Als Task wurden dazu 450 Sätze (300 Sätze als Trainings- und 150 als Testset) verwendet. Die Sätze wurden dabei als ganze Einheit betrachtet, die Erkennung erfolgte also nicht auf Wortebene sondern der Satz wurde als ganzer erkannt. Dabei wurden insgesamt 35 Merkmale (22 statische und 13 dynamische) verwendet. Es wurde eine Erkennungsrate von 25.3% erreicht. Die Bilddaten wurden dabei mit einer am Kopf befestigten Kamera aufgenommen.

Peter L. Silsbee and Alan C. Bovik [44]

entwickelten an der Universität Texas einen sprecherabhängigen HMM basierten Lippenleser. Als Task wurden dabei die amerikanische Vokale verwendet. Bei Kombination mit einem akustischen HMM Erkenner wurde eine Fehlerreduktion von 30-60% erreicht. Zur visuellen Vorverarbeitung wurde eine Variante des Vektor-Quantisierungs-Algorithmus verwendet. Dabei wurden Bilder der Auflösung 80x80 Pixel auf 32 Symbole abgebildet.

Javier R. Movellan [33]

verwendet für das maschinelle Lippenlesen einen HMM-Erkenner. Als Task wurden dabei die Ziffern 1 bis 4 verwendet, die von 12 verschiedenen Sprechern gesprochen wurden. Die relevanten Frames der Graustufenbilder wurden dabei von Hand segmentiert. Folgende Vorverarbeitungsschritte werden bei diesem Ansatz angewand:

[1]Signal to Noise Ratio (S/N) = Signal Rausch Verhältnis

- Symmetriebildung: Die korrespondierenden rechten und linken Pixel des zentriert aufgenommenen Bildes werden gemittelt. Dadurch läßt sich die Eingabedimension halbieren.
- Delta-Bilder: Die Pixeldifferenz der Folgebilder wird berechnet
- Eine nicht näher beschriebene Skalierung / Schwellwertbildung mit Hilfe der logistischen Funktion

Bei diesem Ansatz wurde eine visuelle Erkennungsrate auf einzelnen Ziffern von 89 % erreicht. Die Kombination mit akustischen Daten wurde dabei nicht untersucht.

R.R. Rao und R.M. Merserau [41]

verwenden in ihrem Lippenleser eine automatische, rotations- und größeninvariante Lippenmodelierung, basierend auf *deformable Templates*. Es wird ein zwei Worte Task ('wow','mom') zur Erkennung verwendet. Mit *dynamic time warping* wird eine Erkennung von 85% erreicht.

Diplom-Arbeit In dieser Diplom-Arbeit wird ein sprecherabhängiger Ansatz zur Erkennung des deutschen Alphabeths untersucht. Als Netzarchitektur werden dafür ein MS-TDNN und ein MS-TDNN^{3d} verwendet. Zum Training werden dafür 140 kontinuierlich buchstabierte Sequenzen benutzt, für Test und Crossvalidation werden je 30 Sequenzen benutzt. Es wurde sowohl auf Daten mit manuell extrahierten Lippen wie auch auf Daten mit automatisch extrahierten Lippen trainiert und getestet.

Die Tabellen 2.1, 2.2 und 2.3 geben eine Überblick über die verschiedenen Ansätze und einen Vergleich mit dem in dieser Diplom-Arbeit verwendeten Erkenner. Beim Vergleich der erzielten Erkennungsquoten sind jedoch die sehr stark voneinander abweichenden Tasks zu beachten.

	Task	kontinuierlich	sprecherunabhängig
Petajan	100 Worte	nein	nein
Pentland	5 Ziffern	ja	3 Sprecher
Stork	10 Buchstaben	ja (?)	ja
Yuhas	9 Vokale	?	nein
Goldschen	450 Sätze	nein	nein
Silsbee	Vokale	?	nein
Movellan	4 Ziffern	nein	12 Sprecher
Rao	2 Worte	nein	?
Diplom-Arbeit	26 Buchstaben	ja	nein

Tabelle 2.1: Übersicht der verschiedenen Lippenleser

	Architektur	Lippenextraktion
Petajan	linear time warping	?
Pentland	template matcher	von Hand ausgewählte Bildregionen für optischen Fluß
Stork	TDNN	10 Reflektionspunkte im Gesicht
Yuhas	MLP	?
Goldschen	HMM	am Kopf befestigte Kamera
Silsbee	HMM	?
Movellan	HMM	manuelle Zentrierung der Lippen bei der Aufnahme
Rao	dynamic time warping	deformable templates
Diplom-Arbeit	MS-TDNN und MS-TDNN^{3d}	automatische Gesichtsfindung automatische Lippenextraktion

Tabelle 2.2: Übersicht über die Architekturen verschiedener Lippenleser

	visuell	akustisch	kombiniert
Petajan	?	65%	75%
Pentland	75%	–	–
Stork	51% 51%	? 43%	91% 75%
Yuhas	?	11%	55%
Goldschen	25%	–	–
Silsbee	Fehlerreduktion 30-60%		
Movellan	89%	–	–
Rao	85%	–	–
Diplom-Arbeit	55% 55%	97% 58%	98% 80%

Tabelle 2.3: Übersicht der Ergebnisse verschiedener Lippenleser

Kapitel 3

Datenbasis

3.1 Hardware

Für das Sammeln der visuellen Daten stand in Karlsruhe eine DECstation 5000/200 mit einem Framegrabber (DECVideo), in Pittsburgh eine DEC ALPHA mit Framegrabber (J300) zur Verfügung. Zum Aufnehmen der akustischen Daten wurde ein Analog/Digital Wandler der Firma Gradient (Desklab) verwendet. Akustische und visuelle Daten wurden parallel zueinander aufgenommen. Aufgrund unterschiedlicher Frameraten war ein nachträgliches Synchronisieren der Daten notwendig. Da die Framerate der visuellen Daten ca. 3.5 mal niedriger ist als bei den akustischen Daten, werden jedem visuellen Frame 3-4 akustische Frames zugeordnet.

Zur Aufnahme der akustischen Daten wurde ein Standard-Mikrophon, für die visuellen Daten ein herkömlicher Camcorder (NTSC-Kamera) verwendet.

3.2 Akustische Daten

Die akustischen Daten wurden mit einer Abtastrate von 16 KHz aufgenommen. Die Routinen zur Vorverarbeitung (Fourier-Transformation, Berechnung der MelScale-Koeffizenten) wurden von einem akustischen Buchstabiererkenner [19] übernommen.

Visuelle Vorverarbeitungen	
Methode	Anzahl der Paramter
Graustufenbilder	384
Principal Components	32
Linear Discriminant Analysis	32
2D FFT - 17x17 Amplituden	289
Sektor	36
Ring	29
Slit	36

Tabelle 3.1: Visuelle Vorverarbeitungsmethoden

Alle Daten wurden in ruhiger Umgebung (ohne Hintergrundgeräusche) aufgenommen. Die neuronalen Netze wurden mit diesen sauberen Daten trainiert. Zum Testen der Netze wurden auch künstlich verrauschte Daten verwendet:

- weißes Rauschen (8 und 16 dB SNR)
- Radiomusik (10 und 16 dB SNR)
- Motorengeräusche (16 und 25 dB SNR)

3.3 Visuelle Daten

Zur Aufnahme der visuellen Daten wurden 2 Verfahren verwendet:

1. Aufnahme der Lippenregion
2. Aufnahme des Gesichts und anschließende automatische Extraktion der Lippenregion

Mit der zur Verfügung stehenden Hardware konnten ca. 30 Graustufenbilder (8-Bit) pro Sekunde aufgenommen werden. Als Vorverarbeitung stehen die in Tabelle 3.1 aufgelisteten Methoden zur Verfügung [31].

3.3.1 Aufnahmeverfahren 1

Bei diesem Aufnahmeverfahren muß sich der Sprecher so vor die Kamera setzen, daß seine Lippen in einem festen Rechteck der Größe 144x80 Pixel zentriert im Aufnahmefenster zu sehen sind.

Bedingt durch die Positionierung 'von Hand' ergeben sich in den Daten große Variationen bezüglich Größe der Lippen und auch der genauen Lippenposition innerhalb des Aufnahmefensters.

Dieses Verfahren wurde bereits in früheren Arbeiten [5] [31] verwendet.

3.3.2 Aufnahmeverfahren 2

Um dem Sprecher eine größere Bewegungsfreiheit zu ermöglichen und um eine konsistente Positionierung der Lippen zu gewährleisten, werden bei diesem neuen Verfahren der Face Tracker von Martin Hunke [21] und der Lippenfinder von Dietrich Büsching [8] eingesetzt.

Zusätzliche Hardware: Bei diesem Verfahren wird die Kamera auf eine sogenannte Pan Tilt Unit (PTU) montiert. Der Face Tracker verfolgt den Sprecher im Raum und steuert die Motoren der PTU so an, daß der Sprecher immer zentriert im Kamerabild zu sehen ist. In Karlsruhe wird eine PTU der Firma Directed Perception mit einer NTSC Kamera von Sony verwendet, in Pittsburgh eine NTSC Kamera von Canon mit eingebauten Schrittmotoren.

Vorgehensweise: Um eine möglichst hohe Framerate beim Face Tracker zu erzielen, wird dieser auf einem eigenen Rechner mit Framegrabber gestartet. Dazu wird die Kamera sowohl an den Rechner mit dem eigentlichen Aufnahmesystem wie auch an den Rechner mit dem Face Tracker angeschlossen. Das Aufnahmeprogramm und der Face Tracker werden dann über ein Socketinterface miteinander (asynchron) gekoppelt, um die Positionen der Gesichter auszutaschen. Mit diesem System wird bei einer Aufnahme das komplette Gesicht aufgenommen. Die Lippenregion wird dann in einem Nachbearbeitungschritt durch den Lippenfinder extrahiert.

Dieses Verfahren wird in Kapitel 4.2 genauer beschrieben.

Datenbank	
Sprecher	Aufnahmebedingung
mum1, mum2	je 100 Sequenzen unter 'optimaler' Bedingungen. Dabei wird unter der optimalen Beleuchtung eine gut ausgeleuchtete Gesichtsaufnahme verstanden.
mum3, mum5	je 100 Sequenzen mit stark variierenden Beleuchtungsverhältnissen
mum4	100 Sequenzen unter konstanter, nicht optimaler Beleuchtung
mum6, mum7, mum8	je 100 Sequenzen nur unter Verwendung der Zimmerbeleuchtung
mum9, mum10	wie mum6 - mum8. Dabei wurden allerdings größere Bildausschnitte aufgenommen, um eine künstliche Verschiebung zu ermöglichen.
mum15 - mum23	mit Aufnahmeverfahren 2 in Karlsruhe aufgenommene Daten
mum25 - mum28	mit Aufnahmeverfahren 2 an der CMU aufgenommene Daten

Tabelle 3.2: Beschreibung der verwendeten Datenbanken

3.4 Die Datenbank

Ein Überblick über die zum Training verwendeten Daten wird in Tabelle 3.2 gegeben. Beispiele für die visuellen Daten sind in Abbildung 3.1 und 3.2 dargestellt.

Alle Daten wurden von demselben Sprecher unter verschiedenen Aufnahmebedingungen aufgenommen. Als Task wurde dabei das deutsche Alphabet verwendet. Für das Labeln der akustischen Daten wurden kontextabhängige Phoneme verwendet. Diese Labels wurden automatisch von einem sprecherunabhängigen Buchstabiererkenner [19] erzeugt. Für das Trainieren von rein visuellen Netzen wurden kontextabhängige Viseme[1] [13] verwendet. Dabei

[1] Visem = kleinste Spracheinheit die visuell unterschieden werden kann (visuelles Phonem).

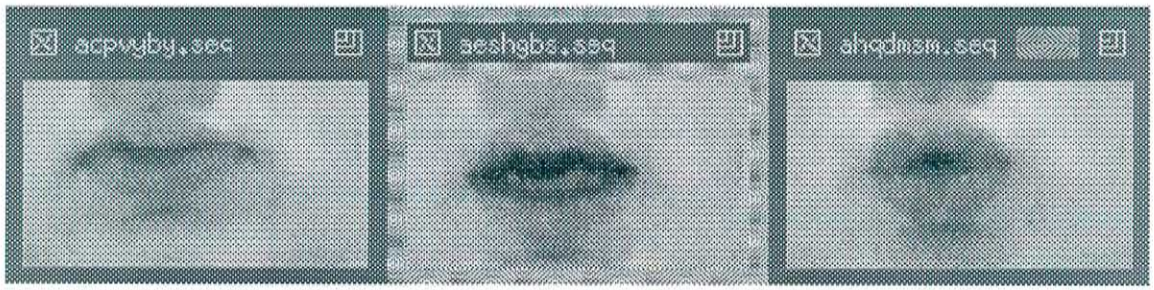

Abbildung 3.1: Beispiele aus der Datenbank mum1

existiert eine $n : m$ Abbildung ($n > m$) von Phonemen auf Viseme. Ein bekanntes Beispiel dafür sind die Phoneme /b/, /p/ und /m/, die visuell nicht unterschieden werden können und somit ein Visem bilden. Tabelle 3.4 zeigt die phonetische Beschreibung der Buchstaben und die entsprechende Beschreibung mit Visemen und Tabelle 3.3 die Zuordung der Phoneme zu den Visemen. Diese Viseme wurden von Hand (vor dem Spiegel) bestimmt. Zur Zeit wird im Rahmen einer Studienarbeit an einem Verfahren gearbeitet Viseme automatisch zu generieren [45].

Visem	Phoneme	Visem	Phoneme	Visem	Phoneme
_siI	si1	_dI	gI	_uhF	uhF
_siF	si2	_hI	hI	_eI	aeI
_ahI	ahI	_h-ah	h-ah	_ae-d	ae-r
_ahF	ahF	_ieI	ieI jI iI	_dF	rF
_b	bI mF pI p	_ieF	ieF	_e-s	ae-s
_b-eh	b-eh p-eh	_ie-o	j-o o-t	_uhI	uhI
_ehF	ehF	_d-ah	k-ah	_f-au	f-au
_s	tI s sF k-s	_e-l	ae-l	_uF	auF
_s-eh	s-eh s-t t-ae	_l	lF l	_f-eh	v-eh
_d	dI tF kI nF	_e-m	ae-m	_i-k	i-k
_d-eh	d-eh g-eh t-eh	_e-d	ae-n	_ueI	ueI
_ehI	ehI	_ohI	ohI	_ie	i
_e-f	ae-f	_ohF	ohF	_oh	o
_f	fF fI vI	_d-uh	k-uh	_eh-d	ae-t

Tabelle 3.3: Phonem-Visem Abbildung

Buchstabe	Phoneme	Viseme
@	si1 si2	_siI _siF
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 _ieF
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 _ohF
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 3.4: Beschreibung des Alphabets mit Phonemen und Visemen

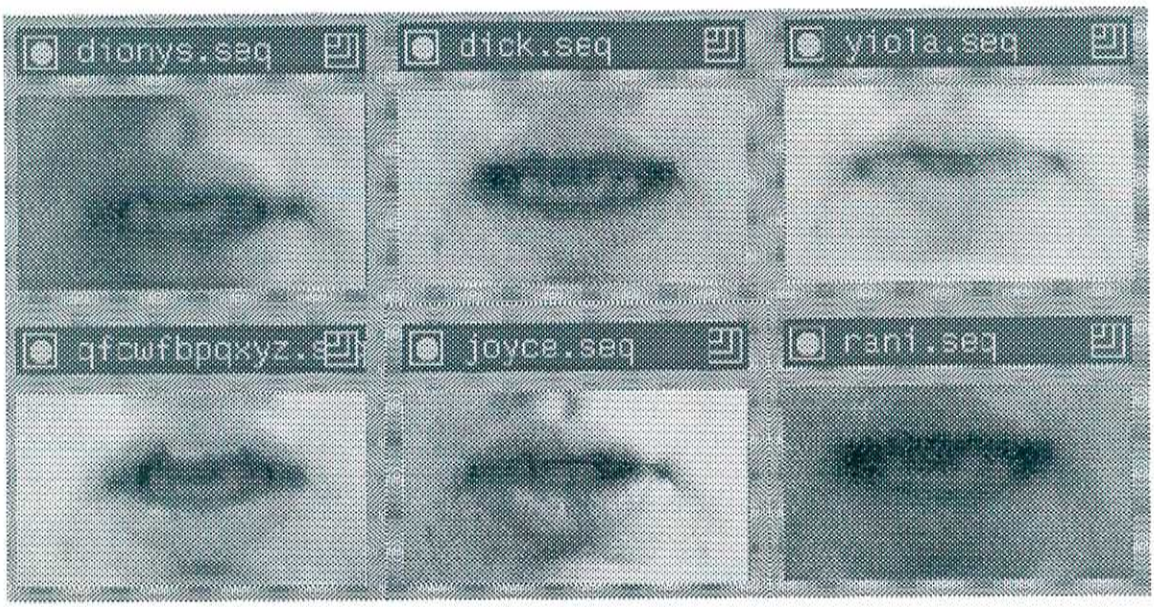

Abbildung 3.2: Beispiele aus der Datenbank mum3

3.5 Das Aufnahmesystem (Verfahren 2)

Das Aufnahemsystem besteht aus folgenden Teilprogrammen:

- Face Tracker
- Programm zum Aufnehmen und Abspeichern der visuellen und akustischen Daten
- Eine in TCL/TK[2] geschriebene Benutzerschnittstelle

Die einzelnen Teilsysteme, die zum Aufnehmen der Daten notwendig sind, werden in Kapitel 4 beschrieben. In Abbildung 3.3 wird das System schematisch dargestellt, Abbildung 3.4 zeigt 2 Screendumps der Benutzerschnittstelle des Aufnahmeprogramms.

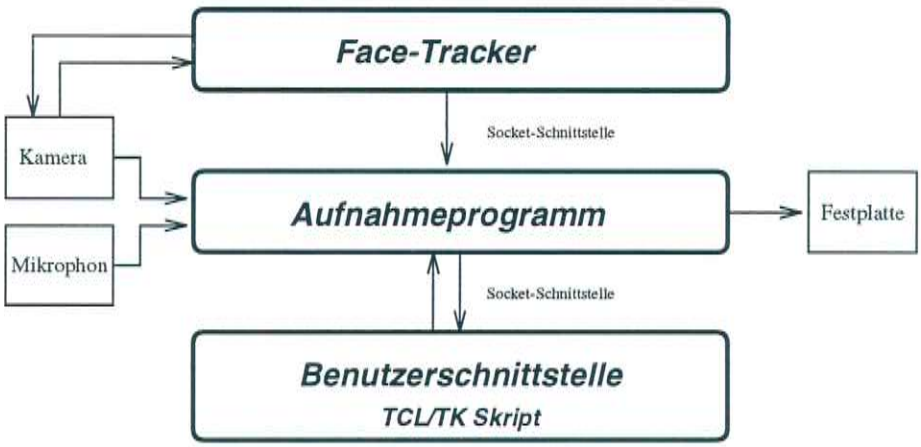

Abbildung 3.3: Schematische Übersicht des Aufnahmeprogramms

[2]TCL ist eine Interpreter-Sprache zur Erstellung von Shell-Skripten, TK ist eine Erweiterung dieser Sprache zur Programmierung von graphischen Benutzeroberflächen

Abbildung 3.4: Screendumps des Aufnahmeprogramms

Kapitel 4

Der MS-TDNN basierte Erkenner

4.1 Architektur

In diesem Kapitel wird dargestellt, wie ein *Multi-State Time-Delay-Neural-Network* (MS-TDNN) zum Lippenlesen verwendet werden kann. Dabei wird auch auf verschiedene Methoden der Kombinierung von akustischen und visuellen Daten, die sich bei dieser Architektur bieten, eingegangen.

Abbildung 4.1 zeigt den schematischen Aufbau eines MS-TDNN [51] wie es z.B. in akustischen Buchstabiererkennern [19] verwendet wird.

Die Idee dabei ist, ein Eingabesignal nicht nur zum Zeitpunkt t, sondern auch noch verzögert zu den Zeitpunkten $t+1, \ldots, t+(d-1)$ anzulegen. Man kann sich dies als ein Eingabefenster der Breite d vorstellen, das entlang der zeitlichen Achse über die Eingabeframes geschoben wird. Dadurch ist das Netz in der Lage, auch zeitdynamische Strukturen, wie sie in der Sprache vorkommen, zu erlernen. In dieser Arbeit wurde die Breite d des Eingabefensters zwischen der Eingabeschicht und der versteckten Schicht mit 3 und zwischen der versteckten Schicht und der Phonem Schicht mit 5 gewählt.

Mit dieser Netzarchitektur können rein akustische Netze und rein visuelle Netze trainiert werden, deren Ergebnisse kann man dann auf einer höheren

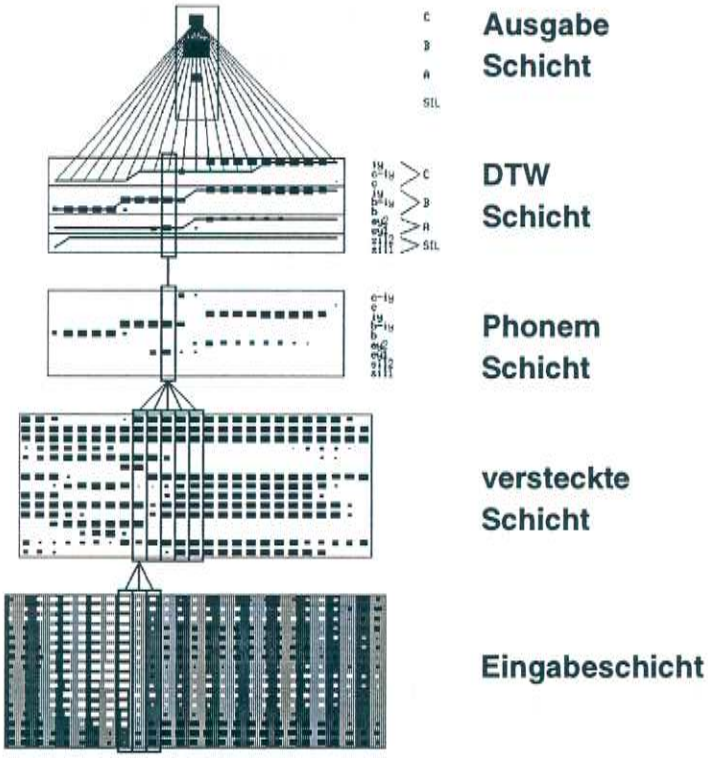

Abbildung 4.1: MS-TDNN, aus [19]

Ebene kombinieren. Diese Netzarchitektur ist in Abbildung 4.2 schematisch dargestellt.

In den ersten drei Schichten (input-hidden-phoneme/viseme) werden das akustische und visuelle Signal getrennt verarbeitet. In der dritten Schicht erhält man die Aktivierungen für 62 Phonem- bzw. 42 Visem-States für die akustischen und visuellen Daten. Die gewichtete Summe der Phoneme und der zugehörigen Viseme wird in die kombinierte Phonemschicht eingetragen. In der DTW Schicht wird durch den *One Stage Dynamic Time Warping Algorithmus* [35] der optimale Pfad durch die Phonem-States gesucht. Als Ergebnis erhält man die wahrscheinlichste Satzhypothese. In beiden Teilnetzen (akustisches TDNN und visuelles TDNN) werden je 15 versteckte Neuronen verwendet. Die Gewichte zur Kombination der Teilnetze werden dynamisch zur Laufzeit berechnet, um die Qualität der Schätzungen der Hypothesen

der beiden Modalitäten zu berücksichtigen. Eine Schätzung hat maximale Zuverlässigkeit, wenn das Netz für genau ein Phonem/Visem die Ausgabe 1 erzeugt und für alle anderen Phoneme/Viseme die Ausgabe 0 erzeugt. Eine minimale Zuverlässigkeit liegt vor, wenn für alle Phoneme/Viseme die gleiche Ausgabe erzeugt wird. Ein gebräuchliches Maß für die Zuverlässigkeit ist die sogenannte Entropie [38]. Sie basiert auf einem Satz von Wahrscheinlichkeiten und gibt Auskunft über den Informationsgehalt dieser Wahrscheinlichkeiten (Ein hoher Informationsgehalt entspricht dabei einer hohen Zuverlässigkeit). Dazu werden die Ausgabehypothesen der TDNNs zu Wahrscheinlichkeiten normalisiert. Für diese Normalisierung wird jede Hypothese durch Summe aller Hypothesen dividiert, die resultierenden Wahrscheinlichkeiten summieren sich somit zu 1. Die Entropie E berechnet sich nach folgender Formel:

$$E = -\sum_i \frac{hyp_i}{\sum_j hyp_j} \log \frac{hyp_i}{\sum_j hyp_j}$$

Sei E_A die Entropie des akustichen TDNNs, E_V die Entropie des visuellen TDNNs, hyp_i^A die akustische Hypothese und hyp_i^V die visuelle Hypothese, dann berechnet sich die bimodale Hypothese hyp_i^B nach folgender Formel:

$$hyp_i^B = w_a hyp_i^A + w_v hyp_i^V$$

Die Entropie-Gewichtungungen w_a und w_v der einzelnen Modalitäten berechnet man mit Hilfe der Entropie wie folgt:

$$w_a = b + \frac{E_A - E_V}{2K}$$

$$w_v = 1 - w_a$$

K entspricht dabei der maximalen Entropie im aktuellen Satz. Der Schwellwert b ist eine Voreinstellung für die Gewichte und bewirkt eine Bevorzugung einer bestimmten Modalität. Diese Voreinstellung wird durch die Entropie abgeschwächt bzw. verstärkt. Der Wert für b wird zur Zeit noch von Hand, abhängig von der Qualität der akustischen Daten, eingestellt. Im Rahmen zweier Studienarbeiten wird an einem Verfahren gearbeitet, das Signal-Rausch-Verhältnis für die akustischen Daten zu schätzen [42] und mit Hilfe dieser Schätzung den Parameter b automatisch zu bestimmen [23].

Es ist auch möglich, die Daten bereits in der Eingabeschicht zu kombinieren (vgl. Abbildung 4.4). Die Kombination in der Eingabeschicht hat den Nachteil, daß wesentlich mehr visuelle Koeffizienten (384 bei Graustufenbildern und 32 bei LDA) vorhanden sind, als akustische Koeffizienten (16 MelScale Koeffizienten). Daher ist zu erwarten, daß das Netz zu stark von den visuellen Parametern beeinflußt wird. Eine weitere Möglichkeit ist die Kombination in der versteckten Schicht (vgl. Abbilung 4.3) und somit vor der Kombination eine Abstrahierung von den Eingabekoeffizienten zu erlernen. Beide Methoden haben den Nachteil daß man keinen Einfluß auf die Art der Kombinierung hat, es ist nicht direkt möglich, Informationen über z.B. die akustische Qualität der Daten einzubringen, wie es bei der Kombination auf phonetischer Ebene durch den Schwellwert b möglich ist. An einer Methode, die SNR bei Kombination auf Eingabeschicht bzw. versteckter Schicht einzubringen, wird zur Zeit im Rahmen einer Studienarbeit [23] gearbeitet.

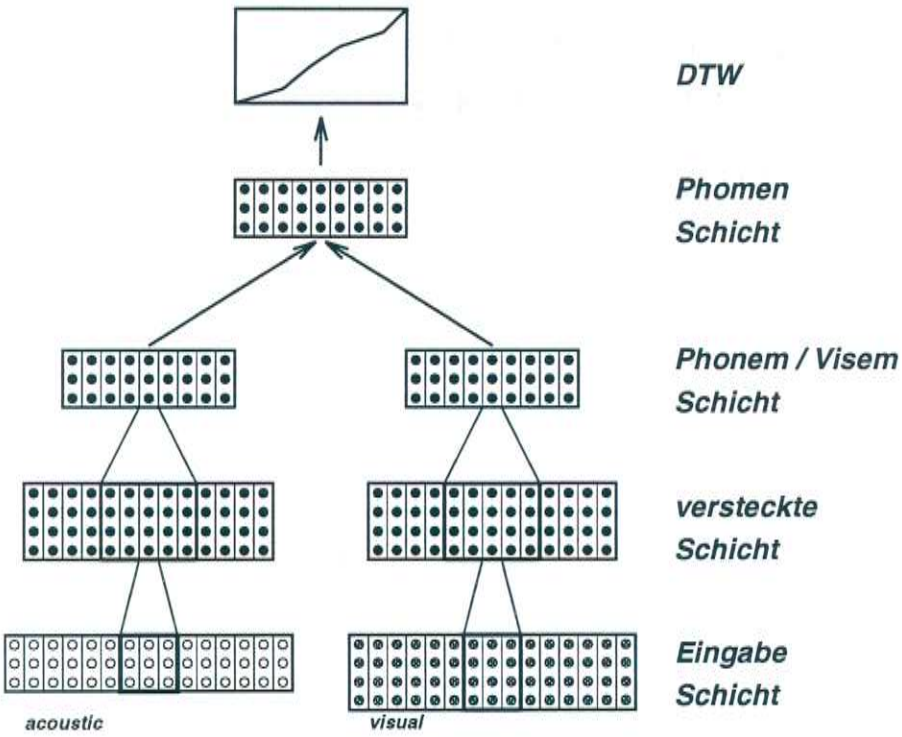

Abbildung 4.2: Kombination auf phonetischer Ebene

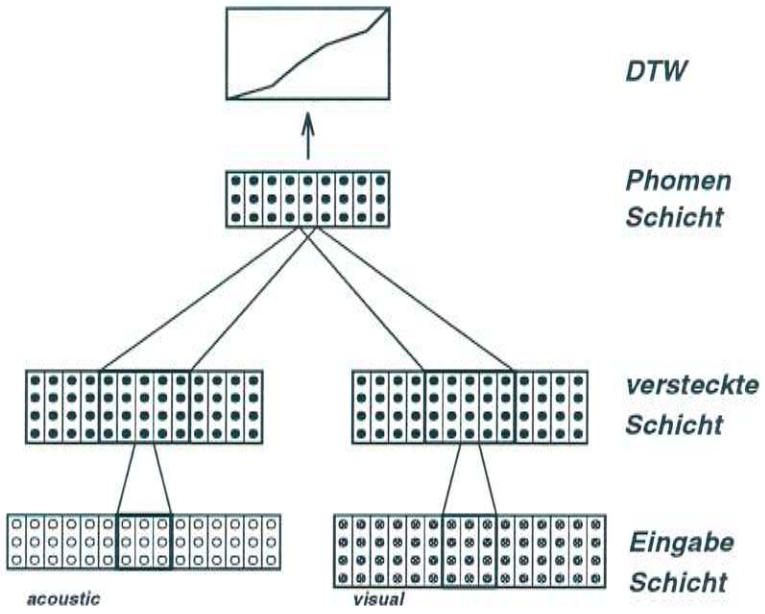

Abbildung 4.3: Kombination in der versteckten Schicht

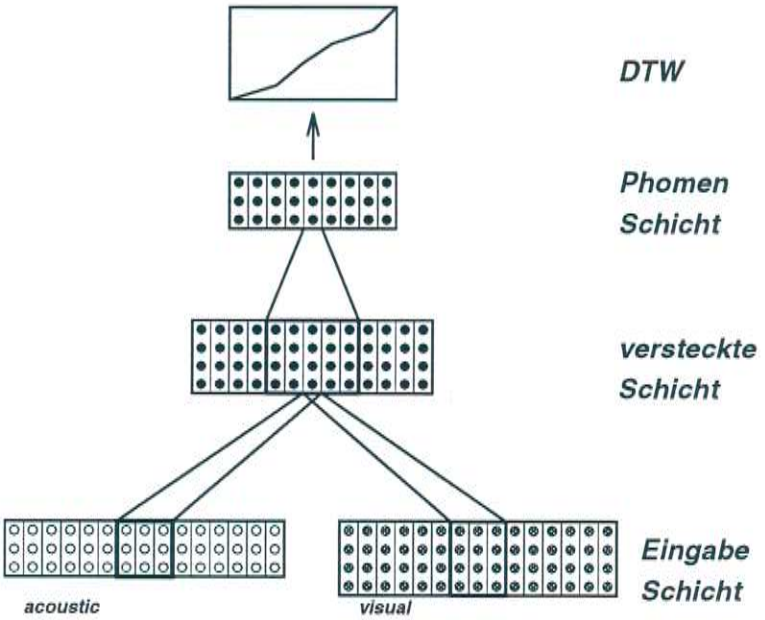

Abbildung 4.4: Kombination in der Eingabeschicht

4.2 Positionsinvariante Vorverarbeitung

4.2.1 Ziele der Vorverarbeitung

Eines der vorrangigen Ziele beim Entwurf des Gesamtsystems war es, daß der Sprecher durch keinerlei technische Hilfsmittel in seiner Bewegungsfreiheit eingeschränkt werden soll. Somit verbietet sich die Verwendung von Reflektionspunkten im Gesicht, speziellen Lippenstiften, am Kopf befestigten Kameras und ähnlichem. Dennoch soll es dem Sprecher ermöglicht werden, sich frei vor der Kamera zu bewegen.

Ein kritischer Punkt für den MS-TDNN Erkenner sind die sogenannten On-Line Bedingungen, wie:

- Positionsverschiebungen der Lippen
- Größenänderung der Lippen
- unterschiedliche Beleuchtungsbedingungen

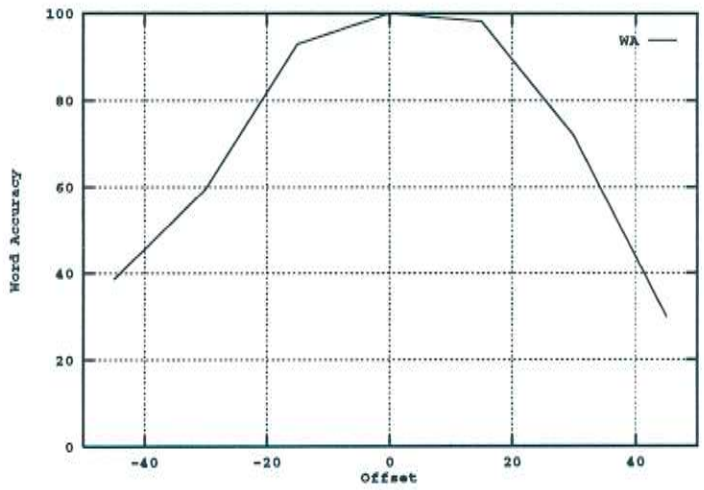

Abbildung 4.5: Verhalten des Erkenners bei unterschiedlicher Beleuchtung. X-Achse: Offset der auf jeden Pixel addiert wurde, Y-Achse: Word-Accuracy

Bereits in [31] wurde gezeigt, daß der MS-TDNN Ansatz auf OnLine Bedingungen besonders empfindlich reagiert. Dabei wurde festgestellt, daß auf einem ausgewählten Testset mit 100% Erkennungsrate je nach Änderung der Aufnahmebedingungen die Erkennungsrate signifikant sinkt. Abbildung 4.5 zeigt zum Beispiel das Verhalten bei unterschiedlichen (künstlich generierten) Beleuchtungsverhältnissen. Dabei wurde wie folgt vorgegangen: Es wurde ein Testset mit 100% Erkennungsrate (Word Accuracy) zusammengestellt. Auf die Grauwerte dieser Bilder wurde dann ein konstanter Offset addiert bzw. subtrahiert und mit diesen künstlich veränderten Bildern der Test erneut durchgeführt. Analog dazu wurden auch Tests mit unterschiedlicher Größe und Positionsänderungen durchgeführt.

Ziel der Vorverarbeitung ist es also, einen Sprecher im Raum zu finden und seine Lippenregion zu extrahieren. Weiterhin muß sich diese Vorverarbeitung robust gegen die OnLine Bedingungen verhalten.

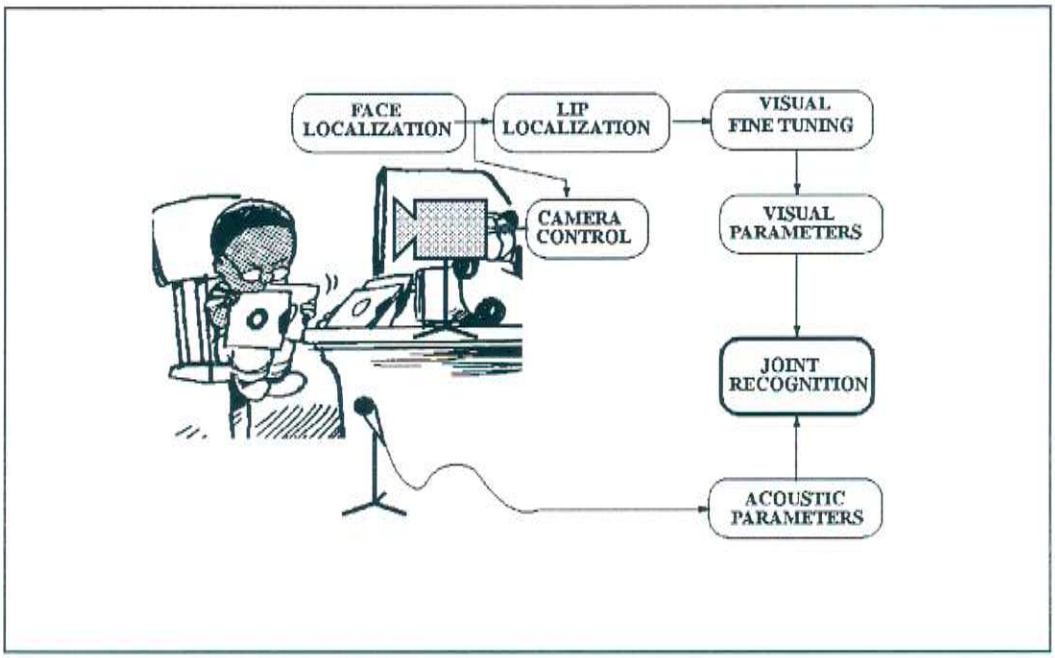

Abbildung 4.6: Überblick für einen Erkenner mit Aufnahmeverfahren 2

4.2.2 Teilmodule der Vorverarbeitung

Das Programm zur Vorverarbeitung besteht aus folgenden Teilkomponenten:

- Face Tracker, um den Sprecher zu finden und zu verfolgen
- Lippenfinder, um die Lippenregion zu extrahieren
- Programme zum Eliminieren der OnLine Probleme Größenänderung und Positionsänderung (Fine-Tuning)
- Beleuchtungsinvariante Vorverarbeitung der Lippenregion (Graustufenbilder) für den MS-TDNN Erkenner.

Ein Überblick über das Gesamtsystem zur Aufnahme der Daten wird in Abbildung 4.6 gegeben.

Der Face Tracker

Die Aufgabe des Face Trackers ist es, ein stabiles Gesichtsbild zu liefern. In dieser Arbeit wird der Ansatz von M. Hunke [21] verwendet. Der Face Tracker lokalisiert Gesichter in beliebigen Umgebungen. Während der Verfolgung des Gesichtes wird die Position der Kamera und die Stellung des Zooms automatisch so eingestellt, daß das Gesicht zentriert und in fester Größe im Kamerabild zu sehen ist. Zu jedem verarbeiteten Frame liefert der Face-Tracker die Positionen des Gesichts zurück.

Das Gesamtsystem ist in Abbildung 4.7 schematisch dargestellt.

Als Merkmale zum Finden und Verfolgen des Gesichtes werden drei Informatinosquellen genutzt:

1. Gesichtsfarbe
2. Bewegung
3. Gesichtsform

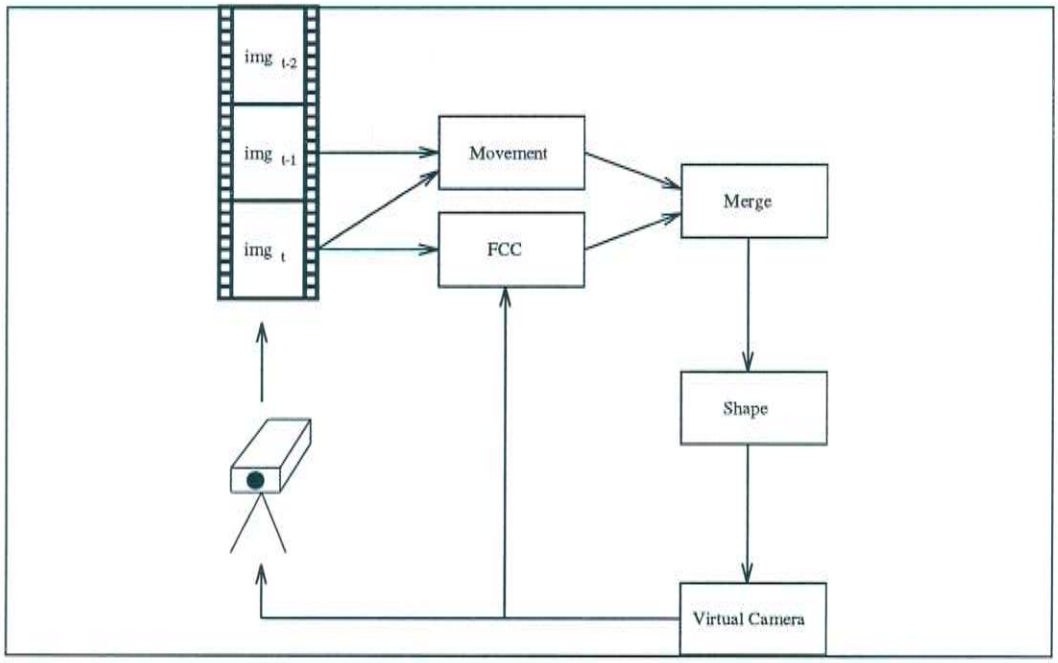

Abbildung 4.7: Systemüberblick Face Tracker, aus [21]

Die Gesichtsfarbe wird durch einen Gesichtsfarbenklassifikator (FCC, *Face Color Classifier*) bestimmt. Dazu wird in ein paar Beispielbildern (3-4 Bilder reichen aus) die Gesichtsregion markiert. Diese Farbinformation wird in eine Farbkarte eingetragen und vom FCC genutzt. Die Bewegung wird durch einfache Differenzenbildung von aufeinanderfolgenden Bildern bestimmt. Diese beiden Informationsquellen dienen dem ersten neuronalen Netz als Eingabe. Als Netzausgabe erhält man die Position des Gesichtes im Kamerabild. Durch ein zweites neuronales Netz wird dann die Größe des Gesichtes geschätzt. Dieses Netz erhält als Eingabe die Region des Kamerabildes, in der das Gesicht zentriert zu sehen ist. Da dieses Netz nur mit zentrierten Gesichtsbildern trainiert wurde und daraus die Größe des Gesichtes schätzt, enthält es implizit Informationen über die Gesichtsform.

Um die Laufzeit des Systems zu beschleunigen, wurde eine sogenannte virtuelle Kamera eingeführt. Das bedeutet, daß zur Netzeingabe nicht das gesamte Kamerabild dient, sondern nur ein kleiner Ausschnitt um die zuletzt gefundene Position des Gesichtes.

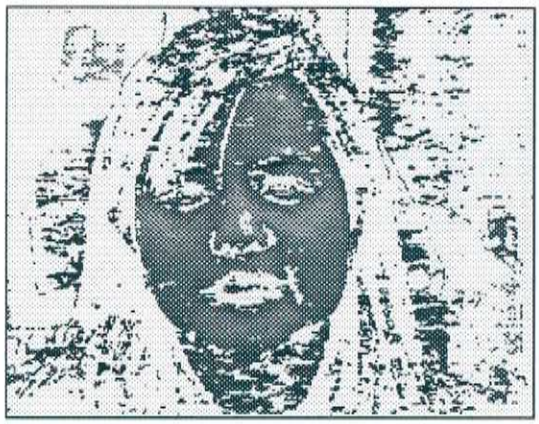

Abbildung 4.8: Beispiel für den Farbklassifikator, aus [21]

Abbildung 4.9: Momentaufnahme zur Laufzeit des Face Trackers

Abbildung 4.7 gibt einen Überblick über die Funktionsweise des Face Trackers. Abbildung 4.9 zeigt die Ausgabe des Face Trackers zur Laufzeit: Oben rechts ist das Kamerabild zu sehen, durch die Rechtecke sind die gefundene Gesichtsposition und die virtuelle Kamera, in der das Gesicht gesucht wird, gekennzeichnet. Oben links ist das aktuell gefundene Gesicht abgebildet. Unten sind, von links gesehen, die Ergebnisse des Gesichtsfarbenklassifikators,

des Bewegungsklassifikators und das kombinierte Ergebnis aus Gesichtsfarbe, Bewegung und Gesichtsform dargestellt.

Der Lippenfinder

Zur Lippenfindung wird der Ansatz von Dietrich Büsching [8] verwendet.

Dabei handelt es sich um ein zweistufiges Verfahren: Zuerst wird eine Grobsuche durchgeführt, mit der die ungefähre Position der Augen und Lippenregion bestimmt wird. Die Architektur des dabei verwendeten künstlichen neuronalen Netzes ist in Abbildung 4.10 dargestellt. Als Eingabe werden die horizontalen und vertikalen Gradienten der Gesichtsbilder verwendet.

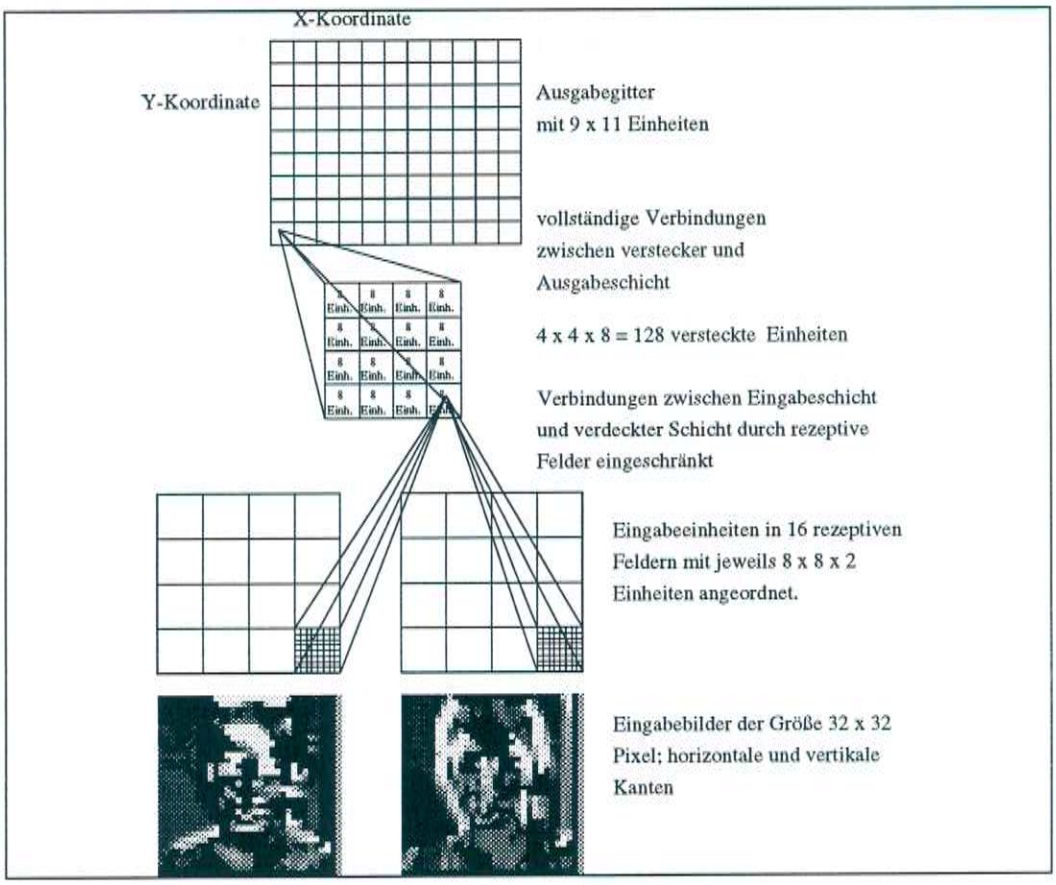

Abbildung 4.10: Netzarchitektur zur automatischen Lippenfindung, aus [8]

Abbildung 4.11: Beispiel zur Lippenfindung

In Abbildung 4.11 ist das Zentrum der so gefundene Position durch ein Rechteck markiert. Anhand der Augenpositionen und der ungefähren Lippenregion wird der Bildbereich bestimmt, in dem die Lippen liegen. Dieser Bereich dient dann in der zweiten Stufe, deren Netzarchitektur ähnlich zu der der ersten Stufe ist, als Eingabe. Als Ergebnis erhält man dann den rechten und linken Mundwinkel zurück, in Abbildung 4.11 durch die zwei Kreuze markiert.

Fine-Tuning

Probleme, die bei der Verwendung des Face Trackers und des Lippenfinders auftreten:

1. Die Größe und Position des durch den Face Tracker gefundenen Gesichtes variieren. Somit kann die für eine Sequenz ermittelte Gesichts- bzw. Lippenposition sehr stark variieren und nicht auf einfache Weise geglättet werden, um Ausreißer in der Erkennung zu eliminieren (vgl. Abbildung 4.13).

2. Der Face Tracker liefert bei der Aufnahme nur für ca. jedes 8. Bild eine neue Gesichtsposition. Für alle dazwischenliegenden Frames muß die Gesichtsposition als konstant angenommen werden. Dies kann zu Problemen führen, wenn sich der Sprecher zu schnell vor der Kamera bewegt oder die Motor-/Zoomsteuerung der Kamera aktiv wird.

3. Die durch den Lippenfinder gelieferten Positionen geben zwar sehr genau die Koordinaten der Mundwinkel an, so daß die komplette Lippe

im Lippenfenster enthalten ist. Die Koordinaten können jedoch trotzdem von der 'idealen' Position, wie immer man diese auch definieren mag, um ein paar Pixel abweichen. Dies ist insofern ein Problem, da der MS-TDNN Erkenner sehr empfindlich auf Verschiebungen innerhalb des Lippenfensters reagiert.

4. Der Abstand zwischen den beiden Mundwinkeln ist abhängig von dem gesprochenem Phonem. So ist zum Beispiel bei einem */o/* Laut der Abstand wesentlich geringer als bei */silence/* (vgl. Abbildung 4.12). Wird lediglich der vom Lippenfinder ermittelte Abstand zwischen den Mundwinkeln zur Skalierung verwendet, so kommt es innerhalb einer Sequenz zu Verzerrungen bzw. Größenänderungen der Lippen in einzelnen Frames.

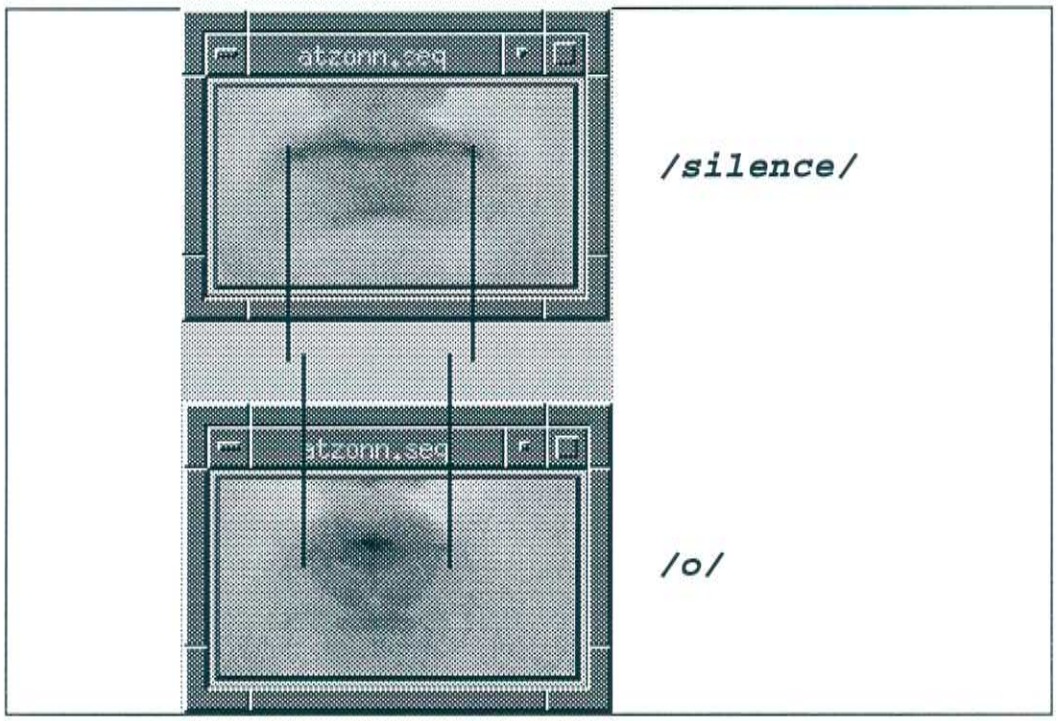

Abbildung 4.12: Abhängigkeit der Lippenbreite vom gesprochenen Phonem

Abbildung 4.13: Folgebilder in der Gesichtssequenz

Vorgehensweise beim Fine-Tuning:

- **Einsatz des Face Trackers:**
  Zur Bestimmung der Gesichtsgröße wird von allen Frames, in denen der Face Tracker eine Position und Größe geliefert hat, die maximale Größe bestimmt. In allen anderen Frames wird dann nur noch die durch den Face Tracker ermittelte Gesichtsposition (deren Mittelpunkt) verwendet. Das Gesicht selbst wird in der oben ermittelten Maximalgröße ausgeschnitten. Dies hat zum einen den Vorteil, daß alle Gesichtsbilder einer Buchstabiersequenz in derselben Größe vorliegen. Zum anderen, was zur robusten Lippenfindung notwendig ist, wird dadurch auch das Problem behoben, daß der Face Tracker nur ca. alle 8 Frames eine neue Gesichtsposition liefert. In diesen 8 Frames muß die zuletzt gefundene Gesichtsposition als konstant angenommen werden. Durch die Verwendung des etwas größeren Gesichtsausschnitts werden somit auch kleinere Bewegungen innerhalb dieser 8 Frames (= 2-3 sec) berücksichtigt, so daß in allen Bildern das Gesicht vollständig enthalten ist.

- **Einsatz des Lippenfinders:**
  Da die Lippenfindung ein sehr zeitaufwendiger Prozeß ist (z.B. müssen dafür die Bilder auf eine Auflösung von 256x256 Pixel vergrößert werden), wird der Lippenfinder nur in den Bildern verwendet, in denen der Face Tracker eine neue Gesichtsposition liefert. In diesen Frames muß der Lippenfinder aufgerufen werden, da die relative Position des Gesichts innerhalb des Bildes sich ändert. Eine Glättung mit den Lippenpositionen der Vorgängerbilder ist aus oben genannten Gründen

nicht möglich. Für alle Folgebilder, in denen keine neue Gesichtsposition geliefert wird, wird diese Position als konstant vorausgesetzt.

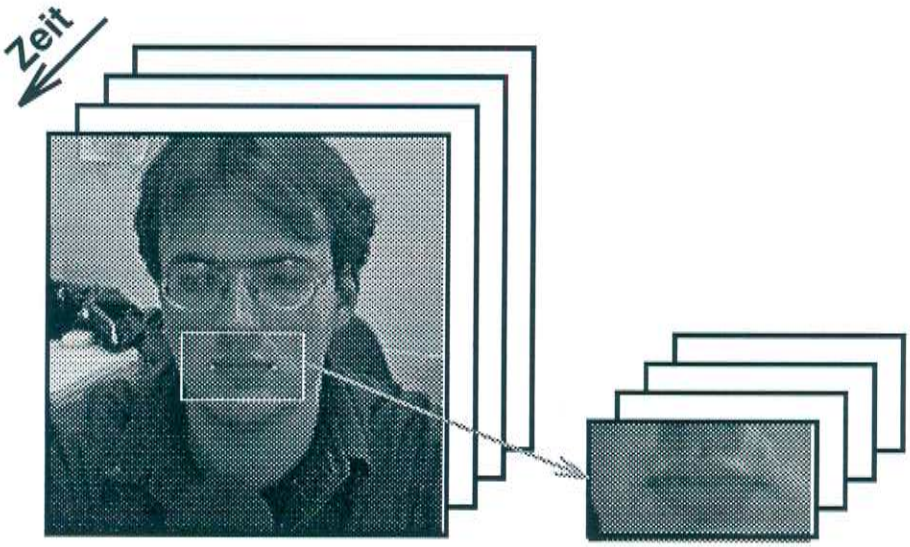

Abbildung 4.14: Automatische Lippenfindung

- **Verwendung von Frame Korrelation:**
  Wird zur Lippenextraktion nur der Face Tracker und der Lippenfinder wie oben beschrieben eingesetzt, ergibt sich folgendes Problem: In den so extrahierten Lippensequenzen bewegen sich die Lippen relativ zum vorhergehenden Bild um einige Pixel.

  Dieses Problem entsteht durch folgende Randbedingungen: Zum einen treten beim Lippenfinder kleinere Ungenauigkeiten in der Positionierung auf und zum anderen wird (aus Zeitgründen) nicht für jedes Bild die Position der Mundwinkel neu bestimmt.

  Dies führt, wie bereits oben erwähnt, beim MS-TDNN Ansatz zu Problemen, da dieser empfindlich auf Positionsänderungen reagiert. Wird

der Erkenner mit verrauschten ('verwackelten') Bildsequenzen trainiert, wird dieser zwar robuster gegenüber verwackelten Daten, die Gesamtperformance ist aber deutlich schlechter als bei einem Erkenner, der auf sauberen Daten trainiert wurde. Daraus ergibt sich die Notwendigkeit, die Bildsequenzen durch einen weiteren Verarbeitungsschritt zu 'entwackeln'.

Dabei wird wie folgt vorgegangen: für einen Teilausschnitt eines Bildes wird die Korrelation mit verschiedenen Teilausschnitten des Vorgängerbildes berechnet.

Sei $g(i,j)$ ein digitisiertes Bild und $t(i,j)$ eine Schablone mit dem Definitionsbereich D. Der Absolutbetrag der Differenzen dieser beiden Funktionen wird als das Maß der Übereinstimmung zwischen dem Bild und der an Position (n,m) zentrierten Schablone definiert:

$$E(m,n) = \sqrt{\sum_i \sum_j [g(i,j) - t(i-m, j-n)]^2} \quad (i-m)(j-n) \in D$$

Dieses Differenzen-Maß soll bezüglich der Position (n,m) minimiert werden.

Es gilt:

$$E(m,n)^2 = \sum_i \sum_j [g(i,j) - t(i-m, j-n)]^2$$

$$= \sum_i \sum_j [g^2(i,j) - 2g(i,j)t(i-m, j-n) + t^2(i-m, j-n)]$$

Da der Term $\sum_i \sum_j t^2(i-m, j-n)$ unabhängig vom Grauwertverlauf an der Position (m,n) im Bild ist, ergibt sich daraus bei weiterer Vereinfachung der Formel die Kreuzkorrelation als Ähnlichkeitsmaß:

$$R_{gt} = \sum_i \sum_j g(i,j)t(i-m, j-n) \quad (i-m)(j-n) \in D$$

Die Korrelation liefert dabei ein Maß für die Übereinstimmung der beiden Teilausschnitte. Wählt man den Teilausschnitt mit der größten Übereinstimmung aus, läßt sich daraus berechnen, inwieweit der Ausschnitt für die Lippenregion verschoben werden muß [40] [34].

Bei der Lippenkorrelation wurde der 144x80 Pixel große Lippenbereich mit einem 164x100 Pixel großen Bereich des Ursprungsbildes korreliert, d.h. die Region der ausgeschnittenen Lippen wurde in jede Richtung um 10 Pixel vergrößert.

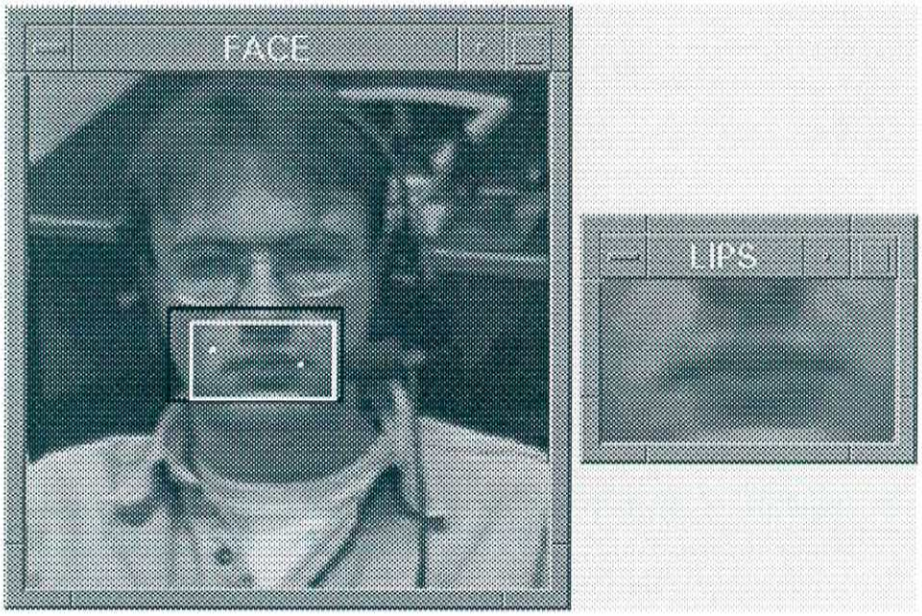

Abbildung 4.15: Beispiel zur Verbesserung durch Framekorrelation

- **Einführung einer Master-Lippe:**
  Bei Verwendung der 3 obigen Teilschritte zur Lippenextraktion erhält man Lippensequenzen, die jede für sich sauber positioniert und auch von konstanter Größe sind. Die verwendete Position der Mundwinkel und die daraus resultierende Größe der Lippen hängen dabei nur von dem ersten Frame einer jeden Sequenz ab. Somit ergeben sich zwischen unterschiedlichen Bildsequenzen größere Variationen bezüglich Größe und Position.

  Um dies zu vermeiden, wird eine sogenannte Master-Lippe eingeführt, die als Referenzbild für jede Sequenz dienen soll. Um diese Master-Lippe zu verwenden, ist es jetzt notwendig, die Korrelation zwischen

den Bildern nicht nur bezüglich der Verschiebung, sondern zusätzlich auch noch für verschiedene Größen zu berechnen.

Diese Korrelation wird zwischen dem ersten Bildframe und der Master-Lippe berechnet. Für alle weiteren Frames ist dies nicht notwendig, da die Größe der Bilder innerhalb einer Sequenz als konstant angesehen werden kann und die Korrelation für die Verschiebung nach obigem, schnelleren Verfahren berechnet werden kann.

Abbildung 4.15 zeigt ein Beispiel zur Verwendung der Framekorrelation. In schwarz ist die ursprünglich gefundene Lippenregion eingezeichnet, in Weiß die durch die Framekorrelation korrigierte Position.

- **Ausschneiden der Lippen-Region:**
  Würde man die jetzt gefundenen Mundwinkel direkt zum Ausschneiden der Lippenregion verwenden, würde dies zu Verzerrungen innerhalb der Bildsequenz führen, da je nach gesprochenem Phonem die Mundwinkel unterschiedlich weit voneinander entfernt sind. Daher wird der Abstand der Mundwinkel des ersten Frames als Musterabstand genommen. Dieser erste Frame zeigt immer einen geschlossenen Mund (*silence*) und stellt somit auch den Maximalabstand dar. Die durch obige Routinen gefunden Mundwinkel werden auf der x-Achse solange zum Bildrand verschoben, bis der durch den ersten Frame bestimmte Musterabstand gegeben ist.

  Nachdem die Mundwinkel im Gesichtsbild bestimmt wurden, muß noch die Lippenregion aus dem Gesamtbild ausgeschnitten werden und auf die von den Vorverarbeitungsroutinen verwendete Standardgröße von 144x80 Pixel skaliert werden. Für diese Standardgröße sind die Positionen, an denen der Mundwinkel liegen soll, fest vorgegeben. Diese Position muß so gewählt werden, daß die Lippen bei jeder Lippenstellung vollständig zu sehen sind. Aus diesen vorgegebenen Positionen und den gefundenen Mundwinkeln läßt sich dann die Höhe und Breite der Lippenregion und der Skalierungsfaktor berechnen.

Zusammenfassend kann man sagen, daß sich mit dieser Vorverarbeitung die Lippenregion sehr genau bestimmen läßt und die Variationen der Sequenzen bezüglich Positions- und Größenänderungen auf ein Minimum reduziert werden.

Der Nachteil dieses Verfahrens liegt darin, daß diese Methode sehr rechenintensiv ist. Der Hauptanteil dieser Rechenzeit wird zum einen für das Vergrößern und Verkleinern der Bilder benötigt, da einige Teilmodule die Bilder in bestimmten Größen voraussetzen. Zum anderen ist die Berechnung der Korrelationen, besonders wenn auch die Größen-Korrelation mit eingeht, sehr zeitintensiv.

Aus diesem Grund sind momentan einige Randbedingungen an das System gestellt, die einen vertretbaren Kompromiß zwischen Laufzeit des Gesamtsystems und allgemeiner Verwendbarkeit darstellen. So wird beispielsweise angenommen, daß sich der Sprecher während der Aufnahme nicht zu stark vor der Kamera bewegt, kleinere Bewegungen werden durch die Frame-Korrelation ausgeglichen. Kann dies nicht gewährleistet werden, so muß für jeden Bildframe die Lippenposition neu bestimmt werden, was in einer höheren Laufzeit des Systems resultiert.

Beleuchtungsinvariante Vorverarbeitung der Lippenregion

Ziel dieser Vorverarbeitung der Lippenregion ist es, unterschiedliche Beleuchtungsverhältnisse auszugleichen. Dabei sind prinzipiell zwei Arten von Beleuchtungsproblemen zu unterscheiden:

1. Das Beleuchtungsniveau des gesamten Bildes ist unterschiedlich.
2. Zusätzlich kann noch starke Seitenbeleuchtung auftreten.

- **bisheriges Verfahren**
  Bei der bisherigen Methode[1] zur Normalisierung der Graustufenbilder wurden die oberen und unteren 5% der Grauwerte bestimmt und zu 255 bzw. 0 gesetzt, die dazwischenliegenden Daten wurden dann linear angepaßt.

[1] diese Methode wurde bereits in früheren Arbeiten verwendet [5] [31].

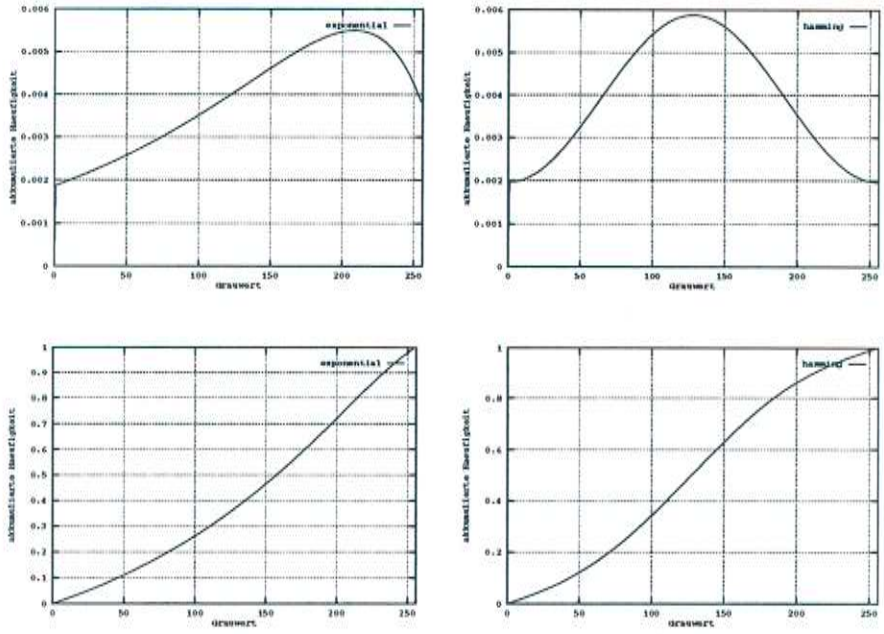

Abbildung 4.16: Zielfunktionen $p_d(g)$ und $P_d(g)$, links eine Exponentialfunktion, rechts eine hamming-ähnliche Funktion

- **Kontrast / Grauwert-Modifikation:**
  Bei der Grauwert-Modifikation wird das Graustufenbild f durch die Transformation T in ein neues Graustufenbild g überführt:

$$g = T[f]$$

  Die Transformation T ändert dabei die Beleuchtungsintensität. Das Histogramm $p(f)$ eines Bildes repräsentiert für jeden Grauwert die Anzahl der Pixels mit diesem Grauwert im Bild. Ziel ist, die Transformation T zu berechnen, die aus der Verteilungsfunktion $p(f)$ das neue Bild g so berechnet, daß $p(g)$ möglichst nahe bei der gewünschten Verteilungsfunktion $p_d(g)$ liegt. Auf diese Weise werden dann die Beleuchtungsunterschiede in den verschiedenen Bildern ausgeglichen.

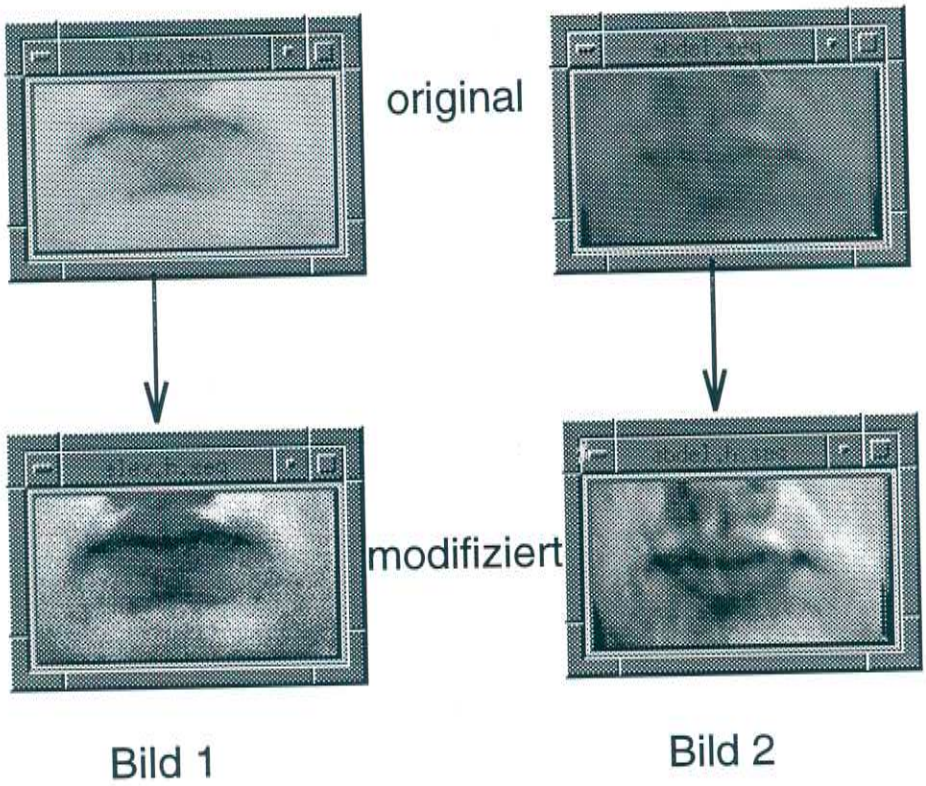

Abbildung 4.17: Beispiel zur Grauwert-Modifikation (Hamming-Verteilung)

Ein möglicher Lösungsansatz für dieses Problem ist es, die Wahrscheinlichkeitsverteilungen $P(f)$ und $P_d(g)$, die sich durch Integration aus $p(f)$ und $p(g)$ berechnen lassen, zu betrachten und die Transformationsfunktion T so zu wählen, daß $P(f)$ möglichst nahe bei $P_d(g)$ liegt, wenn $g = T[f]$ gilt. Die Tatsache, daß $P(f)$ eine monoton steigende Funktion ist, garantiert, daß ein Pixel mit höherer Intensität als ein anderer Pixel auch im Ausgabebild eine höhere Intensität hat. Im diskreten Fall ergibt sich die Berechung somit wie folgt:

$$P(f) = \sum_k p(k) = P(f-1) + p(f)$$

$$P_d(g) = \sum_k p_d(k) = P_d(g-1) + p(g)$$

sei P_d^{-1} die Umkehrfunktion zu P_d, dann berechnet sich die Transformation T wie folgt:

$$T[f] = P_d^{-1}(P(f))$$

Bildlich gesprochen wird also zu jedem Grauwert im Ursprungsbild die akkumulierte Häufigkeit berechnet und allen Pixeln mit diesem Grauwert wird der Grauwert zugewiesen, der in der Zielverteilung dieselbe akkumulierte Häufigkeit hat. Da die Verteilungsfunktion streng monoton steigend ist, kann dieser eindeutig bestimmt werden.

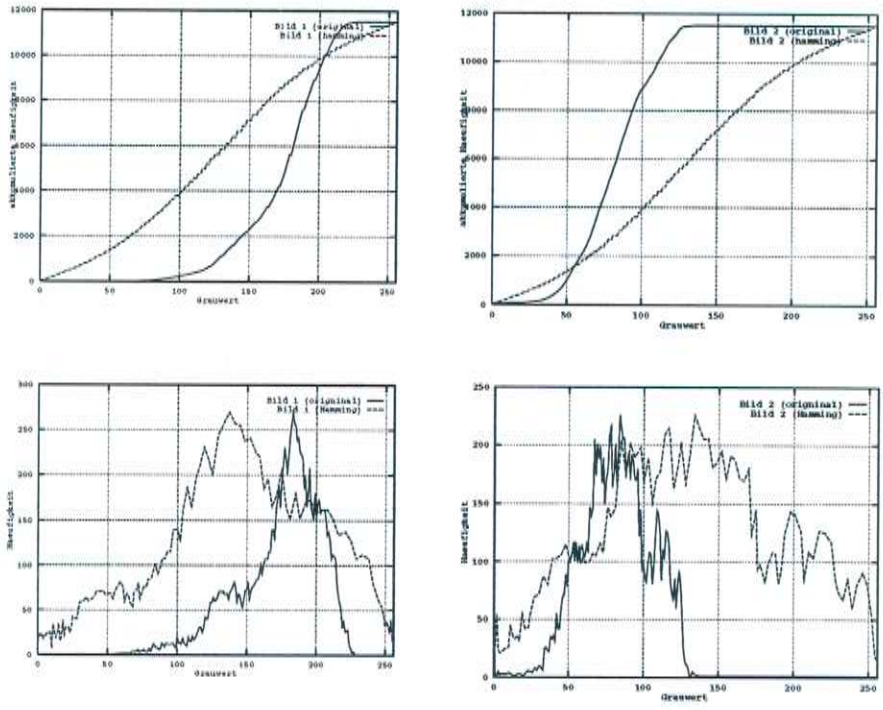

Abbildung 4.18: Verteilung für Bild 1 und Bild 2 (mit Hamming-Verteilung)

Mögliche Ziel-Wahrscheinlichkeitsverteilungen sind in Abbildung 4.16 gegeben. In Abbildung 4.17 sind 2 Bilder und das Ergebnis der Grauwert-Modifikation dargestellt, die zugehörigen Histogramme und Wahrscheinlichkeitsverteilungen sind in Abbildung 4.18 zu sehen.

- **Adaptive Grauwert-Modifikation:**
  Wird die Grauwertmodifikation wie im obigen Verfahren beschrieben berechnet, haben alle Bilder zwar die gleiche Beleuchtungsintensität, das Problem der Seitenbeleuchtung wird dadurch allerdings nicht behoben.

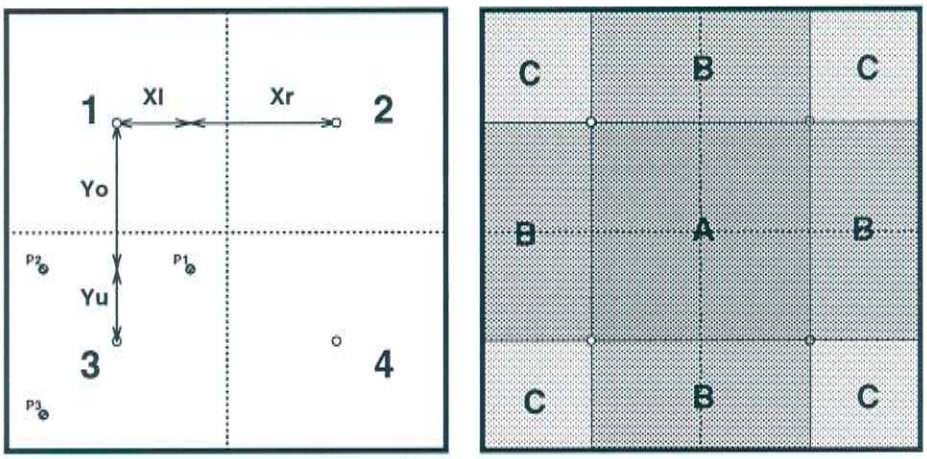

Abbildung 4.19: Aufteilung des Bildes zur adaptiven Grauwert-Modifikation

Um dieses Problem zu beheben, wird wie folgt vorgegangen: Das Bild wird in 4 Quadranten (siehe linkes Bild in Abbildung 4.19) aufgeteilt. Für diese 4 Quadranten werden dann die Transformationen T_1, T_2, T_3 und T_4 getrennt berechnet. Die endgültige Transformation für jeden einzelnen Pixel wird als Linearkombination

$$T(f(p)) = \sum_{i=1}^{4} w_i T_i(f(p))$$

berechnet ($f(p)$ bezeichnet den Grauwert des Pixels p).

Bestimmung der Skalarfaktoren w_i: Jeder der 4 Quadranten (Q_i) wird wiederum in 4 Quadranten (q_{ij}) unterteilt. Aus der Nachbarschaftsbeziehung zu anderen Quadranten berechnet sich dann der Skalarfaktor

(vergleiche rechtes Bild in Abbildung 4.19). Für die Nachbarschaftsbeziehung sind 3 Fälle zu unterscheiden (Regionen A, B und C). Die Berechung der Skalarfaktoren soll exemplarisch für die 3 Punkte P_1, P_2 und P_3 in Quadranten Q_3 gezeigt werden:

$$\begin{aligned} T(P_1) &= \frac{y_u}{y_o + y_u}\left(\frac{x_r}{x_l + x_r}T_1(P_1) + \frac{x_l}{x_l + x_r}T_2(P_1)\right) \\ &+ \frac{y_o}{y_o + y_u}\left(\frac{x_r}{x_l + x_r}T_3(P_1) + \frac{x_l}{x_l + x_r}T_4(P_1)\right) \\ T(P_2) &= \frac{y_u}{y_o + y_u}T_1(P_2) + \frac{y_o}{y_o + y_u}T_3(P_2) \\ T(P_3) &= T_3(P_3) \end{aligned}$$

Die Berechnung für Punkte in den anderen Quadranten erfolgt analog dazu.

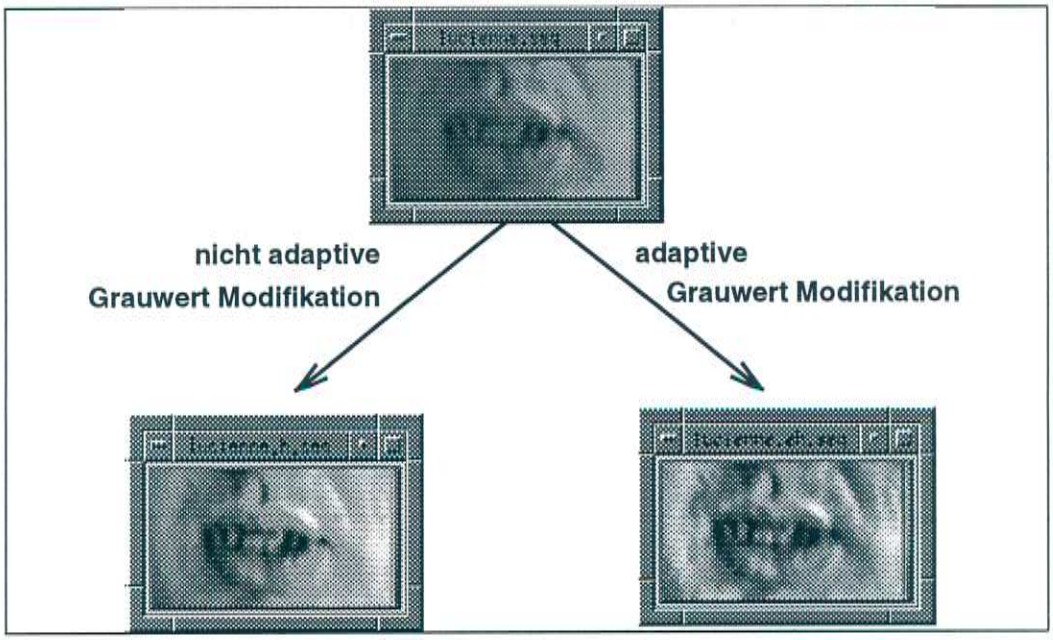

Abbildung 4.20: Vergleich von adaptiver und nicht adaptiver Grauwert-Modifikation

In dieser Arbeit wurden folgende Methoden verwendet:

- nicht modifizierte Bilder
- Hammingfunktion als Zielfunktion
- adaptive Hammingfunktion als Zielfunktion
- Exponentialfunktion als Zielfunktion
- adaptive Exponentialfunktion als Zielfunktion

4.3 Symmetriebilder

Durch Ausnutzung der Symmetrie von Lippenbildern läßt sich auf einfache Weise die Dimension auf die Hälfte reduzieren. Voraussetzung ist, daß die Lippen bezüglich der x-Achse zentriert im Bild vorliegen. Dies ist durch die Verwendung des Lippenfinders garantiert. Auch bei den Daten, die mit Aufnahmeverfahren 1 ohne Lippenfinder gesammelt wurden, kann von zentrierten Lippenbildern ausgegangen werden.

Bei diesem Verfahren wird jeweils der Grauwert eines Pixels auf der linken Hälfte des Bildes mit dem korrespondierenden Pixel[2] auf der rechten Bildhälfte gemittelt. Da jeweils der rechte und linke Pixel denselben Grauwert zugewiesen bekommen, genügt es jetzt, nur noch eine Bildhälfte als Eingabemenge zu verwenden.

Abbildung 4.21 zeigt ein Beispiel aus der Datenbank. Oben ist das Originalbild zu sehen, unten links das durch Symmetriebildung entstandene Bild, unten rechts das reduzierte Bild, das, nach entsprechender Vorverarbeitung (Grauwertmodifikation, ...), im Training verwendet werden kann.

[2]dieser Punkt läßt sich durch Spiegelung an der x-Achse bestimmen: $x_r = breite - x_l$

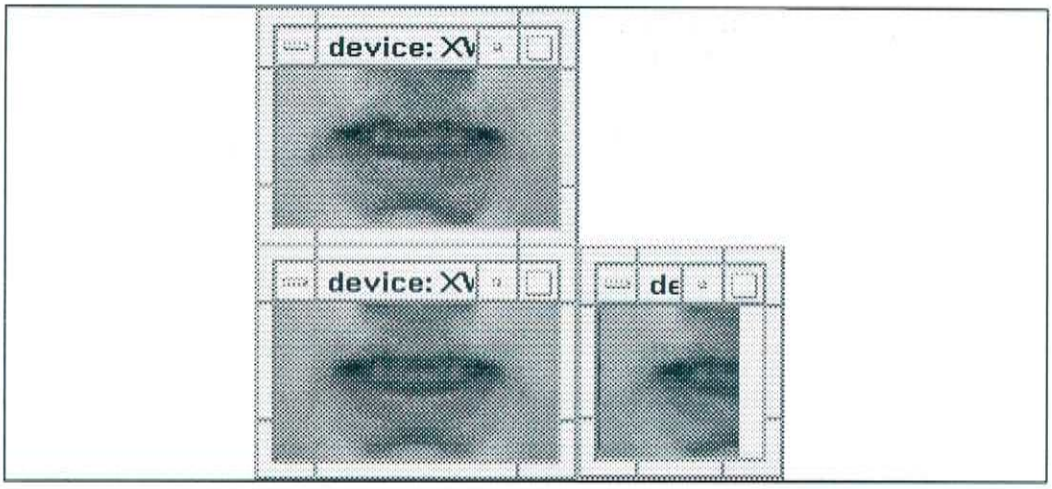

Abbildung 4.21: Dimensionsreduktion durch Symmetriebildung

4.4 Unterschiede beim CMU-Lippenleser

Bei der Portierung des Lippenlese-Systems auf die Hardware an der CMU ergaben sich folgende Unterschiede:

- **Hardware:**
  Für das Aufnehmen der Daten stand eine DEC Alpha mit Framegrabber und Gradientbox zur Verfügung. Für das Aufnehmen der Daten ergaben sich keine Unterschiede in der Framerate.

  Für den Face Tracker stand eine HP Workstation mit Framegrabber zur Verfügung. Durch die spezielle Eigenschaft des HP Framegrabbers, das 'grabben' der Bilder auf Teilausschnitte des gesamten Kamerabildes zu beschränken, ergab sich hier eine deutlich höhere Verarbeitungsgeschwindigkeit.

- **Software:**
  Als Face Tracker wurde eine von Jie Yang und Ricky Houghton weiterentwickelte Version des Face Trackers verwendet. Unterschiede zur bisher in Karlsruhe verwendeten Version liegen in einer höheren Verarbeitungsgeschwindigkeit und einer wesentlich stabileren Gesichtsfindung. Dies wirkt sich vor allem in Situationen aus, in denen sich der

Sprecher (fast) nicht vor der Kamera bewegt. Die Positionsangaben des Gesichtes in Folgeframes weichen dann nicht mehr so stark voneinander ab. Erreicht wurde die höhere Geschwindigkeit zum einen durch weitere Optimierungen am bestehenden Code und zum anderen durch algorithmische Änderungen: Bei dem Verfahren von M. Hunke wurde eine Farbkarte mit Informationen der Gesichtsfarbe gehalten. Das anpassen der Farbkarte an das aktuell gefundene Gesicht ist dabei ein relativ zeitaufwendiger Prozeß. In der neuen Version des Face Trackers wird die Farbverteilung über eine Gaußverteilung angenähert und zur Laufzeit nur noch der Mittlwert und die Varianz dieser Verteilung angepaßt.

Auswirkungen auf des Gesamtsystem:

- **Glätten der Gesichtsposition:**
  Ändert sich die Gesichtsposition von Folgebildern um max. 10 Pixel (bei einer Bildgrösse von 320x240), wird die alte Position beibehalten. In dem für das Lippenlesen typischen Szenario, einem Sprecher, der vor dem Computer sitzt, hat das zur Folge, daß praktisch in der gesamten Sequenz die Gesichtsposition konstant ist.

- **Korrektur der Lippenposition:**
  Aufgrund der konstanten Gesichtsposition in Folgebildern können jetzt die durch den Lippenfinder gefunden Mundwinkelpositionen geglättet bzw. Fehler bei der Lippenfindung behoben werden. Als Merkmal für fehlerhaft gefundene Lippenpositionen können die Y-Koordinaten der Mundwinkel verwendet werden. Weicht die Y-Koordinate der beiden Mundwinkel zu stark voneinander ab, deutet dies auf einen Fehler hin. In diesem Fall wird dann die Position des Vorgängerframes verwendet.

4.5 Das Gesamtsystem

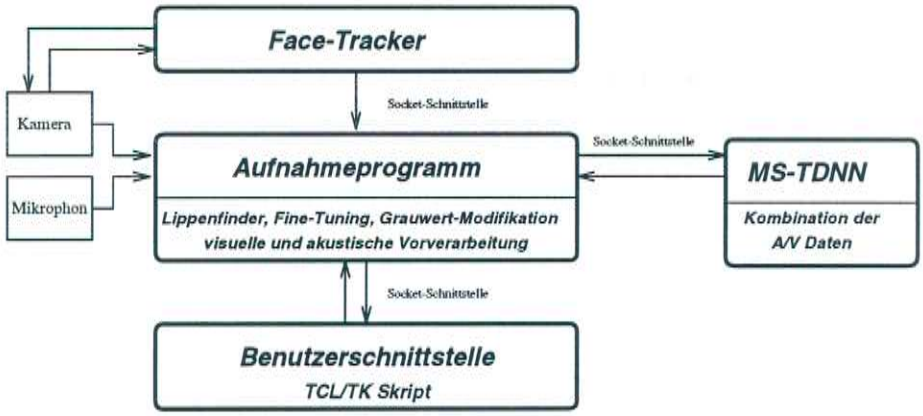

Abbildung 4.22: Gesamtsystem für den OnLine Erkenner

Das Gesamtsystem besteht aus folgenden Programmen:

- Eine in TCL/TK implementierte Benutzerschnittstelle
- Das Aufnahmeprogramm, bestehend aus folgenden Teilmoduln:
  - Lippenfinder
  - Fine-Tuning (Framekorrelation)
  - beleuchtungsinvariante Vorverarbeitung (Grauwert-Modifikation)
  - visuelle Vorverarbeitung (z.B. LDA Berechnung)
  - akustische Vorverarbeitung
- Der MS-TDNN Erkenner

Aufgrund der hohen Anforderungen an die Laufzeit der Programme müssen diese auf getrennten Rechnern laufen, um eine möglichst hohe Framerate zu erzielen. Die Programme werden dazu über eine Socket-Schnittstelle miteinander gekoppelt. Da zum einen der Austausch der Bilddaten über eine Socketschnittstelle zu langsam ist und zum anderen die Framerate des Face Trackers zu niedrig für das Lippenlesen ist, muß die Videokamera sowohl an den Rechner mit dem Aufnahmeprogramm wie auch an den Rechner mit dem

Face Tracker angeschlossen werden. Der Face Tracker und das Aufnahmeprogramm tauschen dann lediglich die Gesichtspositionen miteinander aus. Das Gesamtsystem ist in Abbildung 4.22 dargestellt, in Abbildung 4.24 ist ein Screendump der Demo-Version der Benutzerschnittstelle zu sehen.

Abbildung 4.23 zeigt die Nutzung des Face Trackers und des Lippenfinders zur Lippenextraktion: Der Lippenfinder wird nur in den Frames verwendet, in denen der Face Tracker eine neue Position geliefert hat. Für alle weiteren Frames wird diese Position als konstant angenommen. Um Ungenauigkeiten bzw. Fehler bei der Lippenfindung auszugleichen, wird jeder Frame, in dem eine neue Lippenposition bestimmt wurde, mit seinem Vorgängerframe korreliert und die Position so verschoben, daß die Lippen eine möglichst hohe Korrelation haben. Für den ersten Bildframe wird zusätzlich eine Größenkorrelation mit einer Master-Lippe durchgeführt.

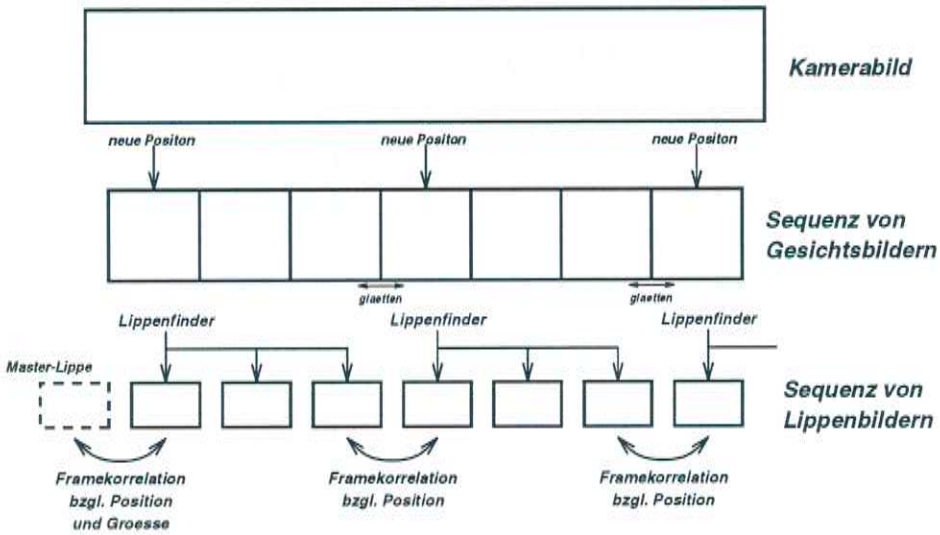

Abbildung 4.23: Nutzung von Face Tracker und Lippenfinder zur Lippenextraktion

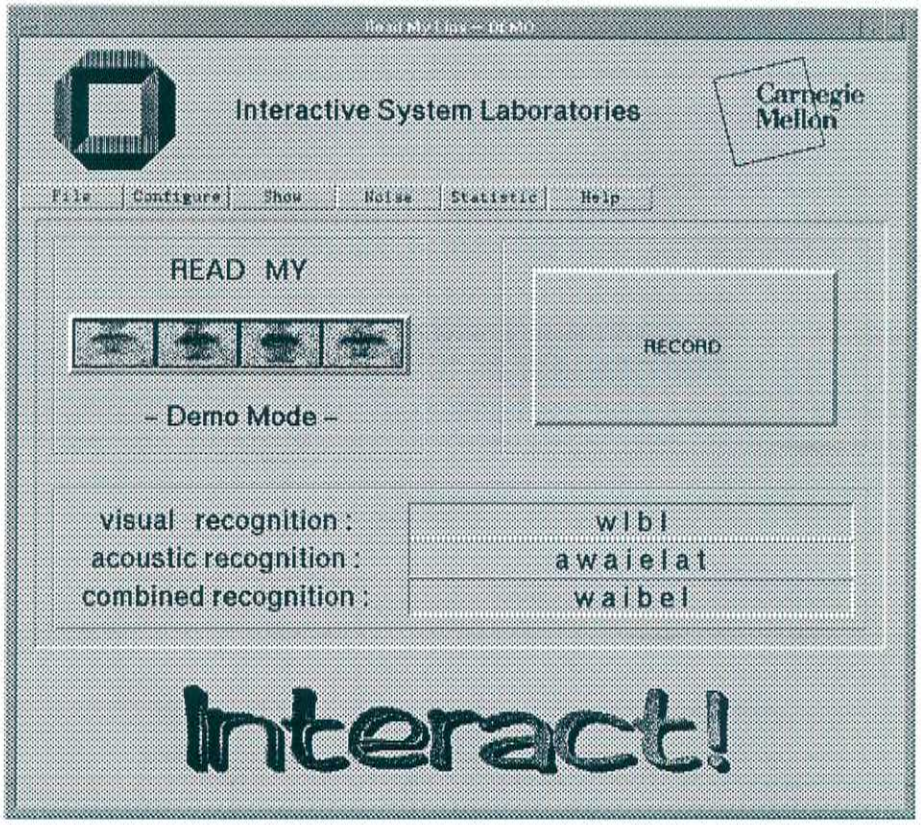

Abbildung 4.24: Online Demo

4.6 Ergebnisse

Die Datensätze bestehen je aus 200 Sequenzen eines Sprechers, davon wurden 140 zum Training, 30 für Cross-Validation und 30 für das Testset verwendet. Die Erkennungsraten werden grundsätzlich als Word Accuracy angegeben. Die Word Accuracy berechnet sich wie folgt:

$$WA = 100\%(1 - \frac{\#\text{SubstitutionError}+\#\text{InsertionError}+\#\text{DeletionError}}{\#\text{Buchstaben}})$$

4.6.1 Aufnahmeverfahren 1

Zunächst wurden die Netze auf den Datensätzen mum1/2 bzw. mum9/10 mit verschiedenen Vorverarbeitungen trainiert. Diese Daten wurden mit dem Aufnahmeverfahren 1 aufgenommen.

Vorverarbeitungsmethoden

Tabelle 4.1 zeigt die erzielte Word-Accuracy auf dem Testset mit Aufnahmeverfahren 1 bei unterschiedlicher Vorverarbeitung. Bei den Vorverarbeitungsmethoden schneiden die Graustufenbilder und die LDA am besten ab. Dabei ist zu beachten, daß die LDA eine Reduktion der Eingabedimension um den Faktor 12 erzielt.

Kombinationsmethoden

Die Tabellen 4.2, 4.3 und 4.4 zeigen die Ergebnisse bei Kombination auf Input-, versteckter und phonetischer Ebene. Erwartungsgemäß schneidet dabei die Kombination auf phonetischer Ebene am besten ab. Beim Training mit Kombination auf Eingabeschicht bzw. versteckter Schicht hat man keinen Einfluß auf die Art der Kombinierung. Ein weiteres Problem beim Kombinieren auf niederer Ebene ergibt sich bei verrauschten Daten: Es werden dieselben Gewichte bei verrauschten und bei sauberen Testdaten verwendet. Bei Kombination auf phonetischer Schicht kann man durch einstellen des Schwellwertes b (vgl. Kapitel 4.1) Einfluß auf die Gewichtung der beiden Modalitäten nehmen.
Die Kombination mit den akustischen Daten wurde auf phonetischer Ebene mit unterschiedlichen Signal-Rauschverhältnissen getestet. Dazu wurden die

akustischen Daten der 30 Sequenzen künstlich mit weißem Rauschen (16 dB und 8 dB SNR) verschlechtert. Die Kombination mit den visuellen Daten erfolgte in der phonetischen Ebene. Die Ergebnisse sind in Tabelle 4.4 dargestellt. Selbst bei sauberen Daten wurde eine Verbesserung von bis zu 68% erzielt.

Die Ergebnisse der phonetischen Kombination sind in Abbildung 4.25 graphisch dargestellt. Dazu wurden auf der X-Achse die mit den akustischen Daten erzielte Word Accuracy, auf der Y-Achse die auf diesen Daten kombiniert erreichte Word Accuracy eingetragen.

Symmetriebilder

In Tabelle 4.9 ist die visuelle Word Accuracy bei Verwendung von Symmetriebildern aufgelistet. Dabei sind VVID die Originalbilder, SYM die daraus durch Symmetriebildung entstandenen Bilder.

An der Tabelle wird deutlich daß die Reduktion der Eingabedaten um 50% durch Verwendung von Symmetriebildern vergleichbare Ergebnisse liefert.

Aufnahmeverfahren 1 visuelle Erkennungsrate			
Daten:	Parameter:	Word Accuracy	
		mum1/2	mum9/10
Graustufen	384	55%	44%
Principal Components	32	52%	45%
Linear Discriminant Analysis	32	56%	59%
FFT Ring	29	50%	38%

Tabelle 4.1: Ergebnisse der visuellen Netze mit Aufnahmeverfahren 1

Aufnahmeverfahren 1 Kombination in der Eingabeschicht			
Daten:	Parameter	Word Accuracy	
		mum1/2	mum9/10
Graustufen	384 + 16	93%	86%
Principal Components	32 + 16	74%	91%
Linear Discriminant Analysis	32 + 16	93%	92%
nur Akustik	16	96%	93%

Tabelle 4.2: Ergebnisse bei Aufnahmeverfahren 1 und Kombination in der Eingabeschicht

Aufnahmeverfahren 1 Kombination in der versteckten Schicht			
Daten	Parameter	Word Accuracy	
		mum1/2	mum9/10
Graustufen	384 + 16	97%	89%
Principal Components	32 + 16	88%	89%
Linear Discriminat Analysis	32 + 16	94%	95%
nur Akustik	16	96%	93%

Tabelle 4.3: Ergebnisse der visuellen Netze mit Aufnahmeverfahren 1 bei Kombination in der versteckten Schicht

Aufnahmeverfahren 1 Kombination in der phonetischen Schicht			
	Word Accuracy (Fehlerreduktion)		
mum1/2	saubere Daten	16 dB SNR	8 dB SNR
Graustufen	99.5% (68%)	73.4% (38%)	66.5% (47%)
PC	99.5% (68%)	71.3% (33%)	60.1% (37%)
LDA	98.9% (31%)	68.1% (25%)	56.3% (31%)
FFT Ring	98.4% (0%)	67.6% (24%)	53.2% (26%)
nur akustische Daten	98.4%	56.9%	36.2%

Tabelle 4.4: Ergebnisse mit Aufnahmeverfahren 1 und Kombination auf phonetischer Ebene und bei unterschiedlichem weißen Rauschen auf den akustischen Daten

Aufnahmeverfahren 1 Symmetriebilder		
Vorverarbeitung	VVID	SYM
mum9/10	44%	47%

Tabelle 4.5: Ergebnisse der visuellen Netze mit Aufnahmeverfahren 1 und Symmetriebildern

Aufnahmeverfahren 1 Kombination in der phonetischen Schicht			
	Word Accuracy (Fehlerreduktion)		
mum9/10	saubere Daten	16 dB SNR	8 dB SNR
Symmetriebilder	98.2% (50%)	76.2% (34%)	56.5% (27%)
nur akustische Daten	96.4%	63.7%	40.5%
nur visuelle Daten	47.0%	47.0%	47.0%

Tabelle 4.6: Ergebnisse mit Aufnahmeverfahren 1 mit Symmetriebildern und Kombination auf phonetischer Ebene bei unterschiedlichem weißen Rauschen auf den akustischen Daten

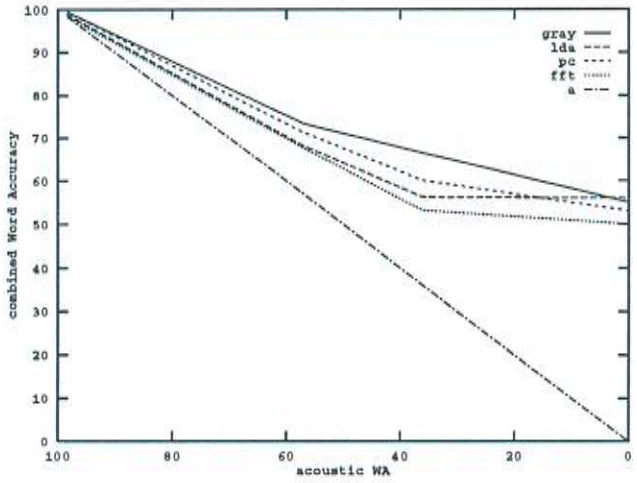

Abbildung 4.25: Ergebnisse, Aufnahmeverfahren 1

4.6.2 Aufnahmeverfahren 2

Weiterhin wurde das Netz auf Daten trainiert, die mit dem Aufnahmeverfahren 2 aufgenommen wurden. Dabei wurden als Vorverarbeitung verschiedene Verfahren der Grauwertmodifikation sowie eine darauf aufbauende LDA verwendet. Die Ergebnisse sind in Tabelle 4.7 aufgelistet. Auffallend ist dabei, daß die Ergebnisse für die visuelle Erkennung mit den Graustufenbildern deutlich schlechter als bei den Daten mum9/10 (von Aufnahmeverfahren 1) sind. Dies liegt darin begründet, daß bei der Aufnahme der Daten mum9/10 der Backlight-Modus der Kamera eingeschalten war, wodurch die Bilder wesentlich besser ausgeleuchtet waren. Dieser Modus kann bei Daten, die mit dem Aufnahmeverfahren 2 aufgenommen werden, nicht verwendet werden, da dabei bei Farb-Aufnahmen die Farben so stark verfälscht werden, daß der Gesichtsfarben Klassifizierer (FCC) des Face Trackers nicht mehr optimal arbeiten kann. Dieser Nachteil wirkt sich allerdings nur bei Verwendung von Graustufenbildern als Netzeingabe aus. Verwendet man z.B. die LDA-Koeffizienten zur Netzeingabe, werden auch hier Erkennungsraten von über 50% erreicht.

Grauwert Modifikation

Vergleicht man in Tabelle 4.7 die Ergebnisse der verschiedenen Grauwertmodifikationen miteinander, fällt folgendes auf: Bei veränderten Grauwertmodifikationen ergeben sich auf den Daten in der Erkennungsleistung keine signifikanten Unterschiede. Dies liegt daran, daß alle Daten am selben Tag unter denselben Bedingungen aufgenommen wurden. Verändert man aber künstlich die Beleuchtung auf diesen Daten oder testet auf Daten, die unter anderen Beleuchtungsbedingungen aufgenommen wurden, zeigt sich hier, daß bei dem Netz mit nicht grauwertmodifizierten Bildern die Erkennungsleistung sehr stark sinkt (vgl. [31]), bei Verwendung von Grauwertmodifikation bleibt diese praktisch konstant (sofern sich nicht auch andere Bedingungen wie Größe oder Position ändern).

Kombination von Akustik und visuellen Daten

Auch mit diesen Daten wurde die Kombination auf phonetischer Ebene mit unterschiedlich verrauschten Daten getestet. Dabei wurden die akustischen Daten anstelle des weißen Rauschens auch mit Musik- bzw. Motorengeräuschen verrauscht. Die damit erzielten Ergebnisse sind in Tabelle 4.8 dargestellt. Auch hier werden Verbesserungen der Erkennungsraten von bis zu 55% erzielt. Abbildung 4.26 stellt diese Ergebnisse nochmals graphisch dar. Die Schwankungen in den kombinierten Erkennungskurven kommen dadurch zustande, daß mit unterschiedlichen Arten von Rauschen getestet wurde.

Symmetriebilder

Tabelle 4.9 gibt einen Vergleich bei Verwendung von Symmetriebildern zur Dimensionsreduktion gegenüber der Verwendung der Originalbilder. Auch hier wird eine Verbesserung durch Symmetriebilder erzielt.

Aufnahmeverfahren 2 visuelle Erkennungsrate			
	Word Accuracy		
	Graustufen Bilder		LDA
Daten:	mum17/18	mum21-23	mum21-23
nicht modifiziert	25%	31%	50%
hamming modifiziert	26%	32%	53%
exponential modifiziert	26%	31%	55%
adaptive hamming	25%	27%	54%
adaptiv exponential	32%	30%	53%

Tabelle 4.7: Ergebnisse bei verschiedenen Grauwertmodifikationen

Aufnahmeverfahren 2 visuell und akustisch kombinierte Erkennungsrate			
	Word Accuracy (Fehlerreduktion)		
mum21-23	Graustufen	LDA	Akustik
saubere Daten	97.6% (20%)	97.6% (20%)	97.0%
25 dB - Motor	95.8% (0%)	97.0% (28%)	95.8%
16 dB - Motor	52.1% (15%)	62.4% (36%)	40.6%
16 dB - Musik	90.3% (23%)	93.9% (51%)	87.3%
10 dB - Musik	72.1% (33%)	80.0% (52%)	58.2%
16 dB - weißes Rauschen	75.2% (55%)	76.4% (28%)	67.3%
8 dB - weißes Rauschen	48.5% (20%)	63.6% (37%)	42.4%

Tabelle 4.8: Ergebnisse mit Aufnahmeverfahren 2 und Kombination auf phonetischer Ebene, Test mit unterschiedlichem künstlichen Rauschen auf den akustischen Daten

Aufnahmeverfahren 2 Symmetriebilder		
Vorverarbeitung	VVID	SYM
mum25-28	31%	36%

Tabelle 4.9: Ergebnisse der visuellen Netze mit Aufnahmeverfahren 2 mit Symmetriebildern

Aufnahmeverfahren 2 visuell und akustisch kombinierte Erkennungsrate			
	Word Accuracay (Fehlerreduktion)		
mum25-28	saubere Daten	16 dB	8 dB
Symmetriebilder	93.2% (28%)	84.6% (22%)	59.8% (30%)
nur akustische Daten	90.6%	80.3%	42.3%
nur visuelle Daten	36.0%	36.0%	36.0%

Tabelle 4.10: Ergebnisse mit Aufnahmeverfahren 2 mit Symmetriebildern und Kombination auf phonetischer Ebene und bei unterschiedlichem weißen Rauschen auf den akustischen Daten

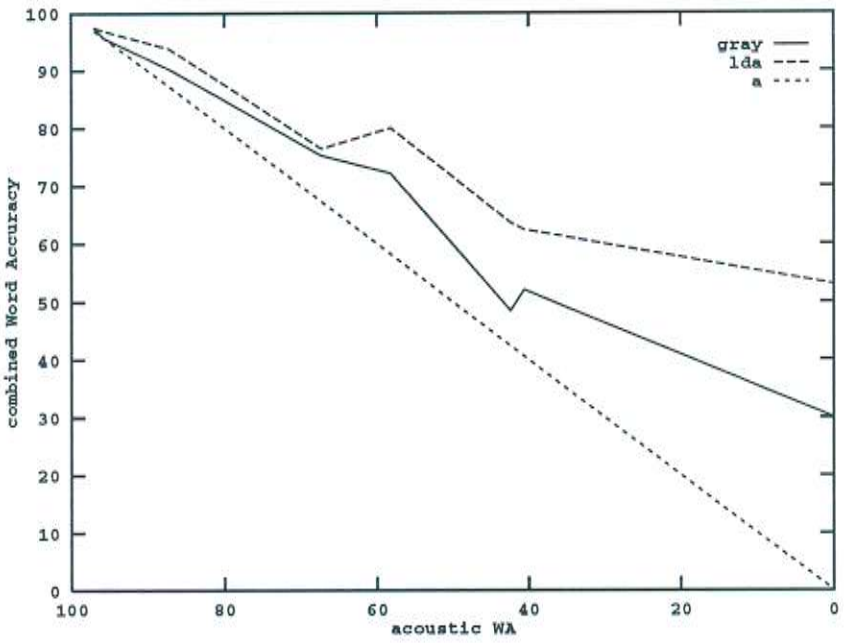

Abbildung 4.26: Ergebnisse, Aufnahmeverfahren 2

Kapitel 5

Der MS-TDNN^{3d} Erkenner

Bei dem in Kapitel 4 beschriebenen MS-TDNN basierten Ansatz wird das Problem der Positionsinvarianz durch eine zeitaufwendige Vorverarbeitung gelöst. Dieser Weg bringt zwei Nachteile mit sich: Zum einen müssen die Daten vor dem Trainieren der Netze durch die in Kapitel 4 beschriebene Vorverarbeitung aufbereitet werden. Dies dauert bei ca. 200 Sequenzen mehrere Stunden. Zum anderen bleibt dieser Zeitfaktor auch in der OnLine Erkennung erhalten. Der Vorteil dieser Architektur liegt allerdings darin, daß die Netze relativ schnell trainiert werden können.

Die Idee ist jetzt, die Positionsinvarianz nicht durch eine Vorverarbeitung zu erreichen, sondern dieses Problem direkt in der Netzarchitektur zu berücksichtigen. Dieser neue Netztyp soll sich also robust gegenüber Verschiebungen der Lippen innerhalb des Bildes verhalten.

Das *3-dimensional Multi-State Time-Delay-Neural-Network* (MS-TDNN^{3d}) soll in den folgenden Abschnitten erläutert werden.

Bei dem MS-TDNN^{3d} werden die Vorteile eines MS-TDNN (vergleiche Abbildung 4.1) bezüglich der Zeitinvarianz und von zwei-dimensionalen Netzen, wie sie zum Beispiel zur Handschriftenerkennung eingesetzt werden [25], bezüglich der Positionsinvarianz miteinander kombiniert.

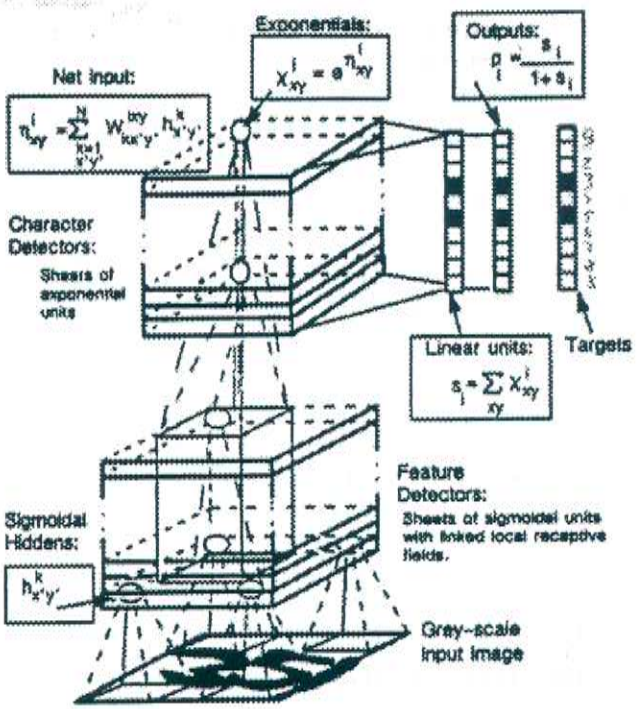

Abbildung 5.1: Schematische Darstellung des ISR-Netzes, aus [25]

5.1 Verwandte Architekturen

5.1.1 ISR-Architektur

Die hier vorgestellte ISR[1]-Netzarchitektur wurde von Keeler, Rumelhart und Leow [25] zur Handschriftenerkennung für Ziffern eingesetzt. Abbildung 5.1 zeigt den Aufbau eines solchen Netzes.

Das Eingabebild kann mehrere, beliebig positionierte Ziffern enthalten. Das Graustufenbild wird dem Netzwerk als 2-dimensionale Matrix präsentiert. Die Units $h^k_{x'y'}$ der ersten Schicht (*Feature Detector*) sind mit rezeptiven Feldern (Teilausschnitten) der Eingabeschicht verbunden. Dabei sind die

[1]ISR = Integrated Segmentation and Recognition

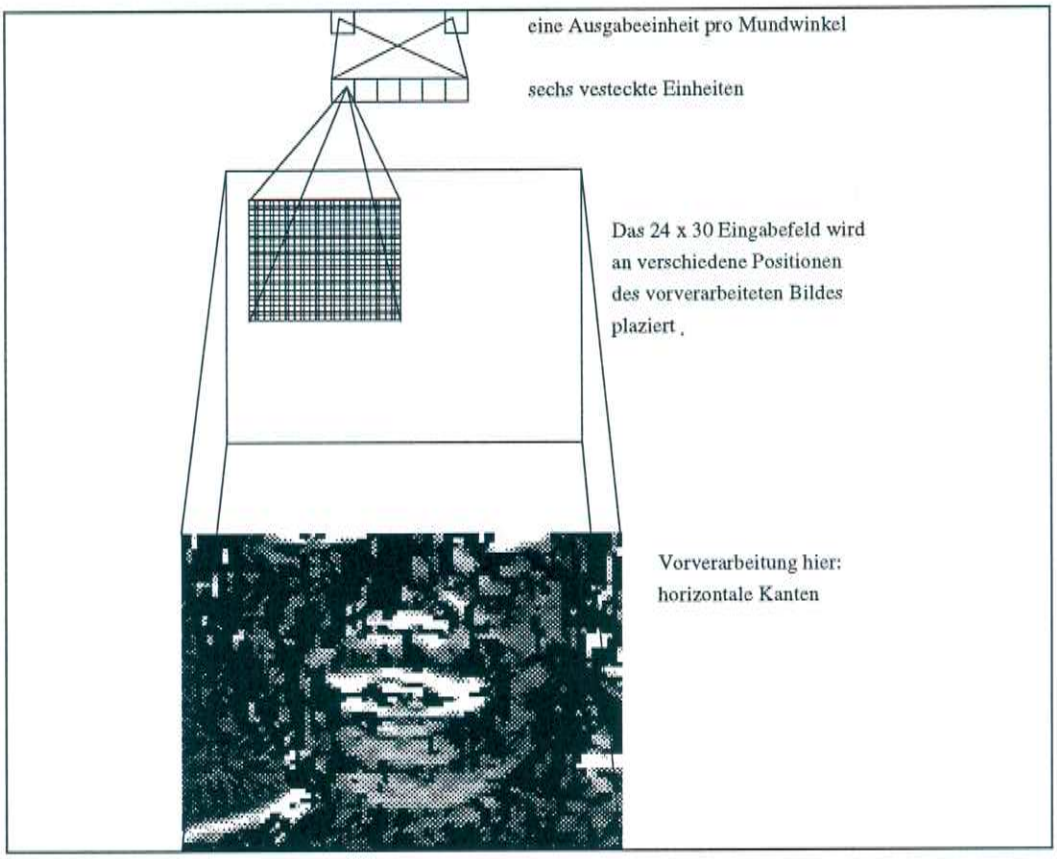

Abbildung 5.2: Schematische Darstellung der template-basierten Architektur, aus [8]

Gewichte für ein Merkmal k und unterschiedliche $x'y'$ miteinander gekoppelt (*linked weights*). In der nächsten Schicht (*Character Detector*) sind die Units mit lokalen rezeptiven Feldern der versteckten Schicht verbunden. Auf diese Weise erhält man positionsabhängige Information über die Ziffern. Diese Information wird dann ebenenweise (zeichenweise) zusammengefaßt. Dadurch erhält man globale Information über die Zeichen, die in der Eingabeschicht vorkommen.

5.1.2 Template-basierte Architektur

Dietrich Büsching hat in seiner Diplomarbeit [8] zur Lippenfindung für die Feinsuche u.a. einen template-basierten Ansatz untersucht. Der Aufbau dieses Netzes ist in Abbildung 5.2 dargestellt. Dabei wird eine Eingabemaske über das Eingabebild verschoben. Als Ausgabe werden dabei lediglich 2 Units verwendet, die anzeigen, ob die Mundwinkel in dem Teilausschnitt zu sehen waren. Für jede mögliche Position werden auch hierbei die Gewichte gekoppelt, d.h. überall die gleichen Gewichte verwendet.

5.2 Die MS-TDNN^{3d} Architektur

Für die Architektur des MS-TDNN^{3d} wurde das MS-TDNN um die Idee der rezeptiven Felder zur positionsinvarianten Bildverarbeitung erweitert. Bisher wurden für das MS-TDNN die Eingabebilder als eindimensionaler Vektor repräsentiert, Bildinformation, wie z.B. Nachbarschaftsbeziehungen, ging dadurch verloren, und dies führte bei der Architektur zu Problemen bzgl. Positionsinvarianz.

Positionsinvarianz

Beim MS-TDNN^{3d} bleibt Bildinformation erhalten, da die Merkmale zunächst positionsabhängig, also für bestimmte Bildausschnitte, detektiert werden und erst auf einer höheren Ebene zu globalen Merkmalen zusammengefaßt werden. Da für jeden Teilausschnitt dieselben Gewichte verwendet werden, können im Bild vorkommende Merkmale unabhängig von der Position detektiert werden.

Zeitinvarianz

Da beim Lippenlesen nicht nur Einzelbilder, sondern Bildsequenzen bearbeitet werden, ergibt sich die Notwendigkeit, auch den Faktor Zeit in der Architektur zu berücksichtigen. Dafür werden analog zum TDNN die *time-delays* verwendet: Die Bildausschnitte werden nicht nur zum Zeitpunkt t, sondern auch noch verzögert zu den Zeitpunkten $t+1, ..., t+(d-1)$ angelegt, um die zeitliche Struktur besser zu erlernen.

Freie Parameter

Bei dieser Netzarchitektur ergeben sich zusätzlich zu den freien Parametern des MS-TDNNs noch folgende Parameter:

- Größe der einzelnen Bildmasken in jeder Schicht
- Überlappungsgrad der einzelnen Masken in jeder Schicht.

Namenskonventionen

I_{xy}	Grauwert an der Position (x, y) im Eingabebild
$H^i_{x'y'}$	i-te versteckte Einheit an der Position (x', y')
$L^j_{x'',y''}$	j-te Phonem an der Position (x'', y'') in der lokalen Phonemschicht
P^j	j-te Phonem in der (globalen) Phonemschicht
$s(x)$	sigmoid-Funktion $\frac{1}{1+e^{-x}}$
y_i	Aktivierung des Ausgabeneurons i
t	Zielaktivierung eines Ausgabeneurons (target)

5.2.1 Strategie 1: Erlernen von Teilmustern

Die Masken der Eingabeschicht werden bezüglich der Bildgröße relativ klein gewählt, evtl. sogar ohne Überlappung. Die Idee ist dabei analog zu der Netzarchitektur bei Schrifterkennung, zuerst einmal Teilmuster der Buchstaben zu detektieren und dann auf höhere Ebene zusammenzufassen.

Forward-Pass

Berechnung:

$$H^i_{x'y'} = s\left(\sum_t \sum_{xy \in mask} w^{it}_{xy} I_{xy}\right)$$

$$L^j_{x'',y''} = \sum_{t'} \sum_{x'y' i \in mask} w^{jt}_{x'y'} H^i_{x'y'}$$

$$P^j = s\left(\frac{1}{n} \sum_{x''y''} L^j_{x'',y''}\right)$$

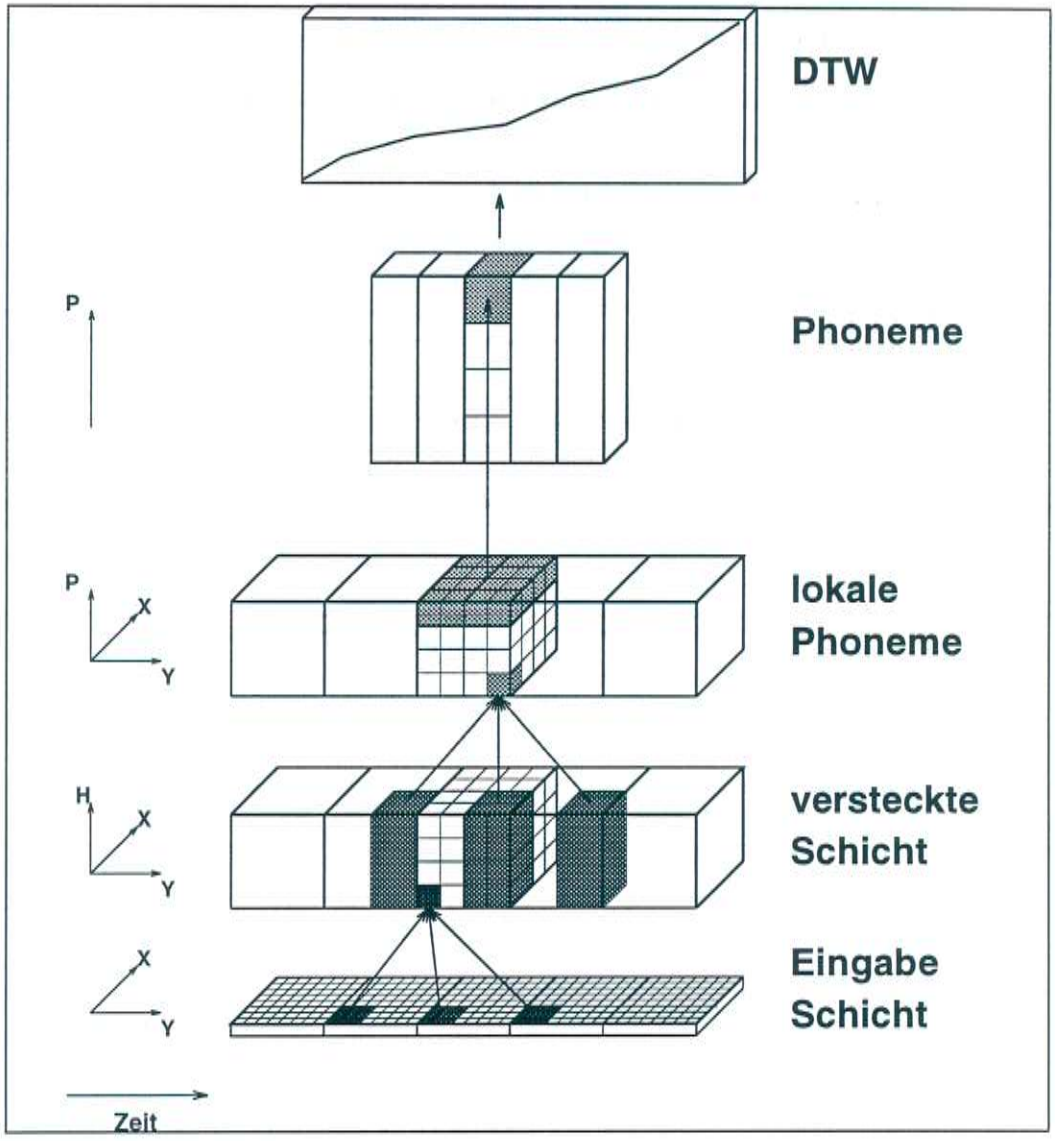

Abbildung 5.3: Schematische Darstellung der MS-TDNN^{3d} Architektur

Fehlerberechnung

Für das Zurückpropagieren des Fehlers wurden 2 Varianten implementiert:

1. **Berechnung des globalen Fehlers**
   Bei dieser Methode wird, wie beim MS-TDNN, der Fehler auf phonetischer Ebene berechnet. Da die Verbindungen zwischen lokalen und globalen Phonemen keine Gewichtung haben, wird dieser Fehler direkt in die lokale Phonemschicht kopiert.

   $$E = -\log\left(1-(y-t)^2\right) = -\log\left(1-d^2\right)$$

   Daraus ergibt sich folgende Ableitung für den Fehler:

   $$\frac{\partial E}{\partial y} = \frac{-1}{1-d^2}(-2d) = \frac{2d}{1-d^2}$$

2. **Berechnung eines lokalen Fehlers**
   Bei dieser Methode wird der Fehler direkt für die lokale Phonem Schicht berechnet:

   $$E = -\ln\left(1-\left(s\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right)-t\right)^2\right)$$

   Daraus ergibt sich folgende Ableitung für den Fehler:

   $$\frac{\partial E}{\partial y_k} = 2\frac{s\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right)}{1-\left(s\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right)-t\right)^2}\frac{\exp\left(-\frac{1}{N}\sum_{k=1}^{N} y_k\right)}{N\left(1+\exp\left(-\frac{1}{N}\sum_{k=1}^{N} y_k\right)\right)}$$

5.2.2 Strategie 2: Auswahl von Teilnetzen

Die Masken werden so groß gewählt, daß sie das zu klassifizierende Objekt (in diesem Fall die Lippen) vollständig umfassen. Die Masken überlappen sich sehr stark, so daß auch kleinere Verschiebungen des Objekts erfaßt werden. Diese Vorgehensweise bietet sich für das Lippenlesen an, da dort eine große Empfindlichkeit gegenüber kleinen Positionsänderungen festgestellt wurde.

Diese Netzarchitektur kann als ein großes Netz angesehen werden, das aus mehreren TDNNs besteht, die sich eine bestimmte Anzahl von Neuronen und Gewichten teilen. Eine extreme Ausprägung dieser Architektur wäre es beispielsweise, die Maskengröße zwischen der versteckten Schicht und der lokalen Phonemschicht in der Größe 1x1 zu wählen. In diesem Fall hätte man mehrere TDNNs, die sich, außer in der Eingabeschicht, keine Daten teilen. Hat man sauber segmentierte Daten, kann dieser Extremfall dazu verwendet werden, mehrere TDNNs zu trainieren, die auf bestimmte Verschiebungen im Bildbereich spezialisiert sind. Im Testfall wird dann, wenn die Daten nicht mehr sauber segmentiert vorliegen, durch die Maximierung das entsprechende Teilnetz ausgewählt.

Forward-Pass

Berechnung:

$$H^i_{x'y'} = s\left(\sum_t \sum_{xy \in mask} w^{it}_{xy} I_{xy}\right)$$

$$L^j_{x'',y''} = \sum_{t'} \sum_{x'y'i \in mask} w^{jt}_{x'y'} H^i_{x'y'}$$

$$P^j = s\left(\frac{1}{n} \max_{x''y''} L^j_{x'',y''}\right)$$

Der Unterschied zur 1. Strategie liegt in der Maximierung über die lokalen Phoneme. Bei dieser Variante des MS-TDNN^{3d} wird davon ausgegangen, daß durch die Masken nicht Teilmuster klassifiziert werden, die dann in einer höheren Schicht zusammengefaßt werden. Stattdessen sollen die Masken so groß gewählt werden, daß sie das gesamte zu klassifizierende Objekt umfassen. In der Phonemschicht wird dann durch Maximierung das Endergebnis berechnet.

Fehlerberechnung

Die Berechnung des Fehlers erfolgt hier analog zu den Methoden der 1. Strategie

5.3 Vorverarbeitung

Die entweder mit dem Aufnahmeverfahren 1 aufgenommenen Lippenregionen bzw. die durch Aufnahmeverfahren 2 und Lippenfinder bestimmten Lippenregionen der Auflösung 144x80 werden auf eine Auflösung von 24x16 reduziert. Anschließend wird die in Abschnitt 4.17 beschriebene Grauwert-Modifikation angewendet. Die daraus resultierenden Bilder werden dann direkt an die Eingabeschicht angelegt.

5.4 Ergebnisse

Die Netze wurden auf den Daten mum9/10 trainiert. Dabei wurden 170 Sequenzen zum Training, 15 zum Cross-Validation und 15 als Testset verwendet. Diese Daten wurden mit dem Aufnahmeverfahren 1 aufgenommen. Als Grauwert-Modifikation wurde eine lineare Anpassung verwendet, d.h. die oberen 5% der Pixel wurden zu 1, die unteren 5% zu 0 gesetzt und die restlichen Grauwerte linear angepaßt. Weiterhin wurden die so trainierten Netze mit künstlich verschobenen Daten getestet. Dazu wurde die Position der 144x80 Pixel grossen Lippenregion um bis zu 6 Pixel in jede Richtung verschoben aus dem Gesichtsbild ausgeschnitten.

Bei dieser Netzarchitektur gibt es eine Vielzahl an möglichen Konfigurationen für das Netz. Hier sollen nur exemplarisch die in Tabelle 5.1 dargestellte Netzkonfigurationen betrachtet werden. Der Hauptunterschied dieser beiden Konfigurationen liegt darin, daß bei Netz 1 mit und bei Netz 2 ohne lokale Phoneme trainiert wird. Tabelle 5.2 zeigt die mit diesen Konfigurationen erzielten Ergebnissen im Vergleich zum Training mit einem MS-TDNN. Wird das MS-TDNN auf nicht künstlich verschobenen Daten trainiert, erzielt es eine deutlich besser Erkennungsleistung auf dem Testset als die MS-TDNN^{3d} Architektur. Werden die Daten für das Training künstlich verschoben, werden vergleichbare Ergebnisse wie beim MS-TDNN^{3d} erzielt, das MS-TDNN^{3d}

mit Netz 2 und lokalem Fehler schneidet dabei besser ab. Testet man diese Architekturen auf künstlich verschobenen Daten (Tabelle 5.3) ergeben sich bei dem MS-TDNN und dem MS-TDNN^{3d} ähnlich schlechte Ergebnisse. Auch hier schneidet die MS-TDNN^{3d} mit der Konfiguration von Netz 2 am besten ab, es kann jedoch nicht von einer Positionsinvarianz gesprochen werden.

Ein Problem bei der MS-TDNN^{3d} Architektur ist die Vielzahl der möglichen Paramtereinstellungen. Aufgrund beschränkter Rechnerkapazitäten und den extrem hohen Laufzeiten bei dieser Architektur konnten nicht alle möglichen Varianten, die diese Architektur bietet, getestet werden. Die beiden hier vorgestellten Konfigurationen sind aus einer großen Anzahl von Versuchen die besten Ergebnisse. In anbetracht der großen Laufzeiten wurden dann allerdings keine weiteren Versuche mehr unternommen, obwohl es sicher noch möglich ist, bessere Parametereinstellungen für diese Architektur zu finden.

Netzkonfigurationen		
	Netz 1	Netz 2
Input Image	24x16	24x16
Input Mask	19x11	19x11
Input Overlap	18x10	18x10
Hidden Unit	6x6	6x6
Hidden Mask	3x3	6x6
Hidden Overlap	0x0	0x0
LocalPhonemes	2x2	1x1

Tabelle 5.1: Netzkonfigurationen

visuelle Erkennungsrate (Word Accuracy)		
	Netz 1	Netz 2
MS-TDNN^{3d} Globaler Fehler, Summe	31%	36%
MS-TDNN^{3d} Globaler Fehler, Maximum	30%	36%
MS-TDNN^{3d} Lokaler Fehler, Summe	32%	41%
MS-TDNN^{3d} Lokaler Fehler, Maximum	32%	41%
MS-TDNN	50%	
MS-TDNN, verschobene Daten	35%	

Tabelle 5.2: Ergebnisse auf mum9/10 (Word Accuracy)

Verhalten auf verschobenen Daten		
	Netz 1	Netz 2
MS-TDNN^{3d} Globaler Fehler, Summe	14%	16%
MS-TDNN^{3d} Globaler Fehler, Maximum	15%	16%
MS-TDNN^{3d} Lokaler Fehler, Summe	13%	12%
MS-TDNN^{3d} Lokaler Fehler, Maximum	13%	12%
MS-TDNN	14%	

Tabelle 5.3: Ergebnisse auf verrauschtem mum9/10 Testset (Word Accuracy)

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit wurde gezeigt, wie man die Problematik unterschiedlicher Beleuchtung, Bildgröße und Lippenposition lösen kann.

Der Einsatz des Face Trackers und des Lippenfinders ermöglicht dem Sprecher eine größere Bewegungsfreiheit. Er kann sich freier vor dem Rechner bewegen und ist nicht mehr an ein Aufnahmefenster gebunden, in dem seine Lippen zentriert sein müssen.

Dieser Vorteil der Bewegungsfreiheit wird allerdings mit einer wesentlich höheren Rechenzeit zur Vorverarbeitung der Daten erkauft. Das Finden und Ausschneiden der Lippenregion und der damit verbundenen Framekorrelation und dem Vergrößern/Verkleinern der Bilder liegt bei einer durchschnittlichen Sequenz von 5 Buchstaben im Bereich von Minuten.

Vorrangiges Ziel bei der Entwicklung dieser Arbeit war es, zu zeigen, daß mit dieser Vorverarbeitung ein robustes Lippenlesen möglich ist. Auf Laufzeit hochoptimierter Code oder der Einsatz von spezieller Hardware wurden dabei nicht berücksichtigt.

Der Hauptanteil der Rechnenzeit wird für das Vergrößern und Verkleinern der Bilddaten benötigt. Es wäre beispielsweise möglich, spezielle Graphik-Hardware zu nutzen und damit diese Operationen in Echtzeit zu berechnen.

Da die Bearbeitung der Bildsequenzen ein inhärent paralleles Problem ist, bietet sich auch der Einsatz von Parallelrechnern oder Multi-Prozessor Systemen an. Auf jedem Prozessor kann dann eine Teilsequenz bearbeitet werden. Auch eine Prozessor-Pipeline zur Bearbeitung der Bilddaten würde sich anbieten.

Mit dem jetzigen System ist ein robustes Lippenlesen möglich. Die nächsten Schritte für das Lippenlesen werden zunächst in Richtung sprecherunabhängigem Buchstabiertask gehen. Weiterhin ist geplant, auf einen größeren kontinuierlichen Task, wie z.B. Wall Street Journal oder auch den FAZ Task zu wechseln.

Literaturverzeichnis

[1] H. Bothe. Sprache – abgesehen von den Lippen. *Spektrum der Wissenschaft*, pages 23–29, November 1993.

[2] H. Bothe. Visual speech and coarticulation effects. *ICASSP*, pages V–634 – V–637, 1993.

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

[4] C. Bregler and Y. Konig. Eigenlips for robust speech recognition. *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing.*

[5] C. Bregler, S. Manke, H. Hild, and A. Waibel. Bimodal sensor integration on the example of speech-reading. *ICNN*, 1993.

[6] C. Bregler, S.M. Omohundor, and Y. Konig. A hybrid approach to bimodal speech recognition. *28th Annual Asimolar conference on Signal speech and Computers.*

[7] U. Bub, M .Hunke, and A. Waibel. Knowing who to listen to in speech recognition: visually guided beamforming. *Proc. Intern. Conference on Acoustics, Speech and Signal Processing*, 1995.

[8] D. Büsching. Automatische Lokalisation von Lippen in Videobildern. Diplomarbeit, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe (TH), Germany, 1994.

[9] M.M. Cohen and D.W. Massaro. Modeling coarticulations in synthetic visual speech. *N.M Thalmann and D.Thalmann, Models and Techniques in Computer Animation.*

[10] M.M. Cohen and D.W. Massaro. What can visual speech synthesis tell visual speech recognition. *28th Annual Asimolar conference on Signal speech and Computers.*

[11] P. Duchnowski, M. Hunke, D. Büsching, U. Meier, and A. Waibel. Toward movement-invariant automatic lip-reading and speech recognition. *Proc. ICASSP*, 1995.

[12] P. Duchnowski, U. Meier, and A. Waibel. See me, hear me: Integrating automatic speech recognition and lip-reading. *International Conference on Spoken Language Processing, ICSLP*, pages 547–550, 1994.

[13] C.G. Fisher. Confusions among visually perceived consonants. *Journal of Speech Hearing*, 11:796–804, 1968. Rockville, Md.

[14] A.J. Goldschen. *Continuous Automatic Speech Recognition by Lipreading.* Dissertation, The School of Engineering and Applied Science of The George Washington University, September 1993.

[15] A.J. Goldschen, O.N. Garcia, and E. Petajan. Continuous optical automatic speech recognition by lipreading. *28th Annual Asimolar conference on Signal speech and Computers.*

[16] P. Haffner and A. Waibel. Multi-state time delay neural networks for continuous speech recognition. In *Neural Information Processing System*, number 4 in NIPS. Morgan Kaufmann, April 1992.

[17] M.E. Hennecke, K.V. Prasad, and D.G. Stork. Using deformable templates to infer visual speech dynamics. *28th Annual Asimolar conference on Signal speech and Computers.*

[18] H. Hild and A.Waibel. Multi-speaker / speaker-independent architectures for the multi-state time delay neural network. *Proc. Intern. Conference on Acoustics, Speech and Signal Processing ,IEEE*, 1993.

[19] Hermann Hild and Alex Waibel. Speaker-Independent Connected Letter Recognition With a Multi-Sate Time Delay Neural Network. In *3rd European Conference on Speech, Communication and Technology (EUROSPEECH) 93*, September 1993.

[20] M. Hunke. Locating and tracking of human faces. *Technical Report CMU-CS-94-155, School of Computer Science, CMU, Pittsburgh, USA.*

[21] M. Hunke. Lokalisieren von Gesichtern mit Neuronalen Netzen. Diplomarbeit, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe (TH), Germany, 1994.

[22] M. Hunke and A. Waibel. Face localisation and tracking for human-computer interaction. *28th Annual Asimolar conference on Signal speech and Computers.*

[23] W. Hürst. Adaptive bimodale Sensorfusion für automatische Spracherkennung und Lippenlesen. Studienarbeit, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe (TH), Germany, 1995.

[24] P.L. Jackson. The theoratical minimal unit for visual speech perception: Visemes and coarticulation. *The Volta Review*, 90(5):99–115, September 1988.

[25] J.D. Keeler, D.E. Rummelhart, and W.K. Leow. Integrated segmentation and recognition of hand-printed numerals. *Advances in neural Information Processing Systems 3*, pages 557–563, 1991.

[26] E.L. Kipila and B. Williams-Scott. Cued speech and speechreading. *The Volta Review*, 90(5):179–189, September 1988.

[27] J.S. Lim. *Two-dimensional signal and image processing.* Prentice Hall.

[28] K. Mase and A. Pentland. Automantic lipreading by optical-flow analysis. *Systems and Computers in Japan*, 22(6):67–76, 1991.

[29] M. McGrath and Q. Summerfield. Intermodal timing relations and audio-visual speech recognition by normal-hearing adults. *Journal of the Acoustical Society of America*, 77(2):678–685, February 1985.

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

[31] U. Meier. Lippenlesen: verschiedene Methoden der visuellen Vorverarbeitung und Merkmalsextraktion. Studienarbeit, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe (TH), Germany, 1994.

[32] A.A. Montgomery, B. Walden, and R. Prosek. Effects of consonantal context on vowel lipreading. *Journal of Speech and Hearing Research*, 30:50–59, 1987.

[33] J. R. Movellan. Visual speech recognition with stochastic networks. *NIPS 94*, 1994.

[34] H.-H. Nagel. *Vorlesungsskript Kognitive Systeme.* Universität Karlsruhe (TH).

[35] H. Ney. The use of a one-stage dynamic programming algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech, Signal Processing ASSP*, 32(2):263–271, April 1984.

[36] E. Owens and B. Blazek. Visems observed by hearing-impaired and normal hearing adult viewers. *Journal of Speech and Hearing Research*, 28:381–393, September 1985. Washington,DC.

[37] E.D. Petajan. Automatic lipreading to enhance speech recognition. *Proc. IEEE Communications Society Global Telecommunications Conference*, 1984.

[38] D.A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance.* Phd thesis, Carnegie Mellon University, Pittsburgh, February 1992.

[39] K.V. Prasad, D.G. Stork, and G.J. Wolff. Preprocessing video images for neural learning of lipreading. *Ricoh California Research Center, Technical Report CRC-TR-93-25.*

[40] W.K. Pratt. *Digital Image Processing.* A Wiley-Interscience Publication.

[41] R.R. Rao and R.M. Mersereau. Lip modeling for visual speech recognition. *28th Annual Asimolar conference on Signal speech and Computers.*

[42] M. Schoch. Schätzung des Signal-Rausch-Abstandes. Studienarbeit, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe (TH), Germany, 1995.

[43] P.L. Silsbee. Sensory integration in audiovisual automatic speech recognition. *28th Annual Asimolar conference on Signal speech and Computers.*

[44] P.L. Silsbee and A.C. Bovic. Audio-visual speech recognition for a vowel discrimination task. *SPIE*, 2049:84–95.

[45] R. Stiefelhagen. Automatische Bestimmung von Visemen für das maschinelle Lippenlesen. Studienarbeit, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe (TH), Germany, 1995.

[46] D.G. Stork, G. Wolff, and E. Levine. Neural network lipreading system for improved speech recognition. *IJCNN*, June 1992.

[47] Q. Summerfield. Audio-visual speech perception, lipreading and artificial stimulation. *Hearing Science and Hearing Disorders*, pages 131–182, 1983. London.

[48] T.Chen, H.P. Graf, and K. Wang. Speech-assisted video processing: interpolation and low-bitrate coding. *28th Annual Asimolar conference on Signal speech and Computers.*

[49] M.T. Vo, R. Houghton, J. Yang, U. Bub, U. Meier, A. Waibel, and P. Duchnowski. Multimodal learning interfaces. *ARPA Spoken Language Technology Workshop*, 1995.

[50] A. Waibel, T. Hanazawa, G. Hinton, and K. Shikano. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3):328–339, 1989.

[51] A. Waibel, M.T. Vo, P. Duchnowski, and S. Manke. Multimodal interfaces. *Artificial Intelligence Review Jounal, special issue*, 1995.

[52] J.T. Wu, S. Tamura, H. Mitsumoto, H. Kawai, K. Kuroso, and K. Okazaki. Neural network vowel-recognition jointly using voice features and mouth shape image. *Pattern Recognition*, 24(10):921–927, 1991.

[53] B.P. Yuhas, M.H. Goldstein, and T.J. Sejnowski. Integration of acoustic and visual speech signals using neural networks. *IEEE Communications Magazine*, pages 65–71, November 1989.

[53] World Wide Web:

`http://mambo.ucsc.edu/psl/lipr.html`
`http://werner.ira.uka.de/~uwe/lippage.html`