AUDIO-VISUAL SPEECH RECOGNITION

Chalapathy Neti (IBM T. J. Watson Research Center, Yorktown Heights),
Gerasimos Potamianos (IBM T. J. Watson Research Center, Yorktown Heights),
Juergen Luettin (Institut Dalle Molle d'Intelligence Artificielle Perceptive, Martigny),
Iain Matthews (Carnegie Mellon University, Pittsburgh),
Herve Glotin (Institut de la Communication Parlée, Grenoble; and
Institut Dalle Molle d'Intelligence Artificielle Perceptive, Martigny),
Dimitra Vergyri (Center for Language and Speech Processing, Baltimore),
June Sison (University of California, Santa Cruz),
Azad Mashari (University of Toronto, Toronto),
and Jie Zhou (The Johns Hopkins University, Baltimore)

Workshop 2000 Final Report

October 12, 2000

Abstract

We have made significant progress in automatic speech recognition (ASR) for well-defined applications like dictation and medium vocabulary transaction processing tasks in relatively controlled environments. However, for ASR to approach human levels of performance and for speech to become a truly pervasive user interface, we need novel, nontraditional approaches that have the potential of yielding dramatic ASR improvements. Visual speech is one such source for making large improvements in high noise environments with the potential of channel and task independence. It is not effected by the acoustic environment and noise, and it possibly contains the greatest amount of complementary information to the acoustic signal. In this workshop, our goal was to advance the state-of-the-art in ASR by demonstrating the use of visual information in addition to the traditional audio for large vocabulary continuous speech recognition (LVCSR). Starting with an appropriate audio-visual database, collected and provided by IBM, we demonstrated for the first time that LVCSR performance can be improved by the use of visual information in the clean audio case. Specifically, by conducting audio lattice rescoring experiments, we showed a 7% relative word error rate (WER) reduction in that condition. Furthermore, for the harder problem of speech contaminated by speech "babble" noise at 10 dB SNR, we demonstrated that recognition performance can be improved by 27% in relative WER reduction, compared to an equivalent audio-only recognizer matched to the noise environment. We believe that this paves the way to seriously address the challenge of speech recognition in high noise environments and to potentially achieve human levels of performance. In this report, we detail a number of approaches and experiments conducted during the summer workshop in the areas of visual feature extraction, hidden Markov model based visual-only recognition, and audio-visual information fusion. The later was our main concentration: In the workshop, a number of feature fusion as well as decision fusion techniques for audio-visual ASR were explored and compared.

Contents

1 Introduction **4**

2 Database, Experimental Framework, and Baseline System **9**

2.1 The Audio-Visual Database 9

2.2 Experiment Framework 10

2.3 Baseline ASR System Training Using HTK 14

3 Visual Feature Extraction **17**

3.1 Discriminant DCT Based Visual Features 18

3.1.1 Face Detection and Mouth Location Estimation 20

3.1.2 Region of Interest Extraction 21

3.1.3 Stage I: DCT Based Data Compression 21

3.1.4 Stage II: Linear Discriminant Data Projection 22

3.1.5 Stage III: Maximum Likelihood Data Rotation 23

3.1.6 Cascade Algorithm Implementation 24

3.1.7 DCT-Feature Visual-Only Recognition Results 24

3.2 Active Appearance Model Visual Features 25

3.2.1 Shape Modeling 26

3.2.2 Shape Free Appearance Modeling 28

3.2.3 Combined Shape and Appearance Model 30

3.2.4 Learning to Fit 33

3.2.5 Training Data and Features 35

3.2.6 Tracking Results 36

3.2.7 AAM-Feature Visual-Only Recognition Results 37

3.3 Summary 38

4 Visual Clustering and Adaptation **40**
4.1 Visual Clustering . . . 40
4.1.1 Viseme Classes . . . 41
4.1.2 Visual Context Questions . . . 42
4.1.3 Phone Tree Root Node Inspection . . . 43
4.1.4 Visual Clustering Experiments . . . 44
4.2 Visual Model Adaptation . . . 46
4.2.1 MLLR Visual-Only HMM Adaptation . . . 47
4.2.2 Adaptation Results . . . 47
4.3 Conclusions . . . 48

5 Models for Audio-Visual Fusion **50**
5.1 Feature Fusion . . . 51
5.1.1 Concatenative Feature Fusion . . . 51
5.1.2 Hierarchical Fusion Using Feature Space Transformations . . . 53
5.1.3 Feature Fusion Results . . . 53
5.2 State Synchronous Decision Fusion . . . 55
5.2.1 The Multi-Stream HMM . . . 55
5.2.2 Multi-Stream HMM Training . . . 56
5.2.3 State Synchronous Fusion Results . . . 57
5.3 Phone Synchronous Decision Fusion . . . 58
5.3.1 The Product HMM . . . 59
5.3.2 Product HMM Training . . . 60
5.3.3 Phone Synchronous Fusion Results . . . 61
5.4 Class and Utterance Dependent Stream Exponents . . . 61
5.4.1 Class Dependent Exponents: Silence Versus Speech . . . 62
5.4.2 Utterance Dependent Stream Exponents . . . 63
5.5 Utterance Level Discriminative Combination of Audio and Visual Models . . . 67
5.5.1 Static Combination . . . 67
5.5.2 Dynamic Combination - Phone Dependent Weights . . . 67
5.5.3 Optimization Issues . . . 68
5.5.4 Experimental Results . . . 68
5.6 Summary . . . 69

6 Summary and Discussion **71**

Acknowledgements **74**

Bibliography **75**

Chapter 1

Introduction

We have made significant progress in *automatic speech recognition* (ASR) for well-defined applications like dictation and medium vocabulary transaction processing tasks in relatively controlled environments. However, for speech to be a *pervasive user interface* in the same league as, for example, graphical user interfaces, it is necessary to make ASR far more *robust* to variations in the environment and channel. Recent studies [55] have shown that ASR performance is far from the human performance in a variety of tasks and conditions. Indeed, ASR to date is very sensitive to variations in the channel (desktop microphone, telephone handset, speakerphone, cellular, etc.), environment (non-stationary noise sources such as speech babble, reverberation in closed spaces such as a car, multi-speaker environments, etc.), and style of speech (whispered, Lombard speech, etc.) [24].

At present, the most effective approach for achieving robustness of environment focuses on obtaining a clean signal through a head-mounted or hand-held directional microphone. However, this is neither tether-free nor hands-free, and it makes speech-based interfaces very unnatural. Moving the speech source away from the microphone can degrade the speech recognition performance due to the contamination of the speech signal by other extraneous sound sources. For example, using monitor microphones for far-field input can severely degrade performance in the presence of noise, but on the other hand using directional desktop microphones constrains the extent of movement of the speaker, thus making the interaction unnatural.

The research work in robust ASR in noise may be classified into three broad areas:

- *Filtering of the noisy speech* prior to classification [50]. In this class of techniques, represented by *spectral subtraction*, an estimate of the clean speech spectrum is obtained by subtracting an average noise spectrum from the noisy speech [6]. A disadvantage of

such techniques is that crucial speech information may be removed during the filtering process.

- *Adaptation* of the speech models to include the effects of noise [36,68]. In this class of techniques, speech models are adapted to include the effects of noise in an attempt to obtain models that would have been obtained in matched conditions.

- Use of *features* that are *robust to noise* [38,46,70]. In this class of techniques, an attempt has been made to incorporate temporal and cross-spectral correlation between speech features modeled after the mammalian auditory processing [38, 70].

These signal-based and model-based techniques to make speech recognition independent of channel and environment have been attempted with limited success [35,50]. Most of these methods make strict assumptions on the environment characteristics and require a sizable sample of the environment to get small improvements in speech recognition performance. Furthermore, modeling reverberation is a hard problem. In summary, current techniques are not designed to work well in severely degraded conditions.

We need novel, nontraditional approaches that use other orthogonal sources of information to the acoustic input that not only significantly improve the performance in severely degraded conditions, but also are independent to the type of noise and reverberation. Visual speech is one such source, obviously not perturbed by the acoustic environment and noise.

It is well known that humans have the ability to lipread: We combine audio and visual information in deciding what has been spoken, especially in noisy environments [92]. A dramatic example is the so-called McGurk effect, where a spoken sound /ga/ is superimposed on the video of a person uttering /ba/. Most people perceive the speaker as uttering the sound /da/ [65]. In addition, the visual modality is well known to contain some complementary information to the audio modality [62]. For example, using visual cues to decide whether a person said /ba/ rather than /ga/ can be easier than making the decision based on audio cues, which can sometimes be confusing. On the other hand, deciding between /ka/ and /ga/ is more reliably done from the audio than from the video channel.

The above facts have recently motivated significant interest in the area of *audio-visual speech recognition* (AVSR), also known as *automatic lipreading*, or *speechreading* [45]. Work in this field aims at improving automatic speech recognition by exploring the visual modality of the speaker's mouth region, in addition to the traditional audio modality. Not surprisingly, automatic speechreading has been shown to outperform audio-only ASR over a wide range of conditions [1, 29, 76, 86, 93]. Such performance gains are particularly impressive in noisy

environments, where traditional ASR performs poorly. Coupled with the diminishing cost of quality video capturing systems, this fact makes automatic speechreading tractable for achieving robust ASR in certain scenarios [45].

However, to date, all automatic speechreading studies have been limited to small vocabulary tasks and, in most cases, to a very small number of speakers [15, 45]. In addition, the number of diverse algorithms suggested in the literature for automatic speechreading are very difficult to compare, as they are hardly ever tested on any common audio-visual database. Furthermore, most such databases are of very small duration, thus placing doubts about the generalizability of reported results to larger populations and tasks. As a result, to date, no definite answers exist on the two issues that are of paramount importance to the design of speaker independent audio-visual *large vocabulary continuous speech recognition* (LVCSR) systems: (a) The choice of appropriate *visual features* that are informative about unconstrained, continuous visual speech; and (b) The design of audio-visual information *fusion* algorithms that demonstrate significant gains over traditional audio-only LVCSR systems, under all possible audio-visual channel conditions.

In the summer 2000 workshop, our goal was to advance the state of the art in audio-visual ASR by seriously tackling the problem of speaker independent LVCSR for the first time. To achieve this goal, we have gathered a team of senior researchers in the area of automatic speechreading with expertise in both visual feature extraction and information fusion [29,63, 71,76], assisted by a number of graduate and undergraduate students [39,97]. In addition, the IBM participants have provided a one-of-a-kind audio-visual database appropriate for LVCSR experiments that has been recently collected at the IBM Thomas J. Watson Research Center [2,80]. The major concentration of the summer workshop team was on audio-visual fusion strategies, however visual feature extraction and certain aspects of visual modeling, as well as visual model adaptation have also been investigated.

In more detail, two algorithms for visual feature extraction have been considered by our workshop team: The first technique belongs to the so called low-level, video pixel based category of visual features [45]. It consists of a cascade of linear transformations of the video pixels representing the speaker's mouth region [80], and it requires successful face and mouth region tracking as a first step [89]. The second technique considered uses a combination of low-level and higher-level, shape based face information [45]. In this approach, both face tracking and feature extraction are based on an active appearance model face representation [19, 30, 63]. High-level shape features have not been considered by themselves in this work, as it is in general agreed that they result in lower speechreading performance [16,29,78].

Both feature sets have been used to train *hidden Markov model* (HMM) based statistical classifiers for recognizing visual-only speech. It is worth mentioning that the visual front end design is not only limited to automatic speechreading: Lip region visual features can readily be used in multimodal biometric systems [33, 49, 100], as well as to detect speech activity and intent to speak [23], among others.

In addition to visual feature extraction, we have investigated various aspects relevant to visual-only HMM training. One important aspect in any LVCSR HMM based system is the issue of clustering of (typically) triphone context dependent units (state or phone models) [82, 103]. Since not all phones are visually distinguishable, but rather they cluster in so-called *viseme* classes [45, 62], it is of interest to investigate whether clustering on basis of visemic instead of phonetic context is advantageous. The design of appropriate visemic questions for tree based HMM state clustering has been addressed in the summer workshop. Another visual modeling issue studied was the problem of visual-only HMM adaptation to unseen subjects. Although visual HMM adaptation has been considered before in small vocabulary tasks [79], this constitutes the first time that successful visual-only model adaptation has been demonstrated in the LVCSR domain.

As stated above, the main concentration of our team has been the audio-visual integration problem. As with visual modeling, HMM only based fusion techniques have been considered in the workshop, although alternative statistical classification methods, such as neural networks, can also be used to address both the speech classification and fusion problems [8, 45, 47].

Two simple *feature fusion* approaches have been tried first. The first one uses the concatenation of synchronous audio and visual feature vectors as the joint audio-visual feature vector, whereas an improved algorithm uses a *hierarchical linear discriminant analysis* (HiLDA) technique to discriminatively project the audio-visual feature vector to a lower dimension.

Subsequently, a number of *decision fusion* algorithms have been investigated. Such algorithms combine the class conditional likelihoods of the audio and visual feature vector streams using an appropriate scoring function at various possible stages of integration. The main model investigated in this approach has been the *multi-stream* HMM. Its class conditional observation likelihood is the product of the observation likelihoods of its audio-only and visual-only stream components, raised to appropriate *stream exponents* that capture the reliability of each modality. Such model has been considered in multi-band audio-only ASR, among others [7, 39, 73]. Although extensively used in small-vocabulary audio-visual ASR tasks [28, 29, 48, 76, 86], this work constitutes its first application to the LVCSR do-

main. Furthermore, to our knowledge, our joint audio-visual multi-stream HMM training by means of *maximum likelihood* estimation has not been considered before. Notice that the multi-stream HMM corresponds in its simplest form to a *state* level integration strategy. By considering the likelihood combination at the HMM *phone* level, we obtain the *asynchronous* multi-stream (*composite*, or *product*) HMM [10, 29, 96], also implemented during the workshop.

In both state and phone level integration strategies, the estimation of appropriate HMM stream exponents is of paramount importance to the resulting model performance. We first considered modality-only based exponents, constant over the entire database. Such exponents were estimated by directly minimizing the word error rate on a held-out data set, since maximum likelihood approaches are inappropriate for training them [76, 103]. Alternative *discriminative* training techniques can also be used for that task [17,18,48,76]. Motivated by the fact that the audio of various speakers and utterances is characterized by varying signal to noise ratio (and thus audio channel reliability), we subsequently refined the stream exponents by making them *utterance dependent* as well. We used a *harmonicity index* [4,39,105] to estimate the average *voicing* per utterance, and we estimated exponents based on this index.

Finally, a late integration, decision fusion technique has been explored based on rescoring *N-best* recognition hypotheses using the general framework of multiple knowledge source integration for ASR developed in [97]. Global, viseme-, and phone-dependent audio-visual weights were explored in this approach, all estimated by means of *minimum error training* on a held-out data set.

In this report, we discuss in detail our summer work. Specifically, in chapter 2, we present the audio-visual database, our general experiment framework, as well as our audio-only baseline system and its training procedure. In chapter 3, we discuss the two visual feature extraction techniques considered at the workshop, and we present visual-only LVCSR results. In chapter 4, we concentrate on two issues relevant to visual modeling, namely visual-only clustering and visual model adaptation. In chapter 5, we report our work on HMM based audio-visual fusion. We first present two feature fusion algorithms, followed by a number of decision fusion techniques at the state, phone, and utterance level. Finally, in chapter 6, we summarize our most important results, and we discuss plans for future work.

Chapter 2

Database, Experimental Framework, and Baseline System

In this chapter, we first present the audio-visual database used in all our summer workshop experiments (section 2.1). In section 2.2, we give an overview of our experimental paradigm. We include information about the database partitioning, the set of audio and visual features used in the experiments, the clean and noisy audio conditions considered, and, finally, the sets of lattices generated pre-workshop at IBM. Such lattices, were rescored by appropriate models, trained using the HTK software toolkit [103] as described in section 2.3.

2.1 The Audio-Visual Database

To allow experiments on continuous, large vocabulary, speaker independent audio-visual speech recognition, a suitable database has been collected at the IBM Thomas J. Watson Research Center, preceding the summer workshop. The database consists of full-face frontal video and audio of 290 subjects (see also Figure 2.1), uttering ViaVoice^{TM} training scripts, i.e., continuous read speech with mostly verbalized punctuation (dictation style), and a vocabulary size of approximately 10,500 words. Transcriptions of all 24,325 database utterances, as well as a pronunciation dictionary are provided. The database video is of size 704×480 pixels, interlaced, captured in color at a rate of 30 Hz (i.e., 60 fields per second are available at a resolution of 240 lines), and it is MPEG2 encoded at the relatively high compression ratio of about 50:1. High quality wideband audio is synchronously collected with the video at a rate of 16 kHz and at a relatively clean audio environment (quiet office, with some background computer noise). The duration of the entire database

Figure 2.1: Example video frames of the IBM ViaVoice^{TM} audio-visual database.

is approximately 50 hours. It is worth mentioning that, to date, this is the largest audio-visual database collected, and it constitutes the only one suitable for the task of continuous, large vocabulary, speaker independent audio-visual speech recognition, as all other existing audio-visual databases are limited to small number of subjects and/or small vocabulary tasks [1, 8, 13, 15, 45, 64, 66, 67, 75, 93].

In addition to the IBM ViaVoice^{TM} audio-visual database, a much smaller broadcast news dataset has also been obtained both at the IBM Thomas J. Watson Research Center and at the Johns Hopkins University, preceding the workshop. This database contains audio-visual sequences of frontal anchor speech, and it has been digitized from CNN and CSPAN broadcast news tapes, kindly provided by the Linguistic Data Consortium (LDC). The entire duration of the database is approximately 5 hours, and it has been collected with the intent of performing audio-visual speaker adaptation experiments, using HMMs trained on the ViaVoice^{TM} data. However, the short duration of the summer workshop did not allow us to complete visual feature extraction for this data. We hope to perform such experiments in the future.

2.2 Experiment Framework

The audio-visual database has been partitioned into a number of disjoint sets in order to train and evaluate models for audio-visual ASR (see also Table 2.1). The *training* set contains 35 hours of data from 239 subjects, and it is used to train all HMMs reported in this work. Two more sets are provided for conducting *speaker-independent* (SI) HMM refinement and

Scenario	Set	Utter.	Duration	Subj.
SI/MS	Training	17111	34.9 hrs	239
SI	Held-out	2277	4.8 hrs	25
	Adaptation	855	2.1 hrs	26
	Test	1038	2.5 hrs	26
MS	Held-out	1944	4.0 hrs	239
	Test	1100	2.3 hrs	239
	Total	24325	50.6 hrs	290

Table 2.1: Database partitioning for speaker independent (SI) and multi-speaker (MS) experiments. Number of utterances, duration, and number of subjects are depicted for each set. A single training set is used in both SI and MS scenarios (SI only experiments are reported in this work).

testing: A *held-out* data set of close to 5 hours of data from 25 subjects and a *test* set of 2.5 hours from 26 subjects. The first is used to train HMM parameters relevant to audio-visual decision fusion (see section 5), while the second is used for testing (evaluation) of the trained models. Of course, all three sets comprise of disjoint subjects. Furthermore, an *adaptation* set is provided to allow speaker adaptation experiments (see section 4.2). This set contains an additional 2 hours of data from the 26 test set subjects. In addition to the above mentioned sets, two more sets are available for *multi-speaker* (MS) HMM refinement and testing, namely a 4 hour held-out data set and a 2.3 hour test set, both containing data from all 239 training set subjects. The later were created in case speaker-independent visual models provided poor generalization to unseen subjects. Our results during the initial weeks of the workshop indicated that this was not the case, therefore, in this report, only speaker-independent experiments are reported.

To assess the benefits of the visual modality to LVCSR for both clean and noisy audio, two audio conditions have been considered: The original database clean wideband audio, and a degraded one, where the database audio is artificially corrupted by additive "*babble*" noise[1] at a 10 db SNR level. Sixty-dimensional acoustic feature vectors are extracted for both conditions at a rate of 100 Hz [2]. These features are obtained by a *linear discriminant analysis* (LDA) data projection, applied on a concatenation of nine consecutive feature frames consisting of a 24-dimensional *discrete cosine transform* (DCT) of mel-scale filter bank energies. LDA is followed by a *maximum likelihood linear transform* (MLLT) based data rotation (see section 3.1 for details on these transforms). *Cepstral mean subtraction* (CMS) and *energy*

[1]This noise consists of simultaneous speech by multiple subjects, recorded at IBM.

normalization [56, 103] are applied to the DCT features at the utterance level, prior to the LDA/MLLT feature projection. It is worth mentioning, that, for both clean and noisy audio, the LDA and MLLT matrices are estimated using the training set data in the *matched* condition. Similarly, all audio-only test set results are reported for HMMs trained on matched audio. For the noisy audio-only system, this is clearly an ideal scenario, which results in improved audio-only performance over systems that use noise compensation techniques when trained on unmatched data.

In addition to the audio features, visual features need to be extracted in order to perform audio-visual speech recognition experiments. As mentioned in the Introduction and discussed in detail in chapter 3, two types of visual features have been considered in this work. The baseline ones consist of a discrete cosine image transform of the subject's mouth region, followed by an LDA projection and an MLLT feature rotation [80]. They have been provided by the IBM participants for the entire database, are of dimension 41, and are synchronous to the audio features at a rate of 100 Hz (see section 3.1). These baseline features are exclusively used in our audio-visual ASR experiments. Alternative visual features based on *active appearance models* are presented in section 3.2, and preliminary visual-only recognition results are reported there. Notice that, in contrast to the audio, no noise has been added to the video channel or features. Many such cases of "visual noise" could have been considered, for example additive white noise on the video frames, blurring, frame rate reduction, and extremely high compression factors, among others. Some preliminary studies on the effects of video degradation to speechreading can be found in [22, 78, 101].

Given the training set utterance transcriptions, the corresponding appropriate features, and the pronunciation dictionary, we can train an HMM based ASR system [82, 103]. However, due to the HTK large memory and speed requirements for LVCSR *decoding*, and in order to allow fast experimentation, we have decided to follow a *lattice rescoring* based decoding strategy. Namely, using a well trained HMM system, we first generate appropriate ASR lattices off line, that contain the "most probable" decoding paths. Subsequently, we rescore these lattices using various HTK-trained HMM systems of interest based on a number of feature sets, fusion strategies, etc. Baseline HTK systems are trained as discussed in section 2.3. For rescoring, we employ the HTK decoder (HVite) that runs efficiently, since the search is constrained by the lattice (grammar) [103]. Notice that the generated lattices are *trigram* lattices [82], and that, on every lattice arc, the log-likelihood value of the trigram *language model* used to generate them has been provided by IBM. During rescoring, the language model weight and the word insertion penalty are roughly optimized by seeking

Lattices	Best	Oracle	Anti-oracle	LM-only	Depth
"Lat"	14.24	5.53	46.83	29.57	64.7
"NLat"	45.43	26.81	96.12	58.31	164.5
"NAVLat"	37.15	16.84	103.69	52.02	271.2

Table 2.2: Word error rate (WER %) of the IBM generated lattices on the SI test set. WER for best path, oracle, anti-oracle, and best path based on language model information alone (LM-only) are depicted. Average lattice depth in words per reference transcription length is also shown.

minimum *word error rate* (WER) on the held-out set. Test set results are reported based on the NIST scoring standard [103].

For the summer workshop experiments, we have generated *three* sets of lattices for all database utterances not belonging to the training set, using the IBM LVCSR recognizer and appropriately trained HMM systems at IBM (cross-word pentaphone systems, with about 50,000 Gaussian mixtures each). The three sets of lattices are:

- *Lat*: Lattices based on the IBM system with *clean audio* features.
- *NLat*: Lattices based on the IBM system with *noisy audio* features (matched training).
- *NAVLat*: Lattices based on the IBM system with *noisy audio-visual* features, using the HiLDA feature fusion technique reported in section 5.1.2.

Table 2.2 depicts the lattice word error rates, as well as other useful lattice information. Lattices "Lat" and "NLat" are rescored by HTK trained systems on clean and noisy audio features, respectively, to provide the baseline clean and noisy audio-only ASR performance. For visual-only recognition experiments, lattices "NLat" are used, because they have the worst accuracy (see Table 2.2). Such experiments are used to investigate the relative performance of the visual features of sections 3.1 and 3.2 and of the various visual modeling and adaptation techniques in chapter 4. The absolute visual-only recognition numbers reported there are clearly meaningless, as they are based on rescoring lattices that contain audio information! Finally, audio-visual fusion experiments are reported by rescoring the "Lat" lattices in the clean audio case. However, the "NAVLat" lattices are used in the noisy audio-visual fusion experiments, because, in this case, performance improves significantly by adding the visual modality (see Table 2.2).

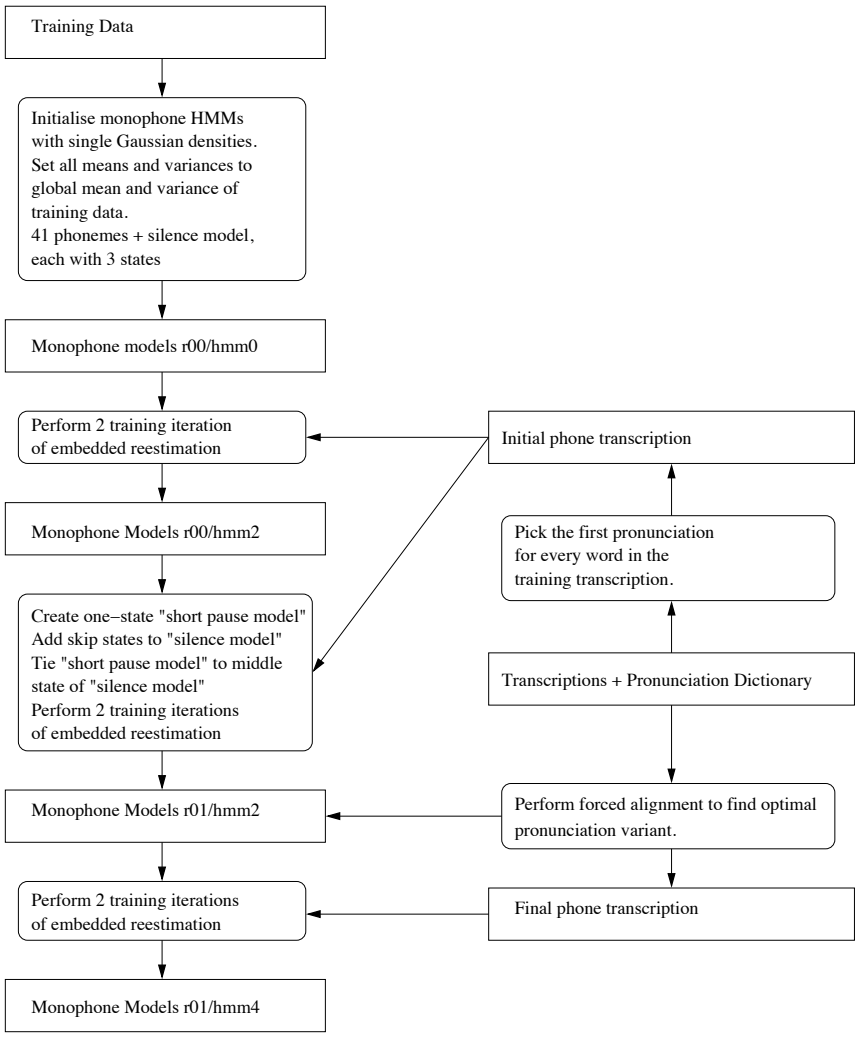


Figure 2.2: Training procedure for monophone HMMs.

2.3 Baseline ASR System Training Using HTK

This section describes the baseline speech recognition system that has been developed. The aim of this system is to represent a state-of-the-art ASR reference system that has similar performance characteristics to the IBM ViaVoice^{TM} system that generated the lattices, and, in addition, it represents a baseline system to which the performance of the developed audio-visual ASR systems can be compared to. The baseline system can be trained with one set of feature vectors that can be audio-only, visual-only, or audio-visual ones (section 5.1). Context dependent phoneme models are used as speech units, and they are modeled with HMMs with *Gaussian mixture* class-conditional observation probabilities. These are trained based

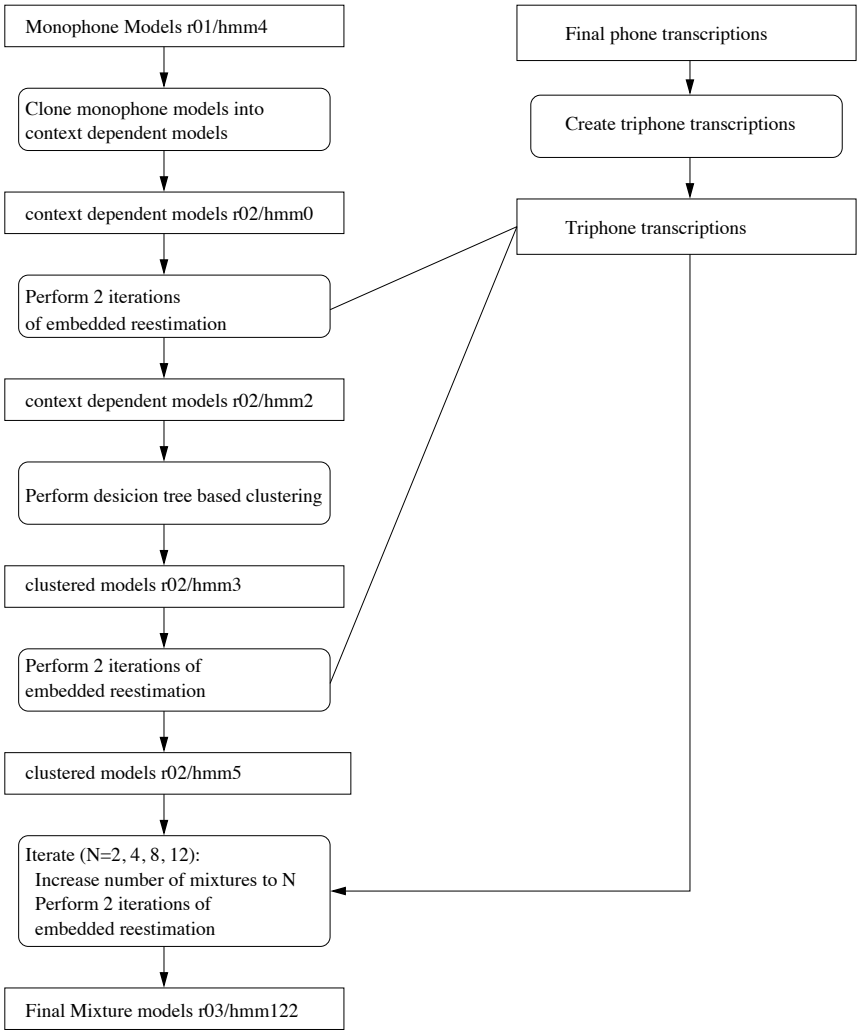


Figure 2.3: Additional training steps for context dependent HMMs.

on maximum likelihood estimation using embedded training by means of the *Expectation-Maximization* (EM) algorithm [25, 82].

The baseline system was developed using the HTK toolkit version 2.2 [103]. The training procedure is illustrated in Figures 2.2 and 2.3. This training procedure is similar to the one described in the HTK reference manual [103] and also to baseline systems developed during previous summer workshops at the Johns Hopkins University. All HMMs had 3 states except the short pause /sp/ model that had only one state. We have used a set of 41 phonemes. The phoneme /el/ has been replaced by the phoneme sequence /eh l/ due to the very small number of occurrences of /el/ in our data. The first training step initializes the *monophone* models with single Gaussian densities. All means and variances are set to the global means

Condition / Lattices	HTK	IBM
Clean-audio / Lat	14.44	14.24
Noisy-audio / NLat	48.10	45.43

Table 2.3: HTK baseline audio-only WER (%) obtained by rescoring the IBM generated lattices on the SI test set. Performance of the IBM system lattices is also depicted.

and variances of the training data. Monophones are trained by embedded reestimation using the first pronunciation variant in the pronunciation dictionary. A short pause model /sp/ is subsequently added and tied to the center state of the silence model /sil/, followed by another 2 iterations of embedded reestimation. *Forced alignment* is then performed to find the optimal pronunciation in case of multiple pronunciation variants in the dictionary. The resulting transcriptions are used from now on for further training steps. Another 2 iterations of embedded reestimation lead to the trained monophone models.

Context dependent phone models are obtained by first cloning the monophone models into context dependent phone models, followed by 2 training iterations using *triphone* based transcriptions. *Decision tree* based clustering is then performed to cluster phonemes with similar context and to obtain a smaller set of context dependent phonemes. This is followed by 2 training iterations. Finally, Gaussian mixture models are obtained by iteratively splitting the number of mixtures to 2, 4, 8, and 12, and by performing two training iterations after each splitting.

The training procedure has been the same for all parameter sets, whether audio-only, visual-only, or audio-visual. The resulting baseline clean and noisy audio-only system performance, obtained by rescoring lattices "Lat" and "NLat", respectively, was 14.44% and 48.10% WER (see also Table 2.3). These numbers are quite close to the ones obtained by the IBM system, therefore our goal of obtaining comparable baseline performance between the IBM and HTK systems has been achieved.

Chapter 3

Visual Feature Extraction

As discussed in the Introduction, the first main problem in the area of speechreading is the question of appropriate visual speech representation in terms of a small number of informative features. Various sets of visual features have been proposed for this purpose in the literature over the last 20 years. In general, they can be grouped into three categories: High-level *lip contour* based features, low-level *video pixel* based ones, and features that are a combination of both [45].

In the first approach, the speaker's inner and (or) outer lip contours are extracted from the image sequence. A parametric, or statistical lip contour model is then obtained [3,16,45, 51,52,60,84,87,90,104], and the model parameters are used as visual features. Alternatively, lip contour geometric features are used, such as mouth height and width [1,13,74,78,85,86].

In the second approach, the entire image containing the speaker's mouth is considered as informative for lipreading (*region of interest* - ROI), and appropriate transformations of its pixel values are used as visual features. For example, in [44] video frame ROI differences are used, whereas in [64] a nonlinear image decomposition for feature extraction is suggested. The most popular low-level feature representation is a *principal component analysis* (PCA) based ROI projection [2,9–11,16,28,29,61,78]. Alternative image transforms of the ROI such as the *discrete cosine transform* (DCT) [27,80] and the *discrete wavelet transform* [75,78] have also been used for feature extraction. A DCT based feature extraction scheme was used in the summer workshop, as described in section 3.1.

Often, the high- and low-level feature extraction approaches are combined to give rise to joint shape and appearance visual features [16,28,29,57,61,63]. Such is the case with the *active appearance model* (AAM) based features [19,30,64] that are presented in section 3.2.

A number of techniques can be used to post-process the extracted visual features in or-

der to improve visual-only *discrimination* among the speech classes of interest, or to provide better visual data maximum likelihood modeling. Such techniques considered in this work are the *linear discriminant analysis* (LDA) [83], as well as a *maximum likelihood linear transformation* (MLLT) of the data, which is aimed at optimizing the observed data likelihood under the assumption of class conditional multi-variate normal distribution with diagonal covariance [42]. For visual speech extraction, LDA has been used as a stand-alone visual front end in [27, 77], and as the second and final visual front end stage (following the application of PCA) in [2, 100]. The visual front end used in the workshop is a cascade of a DCT of the mouth ROI, followed by LDA and MLLT, as in [80]. The three stages of this visual front end are described in the following section. Note that both LDA and MLLT are general pattern recognition and modeling techniques, and, as such, they have also been used in the AAM feature visual-only recognition experiments (see section 3.2.7), as well as in our audio-visual feature fusion work (section 5.1.2).

3.1 Discriminant DCT Based Visual Features

The DCT based visual feature extraction algorithm used in the summer workshop constitutes a pure video pixel, appearance based feature representation of the visual speech activity region, i.e., the immediate face area including and surrounding the subject's mouth. The algorithm comprises of the following five steps, which include three stages of a cascade of linear transformations applied to an appropriate visual data region of interest:

- Face detection and mouth location estimation, discussed in section 3.1.1;
- Region of interest (ROI) extraction, as presented in section 3.1.2;
- *Stage I:* Discrete cosine transform of the ROI (see section 3.1.3);
- *Stage II:* Linear discriminant (LDA) based DCT feature projection (section 3.1.4); and
- *Stage III:* Maximum likelihood feature rotation (MLLT), discussed in section 3.1.5.

The schematic of the algorithm is depicted in Figure 3.1. Implementation details, including some DCT feature post-processing following Stage I, are presented in section 3.1.6. Visual-only recognition experiments are reported in section 3.1.7.

The algorithm requires the use of a highly accurate face and mouth region detection system (e.g., [43, 89]) as its first step. Subsequently, for every video frame $\{ V_t(m, n) \}$, at

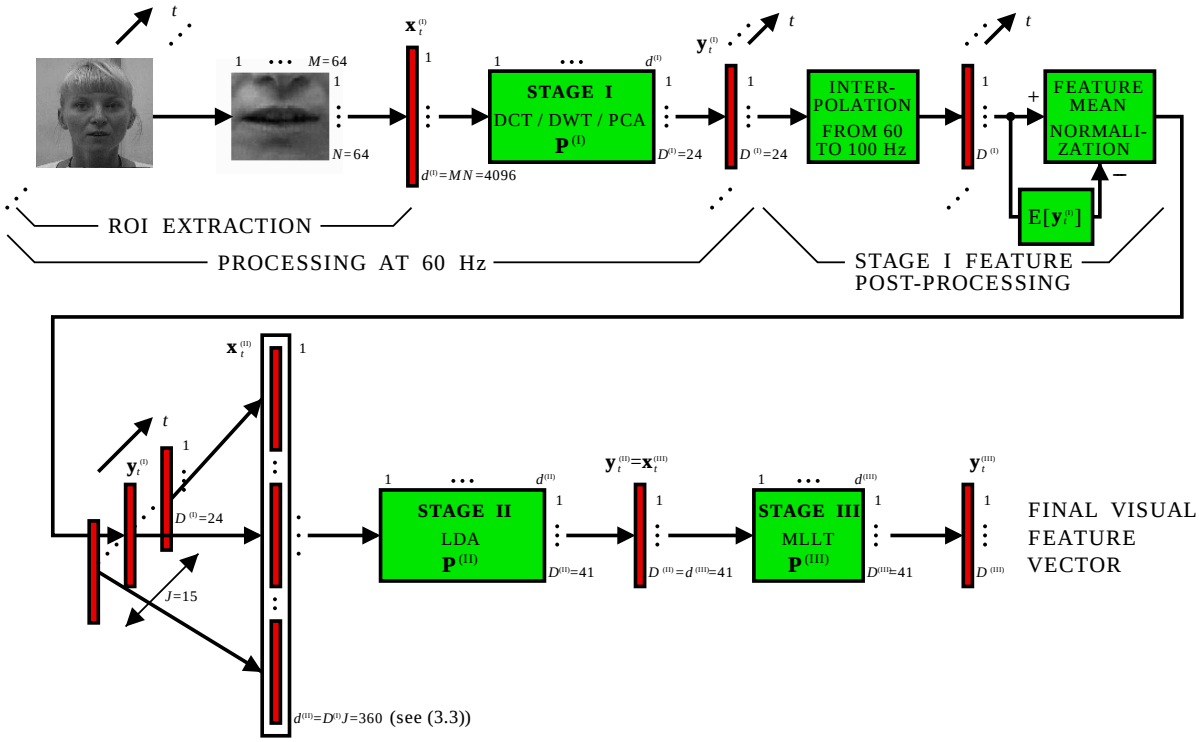


Figure 3.1: The DCT based cascade algorithm block diagram of the visual front end used in our audio-visual ASR experiments.

time t, the two-dimensional ROI centered around the speaker's mouth center (m_t, n_t), is extracted, as discussed in the following sections. The ROI video pixel values are then placed into the vector[1]

$$\mathbf{x}_t^{(\mathrm{I})} \leftarrow \{\, V_t(m,n) :\; m_t - \lfloor M/2 \rfloor \le m < m_t + \lceil M/2 \rceil\,,$$

$$n_t - \lfloor N/2 \rfloor \le n < n_t + \lceil N/2 \rceil \,\}\,, \tag{3.1}$$

of length $d^{(\mathrm{I})} = MN$. The proposed three-stage cascade algorithm seeks three matrices, $\mathbf{P}^{(\mathrm{I})}$, $\mathbf{P}^{(\mathrm{II})}$, and $\mathbf{P}^{(\mathrm{III})}$, that when applied to the data vector $\mathbf{x}_t^{(\mathrm{I})}$, in a cascade fashion, they result in a compact visual feature vector $\mathbf{y}_t^{(\mathrm{III})}$ of dimension $D^{(\mathrm{III})} \ll d^{(\mathrm{I})}$ (see also Figure 3.1). Such vector should contain most discriminant and relevant to visual speech information, according to criteria defined in sections 3.1.3, 3.1.4, and 3.1.5. Each matrix $\mathbf{P}^{(\bullet)}$ is of dimension $D^{(\bullet)} \times d^{(\bullet)}$, where $\bullet = \mathrm{I}, \mathrm{II}, \mathrm{III}$. To obtain matrices $\mathbf{P}^{(\bullet)}$, L *training* examples are given, denoted by $\mathbf{x}_l^{(\mathrm{I})}$,

[1] Throughout this work, boldface lowercase symbols denote column vectors, and boldface capital symbols denote matrices.

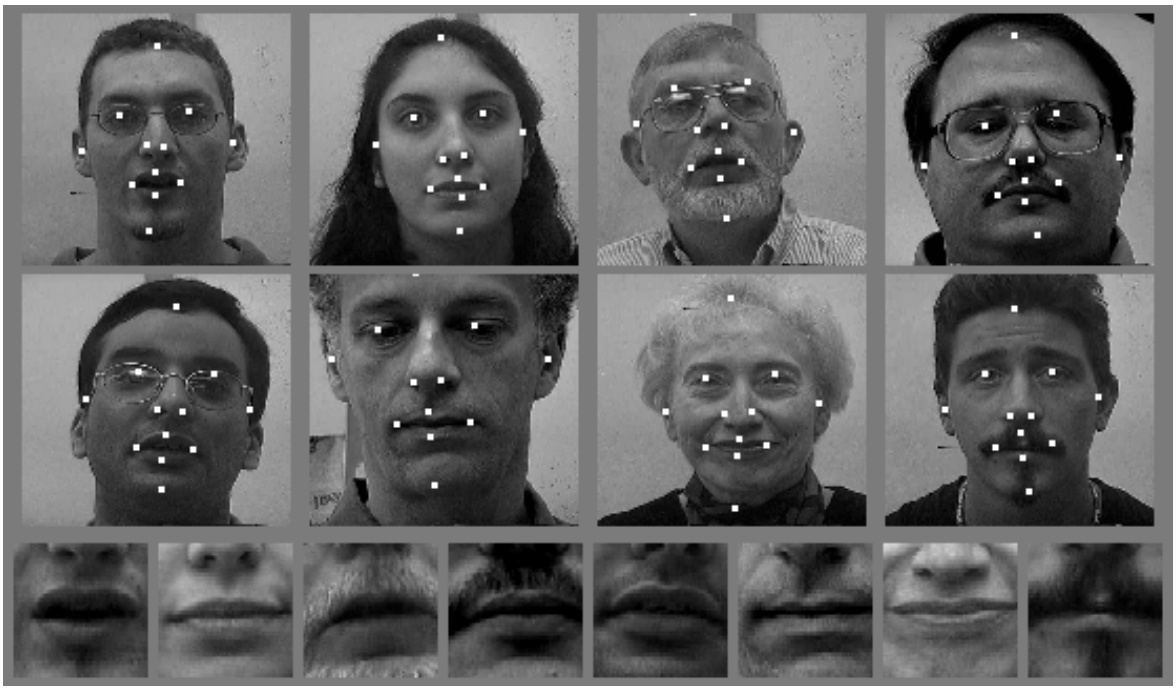


Figure 3.2: Region of interest extraction examples. *Upper rows*: Example video frames from 8 database subjects, with detected facial features superimposed. *Lower row*: Corresponding extracted mouth regions of interest.

for $l = 1,...,L$.

3.1.1 Face Detection and Mouth Location Estimation

We use the face detection and facial feature localization method described in [89]. Given a video frame, face detection is first performed by employing a combination of methods, some of which are also used for subsequent face feature finding. A face template size is first chosen (11×11 pixels, here), and an image pyramid over the permissible scales (given the frame size and the face template) is used to search the image space for the possible face candidates. Since the video signal is in color, skin-tone segmentation is first used to narrow this search to candidates that contain a significantly high proportion of skin-tone pixels. Every remaining face candidate is given a score based on both a two-class Fisher linear discriminant [83] and its *distance from face space* (DFFS). All candidate regions exceeding a threshold score are considered as faces.

Once a face has been found, an ensemble of facial feature detectors are used to extract and verify the locations of 26 facial features, including the lip corners and centers (ten

such facial features are marked on the frames of Figure 3.2). The search for these features occurs hierarchically. First, a few "high"-level features are located, and, subsequently, the 26 "low"-level features are located relative to the "high"-level feature locations. The feature locations at both stages are determined using a score combination of prior statistics, linear discriminant and DFFS [89].

The algorithm requires a training step to estimate the Fisher discriminant, face space eigenvectors, and prior statistics for face detection and facial feature estimation. Such training uses a number of frames labeled with the faces and their visible features (see also section 3.1.6).

3.1.2 Region of Interest Extraction

Given the output of the face detection and facial feature finding algorithm described above, five located lip contour points are used to estimate the mouth center and its size at every video frame (four such points are marked on the frames of Figure 3.2). The mouth center estimate is smoothed over twenty neighboring frames using median filtering to obtain the ROI center $(m_t\,, n_t)$, whereas the mouth size estimate is averaged over each utterance. A size normalized ROI is then extracted as in (3.1), with $M = N = 64$, in order to allow for fast DCT implementation (see also Figure 3.2). ROI *greyscale* only pixel values are placed in $\mathbf{x}_t^{(\mathrm{I})}$. Furthermore, in our current implementation, no rotation normalization, general three-dimensional pose compensation, or lighting normalization is directly applied to the ROI.

3.1.3 Stage I: DCT Based Data Compression

At the first algorithm stage, we seek a $D^{(\mathrm{I})} \times d^{(\mathrm{I})}$-dimensional *linear transform* matrix $\mathbf{P}^{(\mathrm{I})} = [\,\mathbf{p}_1,...,\mathbf{p}_{D^{(\mathrm{I})}}]^\top$, such that the transformed data vector $\mathbf{y}_t^{(\mathrm{I})} = \mathbf{P}^{(\mathrm{I})}\mathbf{x}_t^{(\mathrm{I})}$ contains most speechreading information in its $D^{(\mathrm{I})} \ll d^{(\mathrm{I})}$ elements, thus achieving significant data compression. This can be quantified by seeking such elements to maximize the total *energy* of the transformed training feature vectors $\mathbf{y}_l^{(\mathrm{I})} = \mathbf{P}^{(\mathrm{I})}\mathbf{x}_l^{(\mathrm{I})}$, for $l = 1,...,L$, given the desired output vector length $D^{(\mathrm{I})}$ (see (3.2), below). Alternatively, one could seek to minimize the *mean square error* between the training data vectors $\mathbf{x}_l^{(\mathrm{I})}$ and their reconstruction based on $\mathbf{y}_l^{(\mathrm{I})}$, for $l = 1,...,L$, as in PCA [14].

A number of linear, *separable* image transforms can be used in place of $\mathbf{P}^{(\mathrm{I})}$. In this work, we consider the DCT. Let square matrix $\mathbf{B} = [\,\mathbf{b}_1,...,\mathbf{b}_{d^{(\mathrm{I})}}]^\top$ denote the DCT matrix,

where $\bullet^\top$ denotes vector or matrix *transpose*. Then, matrix $\mathbf{P}^{(\mathrm{I})}$ contains as its rows the rows of $\mathbf{B}$ that maximize the transformed data *energy*

$$\sum_{d=1}^{D^{(\mathrm{I})}} \sum_{l=1}^{L} < \mathbf{x}_l^{(\mathrm{I})}, \mathbf{b}_{j_d} >^2, \tag{3.2}$$

where $j_d \in \{1,...,d^{(\mathrm{I})}\}$ are disjoint, and $<\bullet\,,\bullet>$ denotes vector *inner product*. Obtaining the optimal values of j_d, for $d = 1,...,D^{(\mathrm{I})}$, that maximize (3.2) is straightforward. It is important to note that DCT allows fast implementations [81] when M and N are powers of 2. It is therefore advantageous to choose such values in (3.1).

3.1.4 Stage II: Linear Discriminant Data Projection

In the proposed cascade algorithm, and in order to capture important *dynamic* visual speech information, linear discriminant analysis (LDA) is applied to the concatenation of J consecutive image transformed feature vectors

$$\mathbf{x}_t^{(\mathrm{II})} = [\, \mathbf{y}_{t-\lfloor J/2 \rfloor}^{(\mathrm{I})\top}, ..., \mathbf{y}_t^{(\mathrm{I})\top}, ..., \mathbf{y}_{t+\lceil J/2 \rceil - 1}^{(\mathrm{I})\top} \,]^\top, \tag{3.3}$$

of length $d^{(\mathrm{II})} = D^{(\mathrm{I})} J$ (see also Figure 3.1).

In general, LDA [83] assumes that a set of *classes* $\mathcal{C}$ is a-priori given, as well as that the training set data vectors $\mathbf{x}_l^{(\mathrm{II})}$, $l = 1,...,L$, are *labeled* as $c(l) \in \mathcal{C}$. LDA seeks a projection $\mathbf{P}^{(\mathrm{II})}$, such that the projected training sample $\{\, \mathbf{P}^{(\mathrm{II})}\, \mathbf{x}_l^{(\mathrm{II})},\, l = 1,...,L \,\}$ is "well separated" into the set of classes $\mathcal{C}$. Formally, $\mathbf{P}^{(\mathrm{II})}$ maximizes

$$Q(\mathbf{P}^{(\mathrm{II})}) = \frac{\mathsf{det}\,(\,\mathbf{P}^{(\mathrm{II})\top}\, \mathbf{S}_B\, \mathbf{P}^{(\mathrm{II})})}{\mathsf{det}\,(\,\mathbf{P}^{(\mathrm{II})\top}\, \mathbf{S}_W\, \mathbf{P}^{(\mathrm{II})})}\,, \tag{3.4}$$

where $\mathsf{det}(\bullet)$ denotes matrix *determinant*. In (3.4), $\mathbf{S}_W$, $\mathbf{S}_B$ denote the *within-class scatter* and *between-class scatter* matrices of the training sample. These matrices are given by

$$\mathbf{S}_W = \sum_{\mathsf{c} \in \mathcal{C}} Pr(\mathsf{c})\, \Sigma^{(\mathsf{c})}, \quad \text{and} \quad \mathbf{S}_B = \sum_{\mathsf{c} \in \mathcal{C}} Pr(\mathsf{c})\, (\,\mathbf{m}^{(\mathsf{c})} - \mathbf{m}\,)\, (\,\mathbf{m}^{(\mathsf{c})} - \mathbf{m}\,)^\top, \tag{3.5}$$

respectively. In (3.5), $Pr(\mathsf{c}) = L_\mathsf{c}/L$, $\mathsf{c} \in \mathcal{C}$, is the class empirical probability mass function, where $L_\mathsf{c} = \Sigma_{l=1}^{L} \delta_{c(l)}^{\mathsf{c}}$, and $\delta_i^j = 1$, if $i = j$; 0, otherwise. In addition, each class sample mean

is

$$\mathbf{m}^{(\mathsf{c})} = [\, m_1^{(\mathsf{c})},...,m_{d^{(\mathrm{II})}}^{(\mathsf{c})}\,]^\top, \quad \text{where} \;\; m_d^{(\mathsf{c})} = \frac{1}{L_\mathsf{c}} \sum_{l=1}^{L} \delta_{c\,(l)}^{\,\mathsf{c}}\, x_{l,d}^{(\mathrm{II})}\,, \quad \text{for} \;\; d = 1,...,d^{(\mathrm{II})}\,,$$

and each class sample covariance is $\Sigma^{(\mathsf{c})}$, with elements given by

$$\sigma_{d,d'}^{(\mathsf{c})} = \frac{1}{L_\mathsf{c}} \sum_{l=1}^{L} \delta_{c\,(l)}^{\,\mathsf{c}} (\, x_{l,d}^{(\mathrm{II})} - m_d^{(\mathsf{c})})\,(\, x_{l,d'}^{(\mathrm{II})} - m_{d'}^{(\mathsf{c})})\,, \quad \text{for} \;\; d\,, d' = 1,...,d^{(\mathrm{II})}\,.$$

Finally, $\mathbf{m} = \Sigma_{\,\mathsf{c}\in\mathcal{C}}\, Pr(\mathsf{c})\, \mathbf{m}^{(\mathsf{c})}$, denotes the total sample mean.

To maximize (3.4), we subsequently compute the *generalized* eigenvalues and *right* eigenvectors of the matrix pair ($\mathbf{S}_B$,$\mathbf{S}_W$) that satisfy $\mathbf{S}_B\mathbf{F} = \mathbf{S}_W\mathbf{F}\mathbf{D}$ [41, 83]. Matrix $\mathbf{F} = [\,\mathbf{f}_1,...,\mathbf{f}_{d^{(\mathrm{II})}}]$ has as columns the generalized eigenvectors. Let the $D^{(\mathrm{II})}$ largest eigenvalues be located at the $j_1,...,j_{D^{(\mathrm{II})}}$ diagonal positions of $\mathbf{D}$. Then, given data vector $\mathbf{x}_t^{(\mathrm{II})}$, we extract its feature vector of length $D^{(\mathrm{II})}$ as $\mathbf{y}_t^{(\mathrm{II})} = \mathbf{P}^{(\mathrm{II})}\mathbf{x}_t^{(\mathrm{II})}$, where $\mathbf{P}^{(\mathrm{II})} = [\,\mathbf{f}_{j_1},...,\mathbf{f}_{j_{D^{(\mathrm{II})}}}]^\top$. Vectors $\mathbf{f}_{j_d}$, for $d = 1,...,D^{(\mathrm{II})}$, are the linear discriminant "eigensequences" that correspond to the directions where the data vector projection yields high discrimination among the classes of interest.

We should note that the rank of $\mathbf{S}_B$ is at most $|\mathcal{C}| - 1$, hence we consider $D^{(\mathrm{II})} \leq |\mathcal{C}| - 1$. In addition, the rank of $\mathbf{S}_W$ cannot exceed $L - |\mathcal{C}|$, therefore insufficient training data is a potential problem. In our case, however, first, the input data dimensionality is significantly reduced by using Stage I of the proposed algorithm, and, second, the available training data are of the order $L = O(10^6)$. Therefore, in our experiments, $L - |\mathcal{C}| \gg d^{(\mathrm{II})}$ (see also section 3.1.6).

3.1.5 Stage III: Maximum Likelihood Data Rotation

In difficult classification problems such as large vocabulary continuous speech recognition, many high dimensional multi-variate normal densities are used to model the observation class conditional probability distribution. Due to lack of sufficient data, diagonal covariances are typically assumed, although the data class observation vector covariance matrices $\Sigma^{(\mathsf{c})}$, $\mathsf{c} \in \mathcal{C}$, are not diagonal. To alleviate this, we employ the maximum likelihood linear transform (MLLT) algorithm. MLLT provides a *non-singular* matrix $\mathbf{P}^{(\mathrm{III})}$ that "rotates" feature vector $\mathbf{x}_t^{(\mathrm{III})} = \mathbf{y}_t^{(\mathrm{II})}$, of dimension $d^{(\mathrm{III})} = D^{(\mathrm{II})}$, obtained by the first two stages of the proposed cascade algorithm as discussed in sections 3.1.3 and 3.1.4. The final feature vector is of length

$D^{(\mathrm{III})} = d^{(\mathrm{III})}$, and it is derived as $\mathbf{y}_t^{(\mathrm{III})} = \mathbf{P}^{(\mathrm{III})}\mathbf{x}_t^{(\mathrm{III})}$.

MLLT considers the observation data likelihood in the original feature space, under the assumption of diagonal data covariance in the transformed space. The desired matrix $\mathbf{P}^{(\mathrm{III})}$ is obtained by maximizing the *original* data likelihood, namely [42]

$$\mathbf{P}^{(\mathrm{III})} = \arg\max_{\mathbf{P}} \{ \mathsf{det}(\mathbf{P})^L \prod_{\mathsf{c}\in\mathcal{C}} (\mathsf{det}(\mathsf{diag}(\mathbf{P}\Sigma^{(\mathsf{c})}\mathbf{P}^\top)))^{-L_\mathsf{c}/2} \} ,$$

where $\mathsf{diag}(\bullet)$ denotes matrix *diagonal*. Differentiating the logarithm of the objective function with respect to $\mathbf{P}$ and setting it to zero, we obtain [42]

$$\sum_{\mathsf{c}\in\mathcal{C}} L_\mathsf{c} (\mathsf{diag}(\mathbf{P}^{(\mathrm{III})}\Sigma^{(\mathsf{c})}\mathbf{P}^{(\mathrm{III})\top}))^{-1}\mathbf{P}^{(\mathrm{III})}\Sigma^{(\mathsf{c})} = L(\mathbf{P}^{(\mathrm{III})\top})^{-1} .$$

The latter can be solved numerically [81].

3.1.6 Cascade Algorithm Implementation

Stage I (image transform) of the feature extraction algorithm is applied to each ROI vector $\mathbf{x}_t^{(\mathrm{I})}$ of length $d^{(\mathrm{I})} = 4096$ at the video rate of 60 Hz. To simplify subsequent LDA and MLLT training, as well as bimodal (audio-visual) fusion, we interpolate the resulting features $\mathbf{y}_t^{(\mathrm{I})}$ to the audio feature rate, 100 Hz. Furthermore, and in order to account for lighting and other variations, we apply *feature mean normalization* (FMN) by simply subtracting the feature mean computed over the entire utterance length T (cepstral mean subtraction), i.e., $\mathbf{y}_t^{(\mathrm{I})} \leftarrow \mathbf{y}_t^{(\mathrm{I})} - \Sigma_{t'=1}^{T}\mathbf{y}_{t'}^{(\mathrm{I})}/T$. This is akin to the audio front end processing [56, 82], and it is known to help visual speech recognition [78, 95].

At Stage II (LDA) and Stage III (MLLT), and in order to train matrices $\mathbf{P}^{(\mathrm{II})}$ and $\mathbf{P}^{(\mathrm{III})}$, respectively, we consider $|\mathcal{C}| \approx 3400$ context dependent sub-phonetic classes. We label vectors $\mathbf{x}_t^{(\mathrm{II})}$, $\mathbf{x}_t^{(\mathrm{III})}$, by forced alignment of the audio channel using an audio-only HMM. In the current front end implementation, we use $D^{(\mathrm{I})} = 24$, $D^{(\mathrm{II})} = D^{(\mathrm{III})} = 41$, and $J = 15$.

3.1.7 DCT-Feature Visual-Only Recognition Results

Based on the algorithm presented above, visual features have been extracted for the entire database preceding the workshop, and provided by the IBM participants. Using these features, visual-only HMMs were trained during the workshop as discussed in section 2.3, and subsequently used to rescore lattices "NLat" on the SI test set (see Tables 2.1 and 2.2).

Condition	WER (%)
Visual-only (with LM)	51.08
LM-only (no features)	58.31
Visual-only, with no LM	61.06
Random lattice path	78.14
Noisy audio-only	48.10

Table 3.1: "NLat" lattice rescoring results in WER (%), obtained with or without the use of visual-only trained HMM scores and language model (LM) scores. The baseline noisy audio-only performance is also depicted.

Recognition results are reported in Table 3.1. Recall that lattices "NLat" were obtained using noisy audio-only HMMs (section 2.2), therefore the *absolute* visual-only recognition results reported here are meaningless. Instead, these experiments were carried out to demonstrate that DCT features do provide useful speech information, and, in addition, to allow a preliminary comparison to the AAM features presented next. Indeed, as depicted in Table 3.1, the visual-only WER of 51.08% is significantly lower than the 58.31% WER of the best path through the "NLat" lattices using the language model information alone. Similarly, if we do not use any lattice language model information, the visual-only WER becomes 61.06%, which is much lower than the 78.14% WER of the random path through the "NLat" lattices, obtained when no HMM or language model scores are used. Clearly therefore, the DCT visual features do provide useful speech information.

3.2 Active Appearance Model Visual Features

An active appearance model (AAM) is a statistical model that combines shape and appearance information to derive a flexible model, coupled with an iterative scheme to fit the model to an example image. The AAM algorithm was first described by Cootes, et. al, in [19], and it was directly applied to tracking face images in [30]. In the lipreading context this approach may be viewed as a combination of both the low-level, data-driven approach with the high-level, model-based one, as mentioned in the introduction of this chapter.

Examples of using both appearance and shape to extract features for automatic recognition of visual speech include [16, 58, 59, 61]. However, never before have appearance and shape been combined in a *single* model. An active appearance model provides a framework to statistically combine both of these techniques. Building an AAM requires three applications

Figure 3.3: Example landmark points: Each of the 68 landmarks is hand-labeled on the training images. Red indicates a primary landmark and green a secondary one.

of principal component analysis (PCA):

- Shape eigenspace calculation to model shape deformations;
- Appearance eigenspace calculation to model appearance changes; and
- Using these, calculation of a combined shape and appearance eigenspace.

As the shape and appearance of, for example faces or lips, are often correlated, the aim of the final PCA is to remove this redundancy and create a single model that compactly describes shape and corresponding appearance deformation. Each of these steps is described in more detail in the following sections.

3.2.1 Shape Modeling

Shape deformations of the region being modeled (e.g., the face or lips) can be described compactly using the eigenspace of a set of *landmark* points [20]. In this implementation, landmark points are identified on the set of training images by hand. These points are chosen to approximate the shape of interest as a polygonal (dot-to-dot) model.

The number of landmarks used is a trade-off between the significant manual labor required annotating training images and the error in the polygonal approximation to a real, generally smooth shape. An example image is shown in Figure 3.3 with 68 landmark points labeled on the eyebrows, eye lids, nose bridge, under nose, lip inner and outer contour, and jaw line.

It is useful to introduce the concept of primary and secondary landmarks when manually labeling data. A primary landmark (shown in red) is one that should correspond to an easily identifiable image feature, such as the mouth corner. The secondary landmarks (shown in green) are equally spaced between primary landmarks to describe the shape. In this implementation, all landmarks are hand located, and all secondary landmarks are smoothed spatially along a spline. These can then be edited to accurately describe the shape and minimize the introduction of variance due to point mislocation along each curve.

The notion of primary and secondary landmarks exists only to aid the labeling process. For all video data processing, shape is described simply by the $(x\,, y)$ coordinates of all the landmark points. Any shape $\mathbf{s}$, is represented in two dimensions by the $2N$-dimensional vector of N concatenated coordinates

$$\mathbf{s} = [\, x_1, y_1, x_2, y_2 ,..., x_N, y_N \,]^\top .$$

Given a set of labeled landmark points, PCA [14,34] can be used to identify the optimal orthogonal linear transform (rotation of the axes) in terms of the variance described along each dimension. To identify only axes of genuine shape variation, each shape in the training set must be aligned. In this application, shapes are aligned using a similarity transform (translation, rotation and scaling). This is achieved using an iterative procrustes analysis [21, 26].

The main modes of shape variation, i.e., axes of greatest variance, are then found using PCA (discrete Karhunen-Loéve expansion). This simply requires computing the eigenvectors and eigenvalues of the covariance matrix[2] of the aligned shapes. Shape can then be modeled as a projection into this eigenspace,

$$\mathbf{s} = \overline{\mathbf{s}} + \mathbf{P}\,\mathbf{b}\ ,$$

where $\overline{\mathbf{s}}$ is the mean aligned shape, $\mathbf{P} = [\,\mathbf{p}_1, \mathbf{p}_2 ,..., \mathbf{p}_{2N}]$ is the matrix of eigenvectors, and $\mathbf{b}$ is the vector of corresponding weights for each eigenvector (the principal components).

The eigenvalues, λ_i, represent the variance accounted for by the corresponding ith eigenvector, $\mathbf{p}_i$. These allow sensible limits to be defined for each of the principle components. For example, they may be limited to lie within $\pm 3\sqrt{\lambda_i}$, to force points in the model to lie within three standard deviations of the mean.

If the eigenvectors are sorted in decreasing order according to the size of the correspond-

[2]Hence the use of the term eigen-X, where X is the application of your choice.

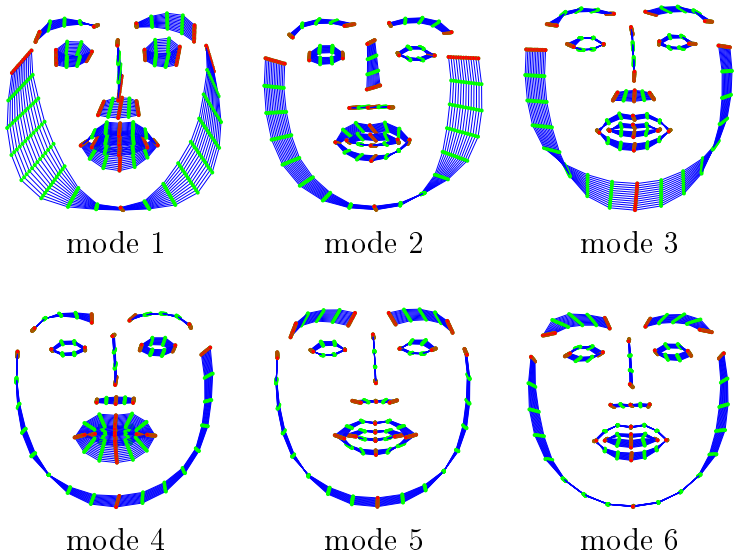


Figure 3.4: Statistical shape model. Each mode is plotted at ± 3 standard deviations around the mean. These six modes describe 74% of the variance of the training set.

ing eigenvalue, then the top t eigenvectors can be used to approximate the actual shape. Typically, t is chosen so that the sum of the top t eigenvectors describe, let's say, 95% of the total variance. This reduces the dimensionality significantly allowing valid shapes to be represented in a compact space

$$\mathbf{s} \approx \overline{\mathbf{s}} + \mathbf{P}_s \, \mathbf{b}_s \ ,$$

where $\mathbf{P}_s$ is the matrix of t shape eigenvectors $[\, \mathbf{p}_{s_1}, \mathbf{p}_{s_2}, ..., \mathbf{p}_{s_t} \,]$, and $\mathbf{b}_s$ is the t-dimensional vector of corresponding weights.

Figure 3.4 shows the mean face shape deformed by projecting up to ± 3 standard deviations for the first six modes. This model uses 11 modes to describe 85% of the variance of 4072 labeled images from the IBM ViaVoice$^{\text{TM}}$ audio-visual database.

3.2.2 Shape Free Appearance Modeling

Principal component analysis can be used in exactly the same manner as used in section 3.2.1 to compactly model pixel intensity, or color, variance over a training set of images. This application of PCA is often called "eigenfaces" [94]. Pixel values in an $N \times M$ image are represented as a single NM-dimensional vector by sampling the image from its rows or

columns, for example. For a greyscale image, its appearance $\mathbf{a}$ is

$$\mathbf{a} = [\, l_1, l_2, ..., l_{NM} \,]^\top,$$

where l_i is the ith luminance value in the image. The extension to a color image is simply to sample each color attribute for each pixel. For example, an RGB color image can be sampled to give the $3NM$-dimensional appearance vector

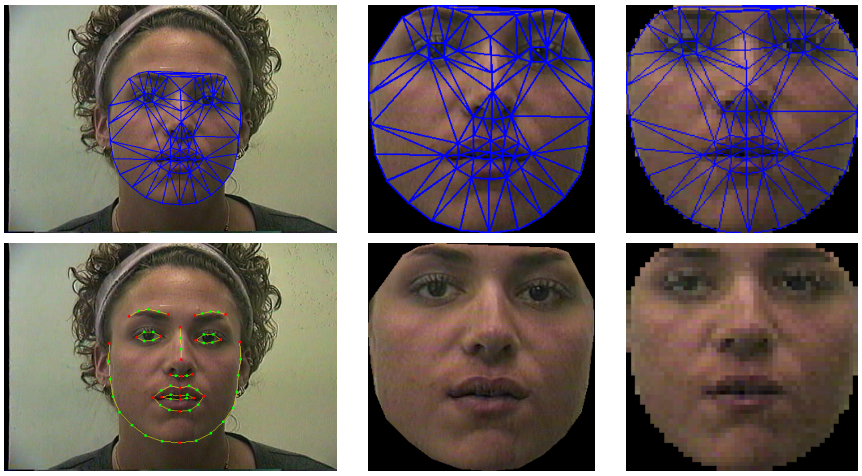
$$\mathbf{a} = [\, r_1, g_1, b_1, r_2, g_2, b_2, ..., r_{NM}, g_{NM}, b_{NM} \,]^\top.$$

A limitation of this approach to modeling appearance is that background pixels in the image can introduce significant variance. Typically, a region of interest (ROI) in the image is located to remove as much background as possible.

A more specific appearance model could be obtained by sampling only the pixel values that lie within the region to be modeled, for example the face. However, this would result in the appearance vector $\mathbf{a}$, that is likely to contain a different number of elements for each image. Simply resampling the modeled region to contain the same number of pixels is not sufficient. This would mean appearance elements in one image would not correspond to the same elements in another because of shape differences between the regions, which precludes the use of PCA.

One solution is to warp all training images to a reference shape before sampling only the ROI. The size of the reference shape can be chosen to define the number of appearance pixels to be modeled. This can easily be achieved by defining a warp using the landmark points labeled for shape modeling as source vertices, and the mean shape points $\overline{\mathbf{s}}$, as destination vertices. These vertices can be triangulated to form a mesh using, for example, a Delaunay triangulation [32]. The image then forms the texture map for a simple texture mapping operation that can be implemented using a graphics API such as OpenGL [102] and is often hardware accelerated.

Figure 3.5 illustrates this process. The landmark points are triangulated and the region covered by the resulting mesh is the ROI. Each triangle of the input mesh is warped to the destination triangle in the output mesh of the reference shape $\overline{\mathbf{s}}$. The reference shape could be arbitrary, it need not be the mean shape, but this is convenient. The reference shape has already been scaled so that the resulting *shape-normalized* image always contains 6000 pixels. The appearance can now be sampled in this reference frame where each pixel is approximately equivalent for all images. The use of texture mapping introduces discontinu-



labeled image face region warped image

Figure 3.5: Appearance normalization. The landmark points define the region of interest. They form the input vertices of a Delaunay triangulation for a texture mapping operation. The output vertices are the mean shape.

ities at each triangle boundary, but, in practice, this approximation to an ideal continuous warping function produces reasonable results very quickly.

A further post-processing step on the shape-normalized images is to normalize them all to have zero mean and unit variance. This removes the global lighting variation between images. PCA can now be used on the normalized appearances to identify the major modes of variation. Shape-normalized appearance is then approximated using the top t eigenvectors as

$$\mathbf{a} \approx \overline{\mathbf{a}} + \mathbf{P}_a \, \mathbf{b}_a \, ,$$

where $\mathbf{P}_a$ is the matrix of t shape normalized appearance eigenvectors $[\, \mathbf{p}_{a_1}, \mathbf{p}_{a_2}, ..., \mathbf{p}_{a_t} \,]$, and $\mathbf{b}_a$ is the t-dimensional vector of corresponding weights.

Figure 3.6 shows the mean shape-normalized appearance and projections at ± 3 standard deviations for the first six modes. This model uses 186 modes to describe 85% of the variance of the 4072 labeled training images from the IBM ViaVoice$^{\text{TM}}$ audio-visual database.

3.2.3 Combined Shape and Appearance Model

In many applications there will be significant correlation between shape and appearance. In the example of lips, the appearance looks different when the mouth is open as the oral

Figure 3.6: Shape free appearance. *Center row*: Mean appearance. *Top row*: Mean appearance +3 standard deviations ($+3\sigma$). *Bottom row*: Mean appearance -3 standard deviations (-3σ). The top six modes describe 41% of the training set variance.

cavity is seen (and possibly the teeth and tongue). A third PCA can be used to decorrelate the individual shape and shape-normalized appearance eigenspaces and create a combined shape and appearance model.

A combined shape and appearance space can be generated by concatenating the shape and appearance model parameters into a single vector

$$\mathbf{c} = [\, \mathbf{b}_s^\top, \mathbf{b}_a^\top \,]^\top.$$

As these models represent $(x\,, y)$ coordinates and pixel intensity values respectively, PCA cannot be applied directly on the combined vectors. This is due to the PCA *scaling problem* [14]. PCA identifies the axes of most variance, so if the data is measured in different units, then scaling differences between them will dominate the analysis, and any correlation between the variables will be lost. This can be compensated for, by introducing a weight to normalize the difference between the variance in shape and appearance parameters. The sum of the retained eigenvalues in the shape and appearance PCA calculation is the respective

variance described by each model, so the required weight can be calculated using

$$w = \sqrt{\frac{\sum_{i=1}^{t_a} \lambda_{a_i}}{\sum_{i=1}^{t_s} \lambda_{s_i}}} ,$$

where λ_{a_i} is the ith eigenvalue from the appearance PCA, λ_{s_i} is the ith eigenvalue from the shape PCA, and t_s and t_a are the number of retained eigenvectors in the shape and appearance PCA, respectively. A weight matrix to be applied to the shape parameters is then simply

$$\mathbf{W} = w\,\mathbf{I} ,$$

where $\mathbf{I}$ denotes the identity matrix.

For all examples in the training set, the labeled landmark points are projected into their shape parameters $\mathbf{b}_s$, and the appearance into appearance parameters $\mathbf{b}_a$. These are concatenated using the variance normalizing weight to form combined shape and appearance vectors

$$\mathbf{c} = [\,\mathbf{W}\,\mathbf{b}_s^\top, \mathbf{b}_a^\top\,]^\top .$$

Then, PCA is used to calculate the combined eigenspace

$$\mathbf{c} \approx \mathbf{P}_c\,\mathbf{b}_c ,$$

where $\mathbf{P}_c$ is the matrix of t shape and appearance eigenvectors $[\,\mathbf{p}_{c_1}, \mathbf{p}_{c_2}, ..., \mathbf{p}_{c_t}\,]$ and $\mathbf{b}_c$ is the t-dimensional vector of corresponding weights. There is no mean vector to add, as both $\mathbf{b}_s$ and $\mathbf{b}_a$ are zero mean. Again, t is chosen so the retained eigenvectors model the desired percentage of variance.

As the model is linear, shape and appearance can still be calculated from the combined shape and appearance model parameters

$$\mathbf{s} \approx \overline{\mathbf{s}} + \mathbf{P}_s \mathbf{W}^{-1} \mathbf{P}_{c_s} \mathbf{b}_c ,$$

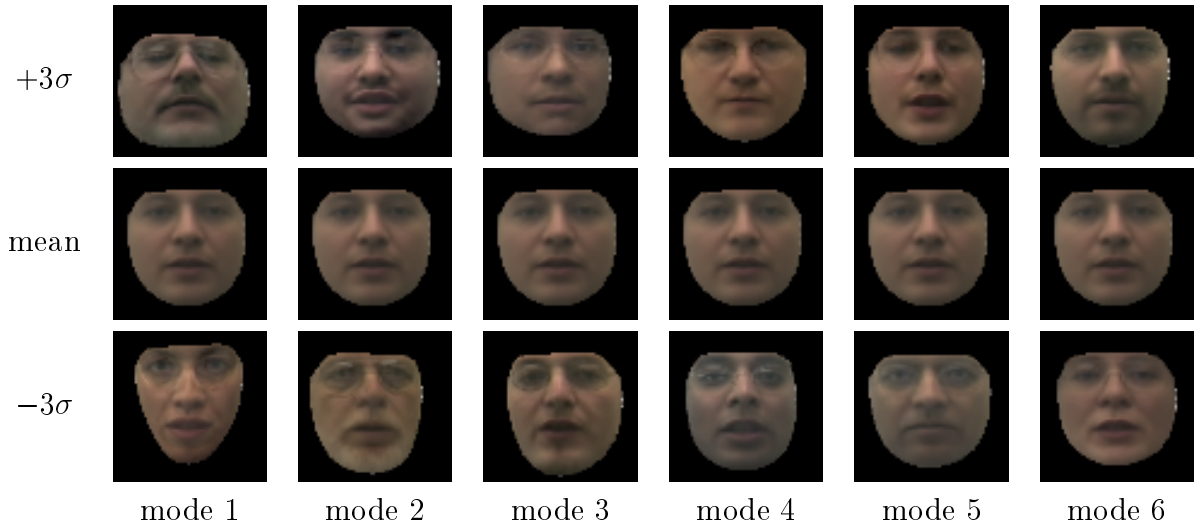


Figure 3.7: Combined shape and appearance. *Center row*: Mean shape and appearance. *Top row*: Mean shape and appearance +3 standard deviations. *Bottom row*: Mean shape and appearance −3 standard deviations. The top 6 modes describe 55% of the combined shape and appearance variance.

and

$$\mathbf{a} \approx \overline{\mathbf{a}} + \mathbf{P}_a \mathbf{P}_{c_a} \mathbf{b}_c \ ,$$

where

$$\mathbf{P}_c = [\, \mathbf{P}_{c_s}^\top, \mathbf{P}_{c_a}^\top \,]^\top \ .$$

Figure 3.7 shows the combined shape and appearance projections at ± 3 standard deviations for the first six modes. This model uses 86 modes to describe 95% of the variance of the 4072 IBM ViaVoice$^{\text{TM}}$ dataset training images.

3.2.4 Learning to Fit

A simple approach to fitting an AAM to a sample image is to use a numerical minimization algorithm, such as the *downhill simplex* [69], to iteratively minimize the fit error in terms of the model parameters. This approach works well for applications with low dimensionality [59] but the large number of iterations required imparts too great a performance penalty in the case of AAMs.

The AAM algorithm formulated by Cootes, et. al, in [19], assumes that, given small

perturbations from the actual fit of the model to a target image, a linear relationship exists between the difference in the model projection and image and the required updates to the model parameters. A similar approach was also used for model fitting by Sclaroff in [88].

All of the model parameters are grouped into a single vector with the pose values that define a similarity transform for projecting the model into the image

$$\mathbf{m} = [\, t_x, t_y, \theta, s, g_o, g_s, b_{c_1}, b_{c_2}, ..., b_{c_t}]^\top,$$

where t_x and t_y are translations in the x and y coordinates, respectively, θ is rotation, s is scale, g_o and g_s are global appearance offset and scaling terms to model changes in lighting conditions, and b_{c_i} is the ith combined shape and appearance model parameter.

If the linear assumption is valid, then small perturbations in the total model parameter set, denoted by $\delta\mathbf{m}$, have a linear relationship to the difference between the current model projection and the image, denoted by $\delta\mathbf{a} = \mathbf{a}_i - \mathbf{a}$, where $\mathbf{a}$ is the image appearance and $\mathbf{a}_i$ is the current model appearance. Clearly, to remove the effects of shape and pose, this difference must be calculated at some reference shape. The model shape-free appearance is calculated for a specific shape (generally the mean shape), so the image at the current model projection is warped back to the same shape to create the image appearance vector $\mathbf{a}_i$.

Given a training set of model perturbations $\delta\mathbf{m}$, and corresponding difference appearances $\delta\mathbf{a}$, the linear fit model

$$\delta\mathbf{m} = \mathbf{R}\,\delta\mathbf{a}\,,$$

can be solved for $\mathbf{R}$, using multiple linear regression. The training set can be synthesized to an arbitrary size using random perturbations of the model parameters and recording the resulting difference appearance.

The fitting algorithm is then a process of iterative refinement:

- Calculate the current difference image $\delta\mathbf{a}$, and current fit error $E_c = <\delta\mathbf{a}\,, \delta\mathbf{a}>$;
- Calculate the predicted update $\delta\mathbf{m} = \mathbf{R}\,\delta\mathbf{a}$;
- Apply a weighted predicted update $\mathbf{m}_p = \mathbf{m} - \alpha\,\delta\mathbf{m}$, where initially $\alpha = 1.0$;
- Calculate the predicted difference image $\delta\mathbf{a}_p$ and predicted fit error $E_p = <\delta\mathbf{a}_p\,, \delta\mathbf{a}_p>$;
- Iterate for values of $\alpha = 1.0\,, 0.5\,, ...$, until $E_p < E_c$, or the maximum number of iterations is reached.

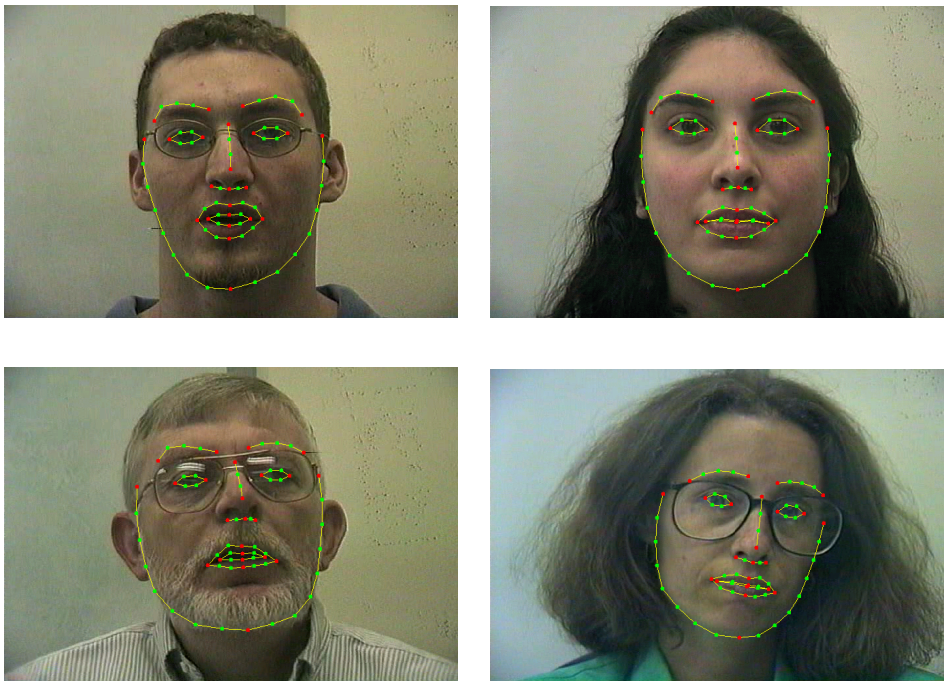


Figure 3.8: AAM landmark point training examples.

This sequence represents one AAM iteration, and it is repeated until there is no improvement in the current fit error E_c.

Given an initial model position reasonably close to the actual image, this algorithm can typically converge within a few iterations. As all calculations involving image data occur in the shape-free appearance reference shape, and at the scale defined by that shape, the entire fit process is independent of image size. However, the model pose transformation is directly related to the image size. For example, $t_x = 5$ is a translation of 5 pixels in the positive x direction. By subsampling the target image, and applying the suitable similarity transform to the current model pose parameters, the fit algorithm is extended to work at multiple resolutions. Starting at a coarse resolution, where a translation of 5 pixels is much more significant, the fit process is run to completion and the next highest resolution is chosen until a final fit is achieved on the original image. This multiresolution approach allows greater freedom in the choice of initial model parameters.

3.2.5 Training Data and Features

The training data consisted of landmark points hand located in 4,072 images taken from 323 sequences in the ViaVoice^{TM} audio-visual database. The model used 68 landmarks to

model the entire face region. This represents a significant amount of labor as each image can take several minutes to label. However, in total, this covers only 2 mins, 13 secs out of the approximately 50 hrs of the full database. Some example labeled images are shown in Figure 3.8.

This training data was used to build a point distribution model retaining 85% of the total shape variance, giving 11 modes of variation (see Figure 3.4). A shape-free appearance model was calculated using the mean shape as the reference shape, but scaled to contain 6000 pixels. This model required 186 modes to describe 85% of the shape-free appearance variance (see also Figure 3.6). These were combined to form the combined shape and appearance model by taking the 86 modes that described 95% of the concatenated shape and shape-free appearance model variance (Figure 3.7).

Features were extracted by applying the AAM fitting algorithm described in section 3.2.4 and recording the final model parameters. Model pose information (translation, rotation, scaling, and global appearance lighting transformation) was ignored as it is scene dependent. The 86-dimensional model parameter vectors were then either used directly as features, or further transformed using the methods described in sections 3.1.4 and 3.1.5.

Models were also built taking only the "beard" region of the face (the lower jaw and up to the bottom of the nose), or only the lip region. In both cases, poor tracking performance from the less detailed model prevented investigation of lipreading performance.

3.2.6 Tracking Results

The full face model was run on ViaVoice^{TM} image sequences over a period of five days at the workshop. Prior to this, all efforts were focused on increasing the amount of labeled training data[3] and increasing the AAM tracking speed. During this time, AAM tracking results were obtained for 4,952 sequences. This represents 1,119,256 image frames, or 10 hrs, 22 mins of video data at 30 frames per second. Average tracking speed was 4 frames per second.

One measure of how well the tracker was able to fit to an image sequence is to take the average over the sequence of the *mean square* pixel *error* (MSE) per image frame. The mean MSE over a sequence lies between 89.11 (for the best fitted sequence) and 548.87 (for the worst one). Example frames from each of these sequences are shown in Figure 3.9. Over all of the tracked data, the average sequence MSE was 254.21. Visual analysis of a sample of the tracking results suggests that, in many cases, the AAM tracker failed to follow small

[3]Many thanks for the efforts of Laurel Phillips at Carnegie Mellon University, and June Sison and Azad Mashari at the workshop.

(a) Example frame from a good fit.

(b) Example frame from a bad fit.

Figure 3.9: AAM tracking result examples. A well fitted frame is shown in (a) and a poorly fitted frame in (b), alongside the original image.

facial motions. In practice, the tracker was more effective at locating the face region than accurately modeling facial expression. Given the small size of the AAM training set, this is perhaps to be expected.

3.2.7 AAM-Feature Visual-Only Recognition Results

Taking the same approach described in section 3.1.7, visual-only HMMs were trained using variations of the AAM features. As AAM tracking results were not available for the full ViaVoice^{TM} database, the AAM features were split into training and test sets that are respectively subsets of the multi-speaker training and test sets described in section 2.1 (see Table 2.1). Unfortunately, this means the AAM results cannot be directly compared to the DCT results in section 3.1.7 (Table 3.1). However, equivalent DCT results were obtained on the same subset used to obtain the AAM results.

Feature set	WER (%)
AAM: 86-dim	65.69
AAM: 30-dim	65.66
AAM: 30-dim + Δ + $\Delta\Delta$ (90-dim)	65.90
AAM: 86-dim + LDA (24-dim) + LDA over 15 frames + MLLT (41-dim)	64.00
DCT: 18-dim + Δ + $\Delta\Delta$ (54-dim)	61.80
DCT: 24-dim + LDA over 15 frames + MLLT (41-dim)	58.14
Noise: 30-dim	61.37

Table 3.2: "NLat" lattice rescoring results on a subset of the SI test set, expressed in WER (%), obtained with visual-only HMMs trained on various visual feature sets.

The rescoring results are summarized in Table 3.2. All results are depicted in percentage word error rate (WER).[4] The top row is the result using all 86 AAM features. The second row is the result using only the top 30 of the 86 AAM features. The third row is obtained by appending first and second derivatives (denoted by Δ and $\Delta\Delta$, respectively) to these 30. The fourth row is the AAM result obtained after using an LDA feature projection to a 24-dimensional space, followed by the LDA/MLLT projection described in section 3.1. The fifth row is the result using DCT features with their first and second derivatives appended, and the sixth row is the result using the LDA/MLLT transformed DCT features. Finally, the bottom row is the result obtained by training models on 30-dimensional uniform random noise features.

It is interesting to note that the only features that give lower word error rate than the random noise features are the LDA/MLLT transformed DCT features. All of the AAM feature variants performed worse than the random noise features, which are effectively exploiting information in the language model combined with the restricted depth lattices.

3.3 Summary

In this chapter, we presented two visual front ends for automatic speechreading, namely features based on the DCT of an appropriately tracked mouth ROI, discussed in section 3.1, and features based on a joint shape and appearance model of the face ROI, by means of AAMs, presented in section 3.2. Both feature sets can be further transformed by using LDA

[4] As mentioned in section 3.1.7, these results cannot be interpreted as visual-only recognition, due to the rescoring of the noisy audio-only lattices.

and MLLT, discussed in section 3.1.

Noisy audio lattice rescoring experiments show that using AAM features results in worse recognition performance than simply using uniform random noise as visual features. The AAM features also perform worse than DCT features on the same subset of the ViaVoice^{TM} dataset. Therefore, the DCT based visual feature representation discussed in section 3.1 is exclusively used in all experiments reported in the following chapters.

There are two reasons for the poor AAM performance: Modeling errors, and tracking errors. The first may be due to a poor choice of model or insufficient training data to generalize the model to the test data. The second may also be due to insufficient training data as the AAM algorithm also uses this to learn how to fit.

The poor recognition performance is related to the significant number of poorly tracked sequences. The tracking algorithm used does not update model parameters if no better fit is found between successive images. This introduces sections where the features remain constant over many frames. As a direct transformation of the image, the DCT method always gives a dynamic feature, even if the face tracking has failed on a given frame.

Given the small amount of labeled AAM training data it may not be surprising that the resulting model is unable to capture all facial changes during speech. Only snap-shots of speech were modeled and this does not appear to be enough to generalize to continuous visual speech. Note also that the tedious task of hand labeling training data images is a significant drawback to the AAM approach.

Chapter 4

Visual Clustering and Adaptation

A prominent aspect in any large vocabulary continuous speech recognition (LVCSR) HMM based system is modeling *context dependence* of speech units (phone models) [82, 103], in order to reliably capture co-articulation. To avoid data sparseness due to the large number of such context dependent phone units, *decision trees* are typically used to cluster them. In the case of visual speech, it is well known that not all phones are visually distinct, but rather they cluster in so-called *visemes* [45,62]. It is then of interest to investigate whether context clustering on basis of visemic instead of phonetic context is advantageous to visual-only ASR. The issue was addressed in the summer workshop and is reported in section 4.1.

The second aspect of visual modeling considered in this chapter is the question of visual-only HMM *adaptation* to unseen subjects. Although this has been studied before for small vocabulary tasks [79], our workshop experiments constitute the first time that adaptation techniques have been investigated in the visual-only LVCSR domain. Visual-only speaker adaptation is discussed in section 4.2.

4.1 Visual Clustering

As stated above, current state-of-the-art LVCSR HMM based speech recognizers use context dependent phones as speech units. In HTK [103], context dependence is modeled by triphone units. Rather than considering all possible triphones, only triphone contexts that are substantially different are chosen. For every phone state, decision tree based clustering is used to group contexts that are similar. At each node in the decision tree, the data is split into two classes by means of *questions* that ask whether the phone to the right or left of the current phone belongs to a group of phones which are similar along some dimension (e.g.,

Silence	/sil/, /sp/
Lip-rounding based vowels	/ao/, /ah/, /aa/, /er/, /oy/, /aw/, /hh/ /uw/, /uh/, /ow/ /ae/, /eh/, /ey/, /ay/ /ih/, /iy/, /ax/
Alveolar-semivowels	/l/, /el/, /r/, /y/
Alveolar-fricatives	/s/, /z/
Alveolar	/t/, /d/, /n/, /en/
Palato-alveolar	/sh/, /zh/, /ch/, /jh/
Bilabial	/p/, /b/, /m/
Dental	/th/, /dh/
Labio-dental	/f/, /v/
Velar	/ng/, /k/, /g/, /w/

Table 4.1: The 13 visemes considered in this work.

acoustic similarity). Such a question is referred to as a context question.

In the workshop we investigated the design of context questions that are based on visual similarity. Specifically, we first defined thirteen visemes, i.e., visually similar phone groupings (see section 4.1.1). Visual context questions based on these visemes were subsequently developed to guide binary tree partitioning during triphone state clustering (section 4.1.2). The resulting phone trees were inspected in order to observe the importance of visual context questions and possibly reveal similar linguistic contextual behavior between phones that belong in the same viseme (section 4.1.3). Finally, visual-only HMMs were trained based on the resulting context trees, and they were compared to ones trained using decision tree clustering on basis of acoustic phonetic only questions (section 4.1.4).

4.1.1 Viseme Classes

Not all phones are visually distinct. However, they can be clustered in visemes, which differ in the *place* of articulation, and, therefore, can be visually distinguished [45, 62]. In the summer workshop, we determined seven consonant visemes, namely the bilabial, labio-dental, dental, palato-alveolar, palatal, velar, and two alveolar visemes. For example, {/p/, /b/, /m/} constitutes the well-known bilabial viseme [45]. Lip rounding during formation of vowels defined the remaining four vowel visemes and an alveolar-semivowel one, whereas one viseme was devoted to the two HTK silence phones, i.e., {/sil/, /sp/}. The thirteen resulting visemes are depicted in Table 4.1. These were subsequently used to guide development of

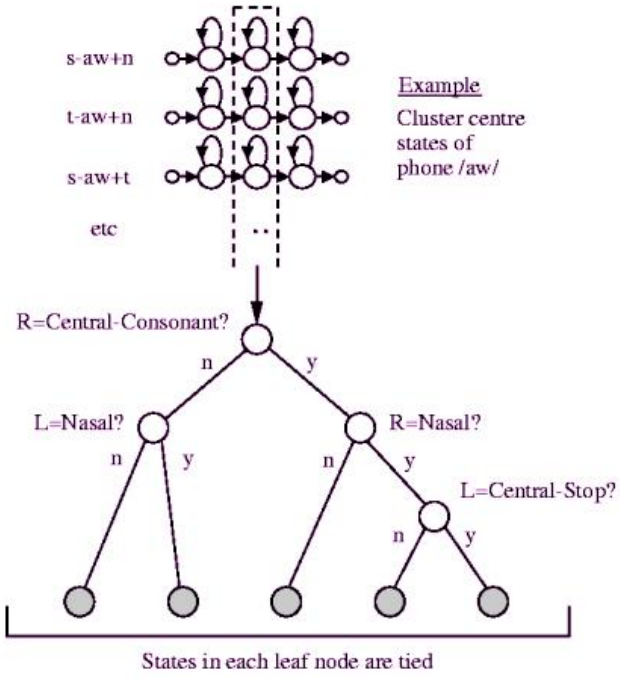


Figure 4.1: Decision tree based HMM state clustering (Figure 10.3 of [103]).

visual context questions needed for decision tree based triphone state clustering, as described next.

4.1.2 Visual Context Questions

Triphone HMM states were clustered using binary trees as depicted in Figure 4.1. Tree partitioning employed questions about the left and right contexts of all triphone states.

We added 76 visual context questions to an existing standard audio context question set. This original set of questions was composed of 116 acoustic context questions and 84 questions that defined the context as particular single phones. Aside from the inclusion of all original acoustic context questions, these same audio context questions (characterized by *manner* of articulation) were split into more specific groupings restricted by place of articulation. For example, a single acoustic context question based on non-nasal stops {/p/, /b/, /t/, /d/} resulted in two separate visual context questions based on its bilabial and alveolar viseme members, {/p/, /b/} and {/t/, /d/}, respectively. The final joint question set consisted of a total of 276 questions.

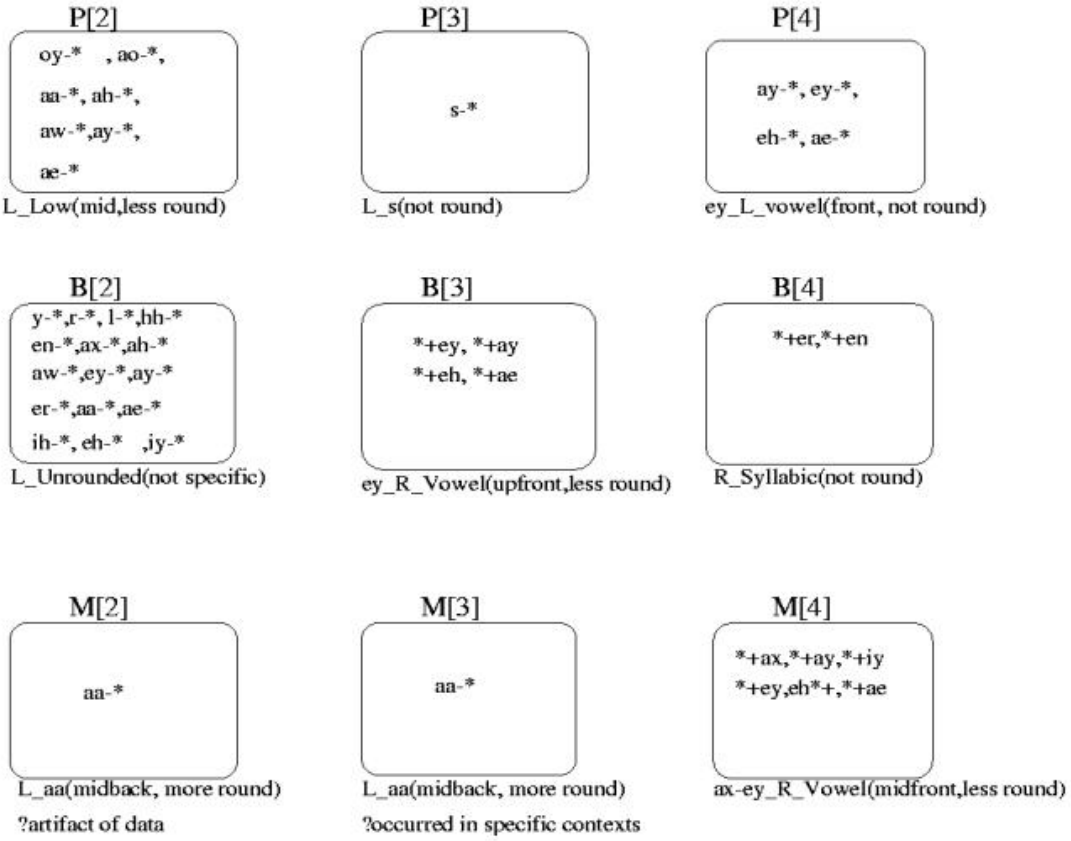


Figure 4.2: Decision tree root questions for the three emitting states (states 2, 3, and 4) of the HMMs for phones /p/, /b/, and /m/, that make up the bilabial viseme.

4.1.3 Phone Tree Root Node Inspection

As a first investigation of the relative influence of the visual context questions in the decision tree design, we used HTK to design 123 phone state clustering decision trees (one for each of the three states of the 41 monophone HMMs, excluding /el/, /sil/, and /sp/). The trees were constructed based on the DCT visual features, appropriately trained context dependent visual HMMs (see section 2.3 and Figure 2.3), and the 276 questions described above. We subsequently analyzed the root nodes of the resulting decision trees. Root node analysis involved simple frequency counts of the number of audio, visual, and single phone context questions that determined partitions at the root of the 123 phone state trees.

This inspection revealed that visual context questions were employed in about one third of the root nodes. Specifically, out of the 123 decision trees, 33 had visual context questions

Dec. Tree	WER (%)
"AA"	51.24
"VA"	51.08
"VV"	51.16

Table 4.2: Visual-only HMM recognition performance based on three different decision trees.

at their root node, 74 had audio context root node partitions, and the remaining 16 had root node partitions obtained by single phone context questions. Clearly, visual context questions played an important role in the decision tree based triphone state clustering.

Further inspection of the decision trees, however, did not reveal similar linguistic contextual behavior between phone trees within the same viseme class. Rather, the results appeared unbalanced and any pattern seemed to be an artifact of the specific data corpus and not driven by linguistic rules (see also Figure 4.2).

4.1.4 Visual Clustering Experiments

In order to test whether decision tree clustering by means of visual context questions improves performance of the resulting visual-only HMMs, a number of such models were trained using the DCT visual features described in section 3.1. We trained three visual-only HMMs on the training set depicted in Table 2.1. Similarly to the experiments reported in chapter 3, all HMMs were used to rescore the "NLat" lattices on the SI test set (see also Tables 2.1 and 2.2).

Two sets of questions were used for the decision tree clustering. The original audio context questions consisted of 200 questions that group phones primarily based on *voicing* and manner of articulation. These questions were also used to train the audio-only baseline HMMs in section 2.3. The second set of questions considered consisted of the 276 questions described in section 4.1.2, namely the 200 audio context questions augmented by the 76 visual context questions. With some abuse of notation, we refer to the two sets as the "audio" and "visual" questions, respectively.

The minimum likelihood gain threshold and the minimum cluster size used in decision tree development were set to the same values used in the audio-only HMM decision tree design (300 and 250, respectively). Such values were likely suboptimal for the visual models, as they have been optimized for audio features. It would be of interest to obtain optimal choices for these values in the visual feature case.

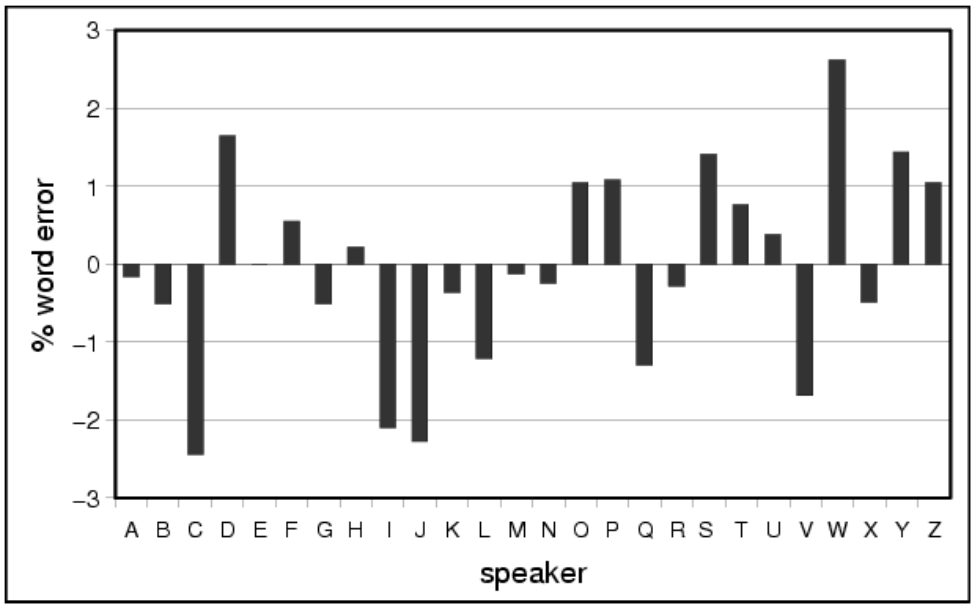


Figure 4.3: Absolute recognition performance difference between the "VA" and "VV" clustered visual-only HMMs, expressed in WER (%), for each of the 26 test subjects. Positive values indicate subjects where the "VV" system is superior.

The three visual-only context dependent HMMs were trained based on clustering by means of three different decision trees. These trees were obtained using various front ends (features) and questions, and are denoted as follows:

- *AA*: Uses audio features and "audio" questions (i.e., the decision tree is identical to the one used for audio-only HMM training);
- *VA*: Uses visual features, but "audio" questions;
- *VV*: Uses visual features and "visual" questions.

The performance of the resulting visual-only HMMs trained using the "AA", "VA", and "VV" decision trees is depicted in Table 4.2, expressed in WER (%). Clearly, there was no significant difference in the performance of the three models. The "AA" based system performed somewhat worse than the other two models, whereas, surprisingly, the "VA" was the best.

We further investigated the "VA" and "VV" system differences on a per subject basis, for each of the 26 subjects of the SI test set. The results are depicted in Figure 4.3. Notice, that although there were not significant overall differences resulting from incorporating the new set of questions in decision tree design, for particular individuals, absolute WER differences were

almost as great as 3%. It is worth mentioning, that visual-only recognition results by both the "VA" and "VV" systems followed the noisy audio-only HMM recognition performance, per subject. This was an artifact of the "NLat" lattice rescoring experiments, which severely restricted decoding (see also sections 3.1.7 and 3.2.7).

We also performed an analysis of how many times each question was used in the "VV" decision trees. This revealed that the 76 introduced viseme based questions were used quite frequently: Within the top 20 questions used in the "VV" tree, 11 were viseme based, thus affecting the relative frequency with which the traditional audio questions were used. Some such audio questions that did not rank high in the "VA" decision tree, were used further up the trees in the "VV" based clustering.

It is also worth noticing that all three decision trees ("AA", "VA", and "VA") formed approximately 7000 clusters. However, the set of visually distinguishable classes (visemes) is much smaller than the number of phones, thus we considered it of interest to investigate smaller "VV" decision tree sizes. We constructed such a decision tree (of the "VV" type) with about 2500 clusters, by increasing the minimum likelihood gain threshold to 900. However, this resulted to some performance degradation of the corresponding visual-only HMM.

These results indicate that viseme based context questions for decision tree based clustering do not appear to improve system performance. We view, however, these experiments to be a first only investigation of visual model clustering. Further work is merited in this area, including full decoding experiments, improvements in the decision tree clustering algorithm, and a possible redesign of visual context questions.

4.2 Visual Model Adaptation

Every subject has unique speech characteristics, and, in the case of visual speech, also unique visual appearances. It is therefore expected that a speaker independent (SI) trained audio-visual ASR system is not necessarily sufficient to accurately model each new subject, or even specific enough to each subject in its training population.

In practice, it is often the case that a small training data amount of a previously unseen (in training) subject becomes available. This scenario corresponds, for example, to subject enrollment in commercial LVCSR dictation systems. Such data is typically not sufficient to train a subject specific HMM recognizer, however it can be used to transform SI HMM parameters to obtain a *speaker adapted* (SA) HMM, capable of capturing the subject speech characteristics better. Speaker adaptation algorithms have been successfully used for this

task in traditional audio-only ASR [37, 54, 72]. Such common algorithms include the *maximum likelihood linear regression* (MLLR) [54], *maximum-a-posteriori* (MAP) [37] adaptation, and methods that combine both [72]. The first is especially useful when the adaptation data is of very small duration (*rapid* adaptation).

In contrast to audio-only HMM adaptation, in the visual-only and audio-visual ASR domains, speaker adaptation has only been considered for small vocabulary tasks [79]. In this section, we investigate the use of MLLR to visual-only adaptation in the LVCSR domain.

4.2.1 MLLR Visual-Only HMM Adaptation

Speaker adaptation by means of MLLR transforms only the means of the Gaussian mixtures that model the class conditional HMM observation probabilities. Let us denote such means by $\mathbf{m}_{\mathsf{c}j}$, where c denotes any HMM class (context dependent state cluster), and j any mixture component for this class (see also (5.2)). Let also $\mathcal{P}$ denote a partitioning of the set of all Gaussian mixture components, obtained by K-means clustering, for example [82, 103], and let $p \in \mathcal{P}$ denote any member of this partition. Then, for each mixture cluster, MLLR seeks a transformation matrix $\mathbf{W}_p$, that linearly transforms the SI mixture means of the cluster to obtain SA means, by

$$\mathbf{m}_{\mathsf{c}j}^{(\mathrm{SA})} = \mathbf{W}_p [\, 1 \,,\, \mathbf{m}_{\mathsf{c}j}^{\top}]^{\top} \,, \quad \text{where} \quad (\mathsf{c} \,, j) \;\in\; p \,, \tag{4.1}$$

to maximize the adaptation data likelihood. In (4.1), matrices $\mathbf{W}_p$ are of size $D \times (D+1)$, where D is the mean vector dimension. Hence, MLLR also adds a bias term to the SI Gaussian means [54]. To avoid overtraining, matrices $\mathbf{W}_p$ are often block-diagonal.

4.2.2 Adaptation Results

We have conducted *supervised* visual-only HMM adaptation experiments using part of the SI adaptation set of our audio-visual database (see Table 2.1). For simplicity, we have considered only 10 of the 26 subjects of this set, and used an average of 5 minutes of data per subject to create a SA visual-only HMM for each. The performance of the SA HMM was evaluated on the SI test set that corresponds to the specific subject, and compared to the performance of the SI HMM system. To simplify experiments, all context dependent HMMs had a single Gaussian observation probability. In addition, a single, full-matrix MLLR transform was used in each of the 10 speaker adaptation experiments. The obtained results are depicted in Table 4.3.

Subject	SI	SA
AXK	44.05	41.92
BAE	36.81	36.17
CNM	84.73	83.89
DJF	71.96	71.15
JFM	61.41	59.23
JXC	62.28	60.48
LCY	31.23	29.32
MBG	83.73	83.56
MDP	30.16	29.89
RTG	57.44	55.73

Table 4.3: Visual-only HMM adaptation experiments using MLLR: Speaker-independent (SI) and speaker-adapted (SA) visual-only HMM performance is reported in WER (%), per subject, obtained by rescoring "NLat" lattices.

It is clear from Table 4.3 that adaptation consistently improved visual-only HMM performance for all subjects. For several individual subjects (e.g., AXK, JXC, JFM, LCY, and RTG), we actually observed significant improvements. These results could likely be further improved by using multiple block-diagonal MLLR transformation matrices, and possibly by applying MAP adaptation, following MLLR [72].

4.3 Conclusions

As pointed out above, modeling context dependence is a key element of the progress that has been made in audio-based speech recognition. Most of the speech community has converged on using triphone contexts, while others (including IBM) use pentaphone contexts. In both cases, it is essential to discover the most meaningful contexts. This is often done by automatically grouping (using decision trees, for instance) phonetic contexts that are similar along some acoustic dimension. Acoustic similarities however are not necessarily the most appropriate for training visual-only systems. So, in this chapter, we explored ways to develop visually meaningful phone groupings (based on the place of articulation), and we designed a set of decision tree questions to develop viseme based triphone contexts. Analysis of the resulting decision trees indicated that questions about visually relevant groupings do get used at high levels in the decision trees. However, preliminary experiments using visually clustered HMMs did not show any improvements relative to the baseline acoustically clus-

tered HMM system. This is a somewhat surprising result. However, we believe that more work is needed in this direction, before we can draw any conclusions. In particular, we did not adequately optimize the parameters that guide the process of developing decision tree triphone clusters. Also, we used the visual questions as a complement to the acoustic questions. Instead, it may have been more appropriate to use the visual questions by themselves. In this chapter, we also considered visual-only HMM supervised adaptation in the LVCSR domain to new subjects. A simple implementation of the MLLR adaptation algorithm in this domain showed some expected, but small, improvements.

Chapter 5

Models for Audio-Visual Fusion

The main concentration of our workshop team was on the audio-visual integration problem. Our aim was to investigate and propose algorithms for the automatic recognition of audio-visual speech within the traditional HMM based speech classification framework, hoping to obtain significant performance gains over audio-only recognition for both clean and noisy audio conditions considered.

Audio-visual fusion is an instance of the general classifier combination problem [47]. In our case, two observation streams are available (audio and visual modalities) and provide information about hidden class labels, such as HMM states, or, at a higher level, word sequences. Each observation stream can be used alone to train single-modality statistical classifiers to recognize such classes. However, one hopes that combining the two streams will give rise to a bimodal classifier with superior performance to both single-modality ones.

A number of techniques have been considered in this workshop for audio-visual information fusion, which can be broadly grouped into *feature fusion* and *decision fusion* methods. The first ones are the simplest, as they are based on training a traditional HMM classifier on the concatenated vector of the audio and visual features, or any appropriate transformation of it. This is feasible, as both audio and video streams provide time synchronous features (see also section 3.1.6). Feature fusion is presented in section 5.1.

The remaining sections in this chapter are devoted to decision fusion methods. Such techniques combine the single-modality (audio- and visual-only) HMM classifier outputs to recognize audio-visual speech. Specifically, class conditional log-likelihoods from the two classifiers are linearly combined using appropriate weights that capture the reliability of each classifier, or data stream [47]. This likelihood recombination can occur at various levels of integration, such as the state, phone, syllable, word, or utterance level. In the summer

workshop we explored three such levels of integration:

- *State* level combination, which gives rise to the *multi-stream* HMM, and it is discussed in section 5.2;
- *Phone* level combination, which extends the multi-stream HMM to the *product*, or *composite*, HMM, discussed in section 5.3; and
- *Utterance* level combination, which is based on a *discriminative model combination* approach and rescoring of n-best hypotheses (see section 5.5).

In all cases, estimation of appropriate log-likelihood combination weights is of paramount importance to the resulting model performance. Weight estimation for multi-stream and product HMMs is discussed in section 5.4, and for the discriminative model combination approach in section 5.5. A summary of the best audio-visual fusion results is given in section 5.6.

5.1 Feature Fusion

In the summer workshop, we have considered two feature fusion techniques, which are schematically depicted in Figure 5.1. The first method uses the traditional concatenation of the synchronous audio and visual features as the joint audio-visual feature vector, on the basis of which an HMM based recognition system is trained. The second method seeks to reduce the size of the concatenated audio-visual feature vector, before training HMMs on it. This is achieved by projecting it to a lower dimensional space by means of LDA, followed by an MLLT (see also sections 3.1.4 and 3.1.5). As LDA has already been applied to obtain both audio- and visual-only feature vectors separately, the proposed additional projection amounts to its second application. Therefore, this novel fusion method is named *hierarchical* LDA (HiLDA). It is worth pointing out that, unlike decision fusion techniques, neither feature fusion algorithm makes any conditional independence assumption between the two modalities.

5.1.1 Concatenative Feature Fusion

Let us denote the time synchronous audio- and visual-only feature vectors (observations) at instant t, by $\mathbf{o}_s^{(t)} \in \mathsf{R}^{D_s}$, of dimension D_s, where $s = \mathrm{A}, \mathrm{V}$, respectively. The joint audio-visual

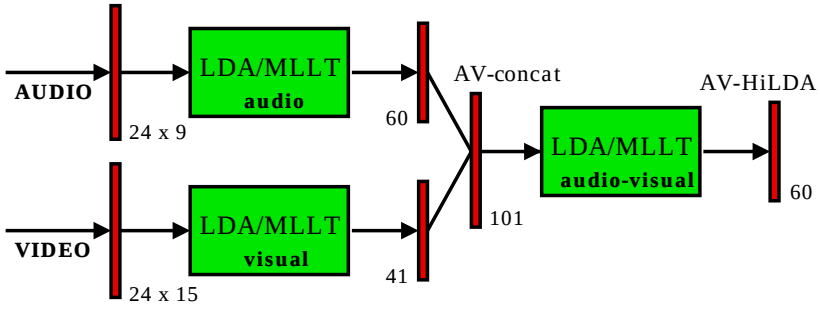


Figure 5.1: Two types of feature fusion considered in this section: Plain audio-visual feature concatenation (AV-concat) and hierarchical LDA / MLLT feature extraction (AV-HiLDA). Feature vector dimensions are also depicted.

feature vector is then simply the concatenation of the two, namely

$$\mathbf{o}^{(t)} = [\, \mathbf{o}_{\mathrm{A}}^{(t)\top}, \mathbf{o}_{\mathrm{V}}^{(t)\top} \,]^{\top} \in \mathsf{R}^{D} \,, \tag{5.1}$$

where $D = D_{\mathrm{A}} + D_{\mathrm{V}}$.

We model the generation process of a sequence of such features, $\mathbf{O} = [\, \mathbf{o}^{(1)}, \mathbf{o}^{(2)}, ..., \mathbf{o}^{(T)} \,]$, by a traditional, *single-stream* HMM, with *emission* (class conditional observation) probabilities, given by

$$Pr\,[\, \mathbf{o}^{(t)} | \, \mathsf{c} \,] = \sum_{j=1}^{J_{\mathsf{c}}} w_{\mathsf{c}j}\, \mathcal{N}_D\,(\, \mathbf{o}^{(t)}; \mathbf{m}_{\mathsf{c}j}\,, \mathbf{s}_{\mathsf{c}j}\,) \,. \tag{5.2}$$

In (5.2), $\mathsf{c} \in \mathcal{C}$ denote the HMM context dependent states (classes). In addition, mixture weights $w_{\mathsf{c}j}$ are positive adding up to one, J_{c} denotes the number of mixtures, and $\mathcal{N}_d\,(\mathbf{o}\,;\mathbf{m}\,,\mathbf{s})$ is the d-variate normal distribution with mean $\mathbf{m}$ and a diagonal covariance matrix, its diagonal being denoted by $\mathbf{s}$.

As depicted in Figure 5.1, in our experiments, the concatenated audio-visual observation vector (5.1) is of dimension 101. This is rather high, compared to the audio- and visual-only feature sizes, and can cause inadequate modeling in (5.2) due to the curse of dimensionality. To avoid this, we seek lower dimensional representations of (5.1), next.

5.1.2 Hierarchical Fusion Using Feature Space Transformations

In general, the visual features currently used contain less speech classification power than audio features, even in the case of extreme noise in the audio channel (see also Table 3.1). One would therefore expect that a lower-dimensional representation of (5.1) could lead to equally, or even better, HMM performance, given the problem of accurate probabilistic modeling in high-dimensional spaces.

It makes a reasonable choice to consider LDA as a means of obtaining such a dimensionality reduction. Indeed, our aim is to obtain the best discrimination among the classes of interest, and LDA achieves this on basis of the data (and their labels) alone, without any a-priori bias in favor of any of the two feature streams. Similarly to section 3.1, LDA is followed by an MLLT based data rotation, in order to improve maximum-likelihood data modeling using (5.2). The proposed method amounts to a hierarchical LDA / MLLT application on the original audio and visual DCT features, as depicted in Figure 5.1, and it is therefore referred to as HiLDA (hierarchical LDA).

The final audio-visual feature vector is (see also (5.1))

$$\mathbf{o}_{\text{HiLDA}}^{(t)} = \mathbf{P}_{\text{MLLT}}\,\mathbf{P}_{\text{LDA}}\,\mathbf{o}^{(t)} \ .$$

Matrices $\mathbf{P}_{\text{LDA}}$ and $\mathbf{P}_{\text{MLLT}}$ denote the LDA projection and MLLT rotation matrices. In our experiments, their dimensions are 60×101 and 60×60, respectively: We have chosen to obtain a final audio-visual feature vector of the same size as the audio-only one, in order to avoid high-dimensionality modeling problems.

5.1.3 Feature Fusion Results

At the workshop, we trained audio-visual HMMs using the two feature fusion techniques discussed above for both the clean and noisy-audio cases. The training procedure for these HMMs is outlined in section 2.3.

Subsequently, we first used the trained HMMs in clean audio to rescore the "Lat" lattices (see Table 2.2). The results are reported in Table 5.1. When using the concatenated features, we observed some performance degradation with respect to the baseline clean audio-only WER. Using, however, the HiLDA feature fusion resulted in a slight improvement over the baseline (about a 4% WER relative reduction). Since we rescored lattices that were generated on basis of audio-only information, we believe that full decoding with the HiLDA technique could have resulted in additional gains.

Audio Condition:	Clean	Noisy	
Rescored Lattices:	"Lat"	"NLat"	"NAVLat"
Audio-only	14.44	48.10	–
AV-concat	16.00	44.97	40.00
AV-HiLDA	13.84	42.86	36.99

Table 5.1: Audio-visual feature fusion performance on the SI test set using concatenated (AV-concat) and hierarchical LDA (AV-HiLDA) audio-visual features: Clean and noisy audio conditions are considered. Both "NLat" and "NAVLat" lattices are rescored in the noisy audio case fusion. All results are in WER (%).

In the noisy audio case, we first rescored "NLat" lattices, generated by the IBM system on basis of noisy audio-only observations and a matched-trained HMM. Both feature fusion techniques resulted in substantial gains over the noisy audio-only baseline performance, with HiLDA being again the best method. As discussed in section 2.2, the "NLat" lattices contain audio-only information, that, in the noisy audio case, is very unreliable. It is therefore more appropriate to rescore lattices that contain audio-visual information. Such are the "NAVLat" lattices, generated by training an HMM on HiLDA audio-visual features, in the noisy audio case. As expected, the results improved significantly. The HiLDA algorithm yielded a 36.99% WER, compared to the baseline noisy audio-only 48.10% WER. This amounts to a 24% WER relative reduction. Notice that "NAVLat" lattice rescoring provides the fair result to report for the HiLDA technique. However, the concatenative feature fusion result is "boosted" by its superior HiLDA-obtained "NAVLat" lattices. Its actual, free decoding performance is expected to be somewhat worse than the 40.00% WER (but better than the 44.97% WER), reported in Table 5.1. In the remaining (decision) fusion experiments, "NAVLat" lattices were exclusively used in the noisy audio case.

It is of course not surprising that HiLDA outperformed plain feature concatenation. In our implementation, concatenated audio-visual features, were of dimension 101, which is rather high, compared to audio-only and HiLDA features, that were both of dimension 60. HiLDA uses a discriminative feature projection to efficiently "compact" the concatenated audio-visual features. The curse of dimensionality and undertraining are possibly also to blame for the performance degradation compared to the clean audio-only system, when plain audio-visual feature concatenation is used.

5.2 State Synchronous Decision Fusion

Although feature fusion by means of HiLDA results in improved ASR over audio-only recognition, it does not explicitly model the reliability of each modality. Such modeling is very important, as speech information content and discrimination power of the audio and visual streams can vary widely, and at a local level, depending on the spoken utterance, acoustic noise in the environment, visual channel degradations, face tracker inaccuracies, and speaker characteristics. Decision fusion provides a framework for capturing the reliability of each stream, by appropriately combining the likelihoods of single-modality HMM classifier decisions [47]. In isolated speech recognition, this can be easily implemented by calculating the combined likelihood for the acoustic and the visual observation for a given word model. However, in continuous speech recognition, the number of possible hypothesis of word sequences becomes very large, and the number of best hypothesis obtained for each stream might not necessarily be the same. Instead, it is simpler to carry out this combination at the HMM state level, by means of the multi-stream HMM classifier.

5.2.1 The Multi-Stream HMM

In its general form, the class conditional observation likelihood of the multi-stream HMM is the product of the observation likelihoods of its single-stream components, raised to appropriate *stream exponents* that capture the reliability of each modality, or, equivalently, the confidence of each single-stream classifier. Such model has been considered in multi-band audio-only ASR, among others [7, 39, 73]. In the audio-visual domain, the model becomes a two-stream HMM. As such, it has been extensively used in small-vocabulary audio-visual ASR tasks [28, 29, 48, 76, 86]. However, this work constitutes its first application to the LVCSR domain.

Given the bimodal (audio-visual) observation vector $\mathbf{o}^{(t)}$, the state emission (class conditional) probability of the multi-stream HMM is (see also (5.1) and (5.2)),

$$Pr[\,\mathbf{o}^{(t)}|\,\mathsf{c}\,] = \prod_{s\in\{\mathrm{A},\mathrm{V}\}} \Big[\sum_{j=1}^{J_{s\,\mathsf{c}}} w_{s\,\mathsf{c}\,j}\,\mathcal{N}_{D_s}(\,\mathbf{o}^{(t)};\mathbf{m}_{s\,\mathsf{c}\,j},\mathbf{s}_{s\,\mathsf{c}\,j}\,)\Big]^{\lambda_{s\,\mathsf{c}\,t}}\,. \tag{5.3}$$

In (5.3), $\lambda_{s\,\mathsf{c}\,t}$ are the stream exponents, that are non-negative, and, in general, depend on the modality s, the HMM state (class) $\mathsf{c}\in\mathcal{C}$, and, locally, on the utterance frame (time) t. Such time-dependence can be used to capture the "local" reliability of each stream, and

can be estimated on basis of stream confidences [1,63,85,93], for example, or acoustic signal characteristics [1], an approach which we consider in section 5.4, below.

In this section, we consider global, modality-dependent weights, i.e., two stream exponents constant over the entire database

$$\lambda_s = \lambda_{s\,\mathsf{c}\,t}\,, \quad \text{for all } \mathsf{c}\in\mathcal{C}\,, \quad \text{all } t\,, \quad \text{and } s = \mathrm{A}\,,\mathrm{V}\,. \tag{5.4}$$

Exponents λ_{A} and λ_{V} are constrained to satisfy

$$0 \leq \lambda_{\mathrm{A}}\,,\lambda_{\mathrm{V}} \leq 1\,, \quad \text{and } \lambda_{\mathrm{A}} + \lambda_{\mathrm{V}} = 1\,. \tag{5.5}$$

Clearly (see (5.3)), and in contrast to feature fusion techniques, the multi-stream HMM assumes class conditional independence of the audio and visual stream observations. This appears to be a non-realistic assumption.

5.2.2 Multi-Stream HMM Training

Training the multi-stream HMM consists of two tasks: First, estimation of its stream component parameters (mixture weights, means and variances), as well as, of the HMM state transition probabilities, and, second, estimation of appropriate exponents (5.4) that satisfy (5.5).

Maximum likelihood parameter estimation by means of the EM algorithm [82, 103] can be used in a straightforward manner to train the first set of parameters. This can be done in two ways: Either train each stream component parameter set separately, based on single-stream observations, and subsequently combine the resulting single-stream HMMs as in (5.3), or, train the entire parameter set (excluding the exponents) at once using the bimodal observations.

In the first case, the EM algorithm is invoked to separately train two single-modality, single-stream HMMs, i.e., an audio-only and a visual-only one, as in section 2.3. The visual-only HMM is forced to use the audio-only set of context dependent classes. This corresponds to the "AA" model discussed in section 4.1.4. Thus, assuming known stream exponents, the two resulting HMMs can be easily combined using emission probabilities given by (5.3), and a linear combination of their two transition matrices, weighted by stream exponents that satisfy (5.5), for example. An obvious drawback of this approach is that the two single-modality HMMs are trained asynchronously (i.e., using different forced alignments), whereas

	Clean audio	Noisy audio
Audio-only	14.44	48.10
AV-HiLDA	13.84	36.99
AV-MS-1	14.62	36.61
AV-MS-2	14.92	38.38

Table 5.2: Audio-visual decision fusion performance on the SI test set by means of the multi-stream HMM, separately trained as two single-stream models (AV-MS-2), or jointly trained (AV-MS-1). For reference purposes, audio-only and AV-HiLDA feature fusion WER (%) results are also depicted.

(5.3) assumes that the HMM stream components are state synchronous.

The alternative is to train the whole model at once, in order to enforce state synchrony. Due to the stream log-likelihood linear combination by means of (5.3), the EM algorithm carries on in the multi-stream HMM case with minor only changes [103]. The only modification is that the state occupation probabilities (or, forced alignment, in the case of Viterbi training) are computed on basis of the joint audio-visual observations, and the current set of multi-stream HMM parameters. Clearly, this approach requires an a-priori choice of stream exponents.

Such stream exponents cannot be obtained by maximum likelihood estimation [76]. Instead, *discriminative* training techniques have to be used, such as the *generalized probabilistic descent* (GPD) algorithm [17, 76], or *maximum mutual information* (MMI) training [18, 48]. The simple technique of directly minimizing the WER on a held-out data set can also be used. Clearly, a number of HMM stream parameter and stream exponent training iterations can be alternated.

Finally, decoding using the multi-stream HMM does not introduce additional complications, since, obviously, (5.3) allows a frame-level likelihood computation, like any typical HMM decoder.

5.2.3 State Synchronous Fusion Results

We have trained two multi-stream HMMs using the training procedures described in the previous section: First, we obtained a multi-stream HMM, referred to as AV-MS-2, by separately training two single-stream models, and subsequently combining them. A second multi-stream HMM, denoted by AV-MS-1, was jointly trained as a single model. For both models, the stream exponents were estimated to values $\lambda_{\mathrm{A}} = 0.7$, $\lambda_{\mathrm{V}} = 0.3$, in the clean

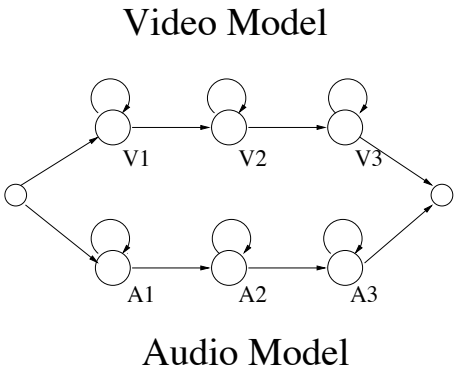


Figure 5.2: Phone synchronous multi-stream HMM for audio-visual fusion.

audio case, and $\lambda_{\rm A} = 0.6$, $\lambda_{\rm V} = 0.4$, in the noisy audio one. These values were obtained by minimizing the WER of various AV-MS-1 trained models on the SI held-out data set (see also Table 2.1). The audio-visual recognition results on the SI test set for both clean and noisy audio environments, obtained by rescoring "Lat" and "NAVLat" lattices, respectively, are depicted in Table 5.2. Baseline audio-only[1] and HiLDA-based audio-visual fusion results are also depicted for reference. As expected, the AV-MS-1 models outperformed the AV-MS-2 ones, but the AV-MS-1 HMM was unable to improve the clean audio-only system. This is somewhat surprising, and could indicate an inappropriate choice of stream exponents in this case. On the other hand, in the noisy audio case, the AV-MS-1 based decision fusion slightly outperformed the AV-HiLDA feature fusion method, and, by a significant amount, the audio-only baseline.

5.3 Phone Synchronous Decision Fusion

It is a well known fact that visual speech activity precedes the audio signal by as much as 120 ms [9,62], which is close to the average duration of a phoneme. The multi-stream HMM discussed above, however, enforces state synchrony between the audio and visual streams. It is therefore of interest to relax the assumption of state synchronous integration, and instead allow some degree of asynchrony between the audio and visual streams. Such a model is discussed in this section.

[1] Of course, the noisy audio-only result is obtained by rescoring lattices "NLat".

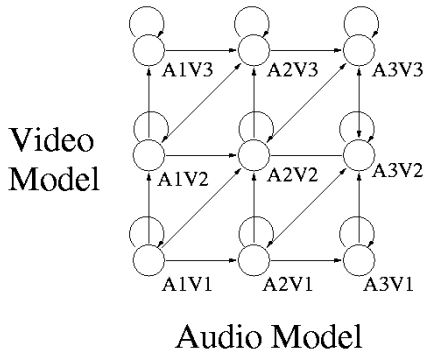


Figure 5.3: Equivalent audio-visual product HMM.

5.3.1 The Product HMM

An extension of the multi-stream HMM allows the single-stream HMMs to be in asynchrony within a model but forces them to be in synchrony at the model boundaries. Single-stream log-likelihoods are linearly combined at such boundaries using weights (or, equivalently, stream exponents, if we consider model probabilities) similarly to (5.3). For LVCSR, HMMs are typically phones, therefore, a reasonable choice for forcing synchrony constitute the phone boundaries. The resulting phone synchronous audio-visual HMM is depicted in Figure 5.2.

Decoding based on this integration method requires to individually compute the best state sequences for both audio and visual streams. To avoid the computation of two best state paths, the model can be formulated as a composite, or product, HMM [28, 29, 96]. Decoding under such a model requires to calculate a single best path. The product HMM consists of composite states that have audio-visual emission probabilities of the form (5.3), with audio and visual stream components that correspond to the emission probabilities of certain audio and visual-only HMM states, as depicted in Figure 5.3: These single-stream emission probabilities are tied for states along the same row, or column (depending on the modality), therefore the original number of mixture weight, mean, and variance parameters is kept in the new model. The transition probabilities of the single-modality HMMs are now shared by several transition probabilities in the composite model.

The product HMM allows to restrict the degree of asynchrony between the two streams, by excluding certain composite states in the model topology. As the number of states in the composite HMM is the product of the number of states of all its individual streams, such restrictions can reduce this number considerably, and speed up decoding. In the extreme case, when only the states that lie in its "diagonal" are kept, the model becomes equivalent

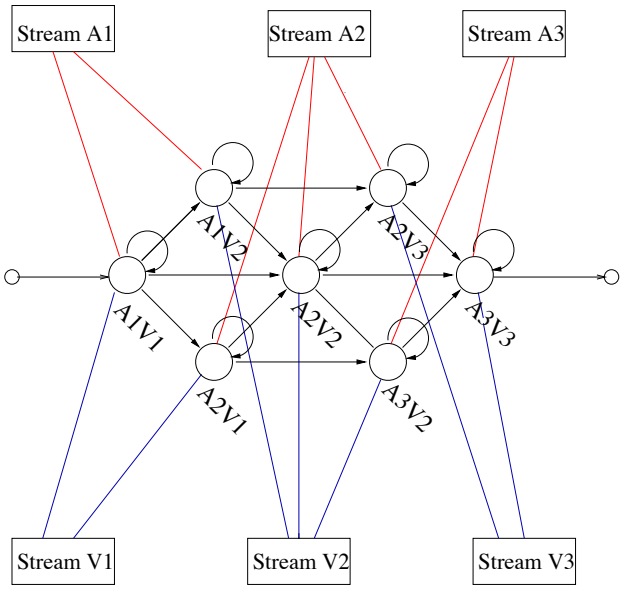


Figure 5.4: Stream tying in a product HMM with limited state asynchrony.

to the multi-stream HMM (see Figure 5.3).

5.3.2 Product HMM Training

Similarly to the multi-stream HMM, training of the product HMM can be done separately, or jointly. In the first case, an audio-only and a visual-only HMM are separately trained, based on single-modality observations. The composite model is then constructed based on the individual single-modality HMMs and appropriately chosen stream exponents and transition probabilities. In joint training, all product HMM parameters (with the exception of the stream exponents) are trained at once, by means of the EM algorithm, and using the audio-visual training data. In our experiments, and in view of the results in the multi-stream HMM case, we have only considered the second training approach. We have also limited the degree of asynchrony allowed to one state only, as shown in Figure 5.4. The resulting product phone HMMs had seven instead of the nine states of the full composite model. Stream tying is also depicted in Figure 5.4. Such tying was only kept up to the point where clustering was performed, as HTK does not support clustering of tied models. Although it would have been possible to tie the streams again after clustering, to our knowledge, the toolkit would not have allowed the creation of mixture distributions, tied both across states and across streams [103].

	Clean audio	Noisy audio
Audio-only	14.44	48.10
AV-HiLDA	13.84	36.99
AV-MS-1	14.62	36.61
AV-PROD	14.19	35.21

Table 5.3: Audio-visual decision fusion performance on the SI test set by means of the product HMM (AV-PROD). For reference purposes, audio-only, AV-HiLDA feature fusion, and AV-MS-1 decision fusion performance is also depicted. All results are in WER (%).

5.3.3 Phone Synchronous Fusion Results

Lattice rescoring experiments were conducted on the SI test set for both clean and audio conditions, using the jointly trained product HMM (AV-PROD) with limited state asynchrony, as discussed above. Stream exponents $\lambda_{\text{A}} = 0.6$, $\lambda_{\text{V}} = 0.4$, were used in the clean audio case, and $\lambda_{\text{A}} = 0.7$, $\lambda_{\text{V}} = 0.3$, in the noisy audio one. The obtained results are depicted in Table 5.3, and are compared to the baseline audio-only performance, as well as to the best feature fusion (AV-HiLDA) and decision fusion (AV-MS-1) techniques, considered so far. Clearly, the product HMM consistently exhibits superior performance to both audio-only and AV-MS-1 models, however it is worse than the AV-HiLDA model for clean speech. Overall, it achieves a 2% WER relative reduction in the clean audio case and a 27% one in noisy audio, over the corresponding audio-only system.

5.4 Class and Utterance Dependent Stream Exponents

In the previous two sections, we presented decision fusion algorithms that focus on both state synchronous (combining likelihoods at the state level) and asynchronous modeling (combining likelihoods at the phone level) of the audio and visual streams. The model investigated in this approach was the multi-stream HMM, and its phone synchronous variant, the product HMM. The state (class) conditional observation likelihood of these models is the product of the observation likelihoods of their audio-only and visual-only stream components, raised to appropriate stream exponents that capture the reliability of each modality. So far, we have considered global such exponents that depend on the modality only, and are estimated using held out data.

In this section, we expand on exponent estimation further, by investigating possible

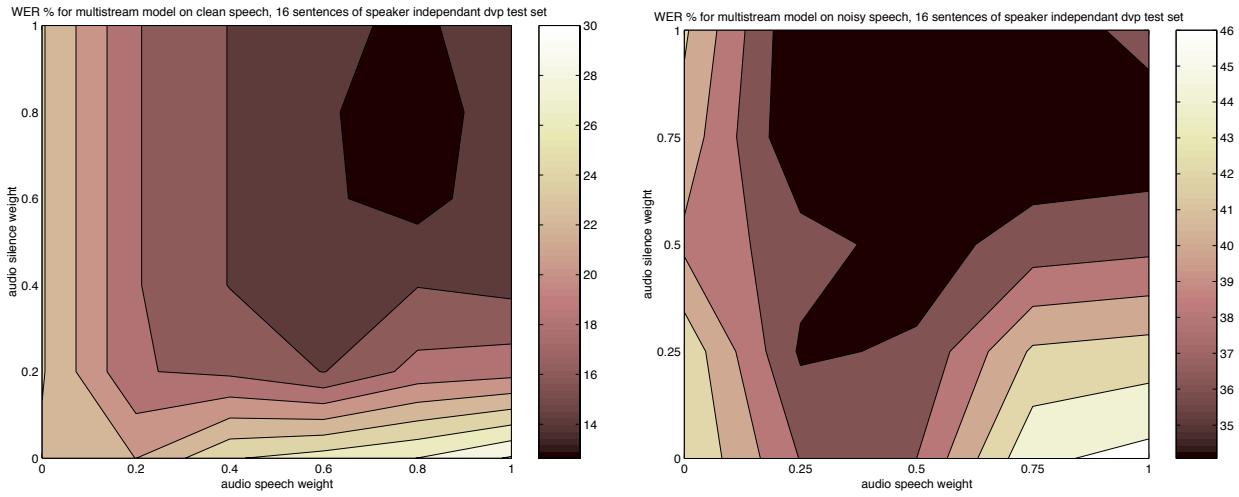


Figure 5.5: Effect of the variation of speech / silence dependent stream exponents on the WER of a 16-utterance subset of the SI held-out set. Audio stream exponents at a resolution of 0.1 have been considered for silence (ordinates) versus speech states (abscissa). *Left*: Clean audio. *Right*: Noisy audio.

refinements of stream exponent dependence. First, we consider exponents that depend on the HMM phone class, in addition to the modality. We investigate a very coarse such dependence, namely silence (/sil/, /sp/) versus non-silence state (phone) stream exponents. A finer such dependence has been considered in [48], with no definite conclusions. Subsequently, we consider exponents that are utterance dependent. Such exponents are estimated on basis of the degree of voicing present in the audio signal. Voicing is considered an indication of the reliability of the audio stream, and as such, this approach follows the concept of audio-visual adaptive weights used in [85,86].

5.4.1 Class Dependent Exponents: Silence Versus Speech

In this section, we investigate the effect of stream exponent dependence on silence ($\mathsf{c} \in \{/\text{sil}/,/\text{sp}/\}$) versus non-silence phone states $\mathsf{c} \in \mathcal{C} - \{/\text{sil}/,/\text{sp}/\}$ (see also (5.4)). We restrict weights $\lambda_{s\,\mathsf{c}}$ to satisfy (5.5), and we subsequently compute the WER for a small sample of 16 random utterances of the SI held-out set, for varying values of the speech and silence audio exponents (a step of 0.1 is used). The multi-stream HMM (AV-MS-1) is used for this task, with its stream exponents replaced by speech/silence state dependent ones. The results are ploted in Figure 5.5, for both clean and noisy audio.

Interestingly, the global audio weights (0.7, 0.7) and (0.6, 0.6), used in the previous sec-

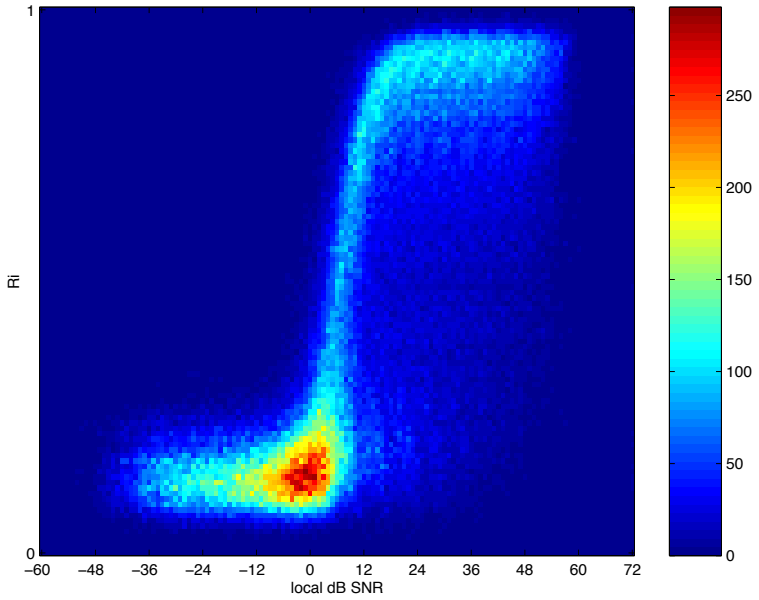


Figure 5.6: Histogram of R1R0 (R_i) of low frequency cells $[115, 629]$ Hz computed on 128 ms speech windows, and for 60 sentences, versus SNR in $[-21, 39]$ db and increments of 3 db, for white additive noise. Notice the clear nonlinear correlation between SNR and R1R0 (after [39]).

tions, for clean and noisy audio, respectively, do lie in the optimal (minimum WER) region. Furthermore, lower WERs are obtained for higher values $\lambda_{\mathrm{A},/\mathrm{sil}/}$ in both conditions. This suggests that silence is better modeled in the audio stream than by the video observations.

Notice however, that these results have been obtained on a very small number of sentences. At this point, no conclusions can be drawn about whether phone class dependent stream exponents are useful in state synchronous decision fusion by means of the multi-stream HMM. No such experiments have been carried out for the product HMM.

5.4.2 Utterance Dependent Stream Exponents

In this section, we investigate utterance dependent stream exponents, based on the audio-stream reliability. Traditionally, such reliability has been measured using the audio modality signal-to-noise ratio (SNR) [1, 45]. Here, instead, we propose the use of a measure of voicing, as a means of estimating the reliability of the audio observations. Specifically, we employ an equivalent to the *harmonicity index* (HNR) [4, 39, 105] to estimate the average voicing per utterance. Based on this index, we subsequently estimate utterance based stream exponents.

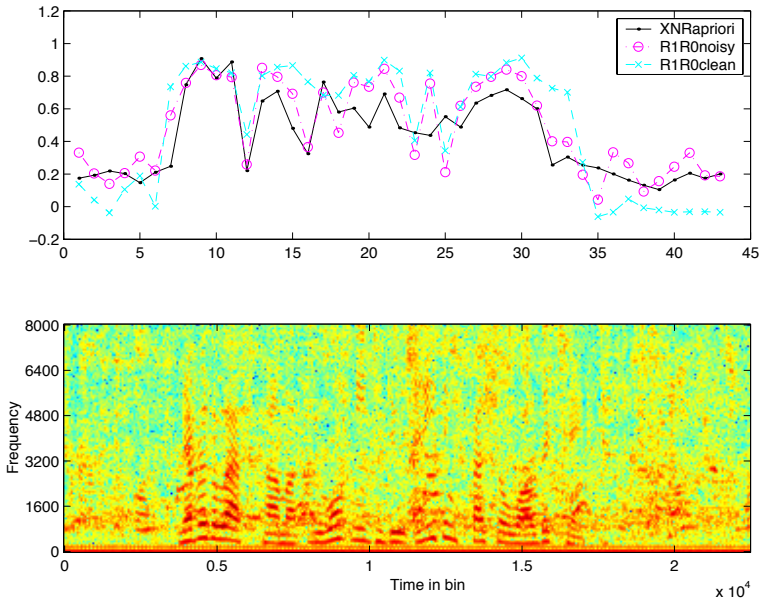


Figure 5.7: *Top*: Local estimates of R1R0 for clean (R1R0clean) and noisy (R1R0noisy) speech, and XNR, for a database utterance. All calculations are performed on 128 ms speech windows shifted by 64 ms. *Bottom*: Noisy audio spectrogram of the same utterance.

Voicing Estimation

We use the autocorrelogram of a demodulated signal as a basis for differentiating between a harmonic signal and noise. The peaks in the autocorrelogram isolate the various harmonics in the signal. The autocorrelogram can also be used to separate a mixture of harmonic noises and a dominant harmonic signal. An interesting property is that such separation can be efficiently accomplished, using a time window in the same range of the average phoneme duration [4,39], and in a frequency domain divided in four subbands (leading to the concept of multi-band speech recognition [7]).

A correlogram of a noisy cell is less modulated than a clean one. We use this fact to estimate the reliability of a cell [40] for which time and frequency definitions are compatible with the recognition process (128 ms of duration). Before the autocorrelation, we compute the demodulated signal after half wave rectification, followed by band-pass filtering in the pitch domain ([90,350] Hz). For each cell, we calculate the ratio $R_i = R1/R0$, where $R1$ is the local maximum in time delay segment corresponding to the fundamental frequency, and $R0$ is the cell energy. This measure is comparable to the HNR index [105]. Furthermore, it

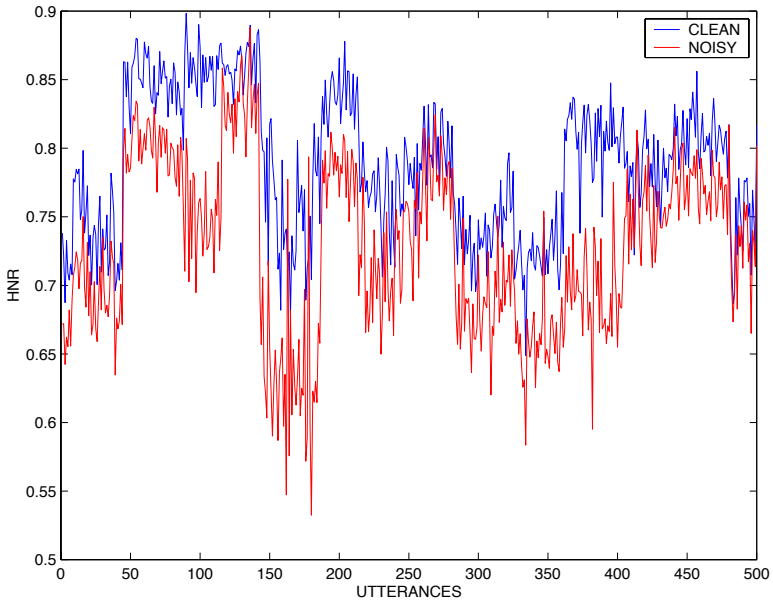


Figure 5.8: Utterance dependent $\lambda_{\mathrm{A}\,t}$ for the first 500 utterances of the SI test set, representing 14 speakers (nearly 40 consecutive utterances for each speaker are considered).

is strongly correlated with SNR in the 5–20 db range, as it is demonstrated in Figure 5.6 [4].

In Figure 5.7, we plot R1R0 estimates on 128 ms speech windows on a noisy database utterance, against the R1R0 estimates in the clean audio case, as well as, an SNR-alike measure, defined as XNR $= 10\ \log_{10}(S/(S+N))$. We observe that the biggest difference in R1R0 between the clean and noisy conditions occurs during silent frames. Notice that R1R0 and XNR are not strictly giving the same kind of information, but they are quite strongly correlated. Indeed, their correlation factor is 0.84, computed over the entire SI test set. Locally, R1R0 is higher than XNR on voiced parts, and it is lower on other parts. This local divergence could be well exploited in case we further refine stream exponent dependence at the frame level.

HNR Based Stream Exponents

In this first approach, audio speech reliability is calculated only from the regions where the speech is dominant. Because of the strong correlation between R1R0 and SNR, we assume that regions where local SNR is higher than 0 db, and strongly correlated to regions where R1R0 > 0.5 (see also Figure 5.6), are speech regions. We subsequently calculate stream

	Clean audio	Noisy audio
Audio-only	14.44	48.10
AV-HiLDA	13.84	36.99
AV-MS-1	14.62	36.61
AV-PROD	14.19	35.21
AV-MS-UTTER	13.47	35.27

Table 5.4: Audio-visual decision fusion performance on the SI test set by means of the the multi-stream HMM with utterance level, HNR-estimated stream exponents (AV-MS-UTTER). For reference purposes, audio-only, AV-HiLDA feature fusion, AV-MS-1, and AV-PROD decision fusion performance is also depicted. All results are in WER (%).

exponents $\lambda_{\mathrm{A}t}$, constant for all t within the utterance, to be the mean of all R1R0 values higher than 0.5. We assume this to be an adequate estimate of voicing within the utterance. Then, $\lambda_{\mathrm{V}t} = 1 - \lambda_{\mathrm{A}t}$ (see (5.5)).

As it is demonstrated by Figure 5.8, $\lambda_{\mathrm{A}t}$ is mostly speaker dependent, and in a smaller extent, utterance dependent, as well. For the entire SI test data set, the average $\lambda_{\mathrm{A}t}$ is calculated to be 0.79 and 0.73 for the clean and noisy audio case, respectively.

Multi-Stream HMM Results Using HNR Estimated Exponents

For each utterance of the SI test set, we replaced the exponents of the AV-MS-1 jointly trained multi-stream HMM, by the new HNR estimated exponents. We denote this fusion technique by AV-MS-UTTER. We subsequently estimated the resulting cumulative WER on the entire SI test set. The results for both clean and noisy audio are depicted in Table 5.4. In both cases, the algorithm outperformed the comparable AV-MS-1 system with global stream exponents. Furthermore, in the clean audio case, the algorithm outperformed even the product HMM with global exponents, resulting to a 7% WER relative reduction with respect to the audio-only system. Preliminary only, non-conclusive experiments were carried out using utterance dependent stream exponents, estimated by means of HNR, in the product HMM.

5.5 Utterance Level Discriminative Combination of Audio and Visual Models

The discriminative model combination (DMC) approach [5] aims at an optimal integration of independent sources of information in a log-linear model that computes the probability for a hypothesis. The parameters of this new model are the weights of the log-linear combination, and are optimized to minimize the errors in a held out set.

The combination can be performed either *statically*, with constant weights [5], or *dynamically*, where the parameters may vary for different segments of a hypothesis [12,99]. In the dynamic combination, the weights aim to capture the dynamic change of confidence on each of the models combined for each hypothesized segment.

5.5.1 Static Combination

We can combine the audio and visual model scores, along with a language model score, as independent sources of information in the DMC framework. If we denote by $P_{\mathrm{A}}(\mathbf{h}|\mathbf{O}_{\mathrm{A}})$ the probability provided by the audio model for a hypothesis $\mathbf{h} = [\, h_1, h_2, ..., h_{|\mathbf{h}|}\,]^{\top} \in \mathcal{H}$, given the acoustic observation $\mathbf{O}_{\mathrm{A}}$, by $P_{\mathrm{V}}(\mathbf{h}|\mathbf{O}_{\mathrm{V}})$ the probability provided the the visual model for the same hypothesis given the visual observation $\mathbf{O}_{\mathrm{V}}$, and by $P_{\mathrm{LM}}(\mathbf{h})$ the language model probability, then we define the log-linear model that combines all the available information $\mathcal{I}$ (audio, visual, and linguistic information) as:

$$P(\mathbf{h}|\mathcal{I}) = \frac{1}{Z_{\Lambda}(\mathcal{I})} P_{\mathrm{A}}(\mathbf{h}|\mathbf{O}_{\mathrm{A}})^{\lambda_{\mathrm{A}}} P_{\mathrm{V}}(\mathbf{h}|\mathbf{O}_{\mathrm{V}})^{\lambda_{\mathrm{V}}} P_{\mathrm{LM}}(\mathbf{h})^{\lambda_{\mathrm{LM}}}, \tag{5.6}$$

where $Z_{\Lambda}(\mathcal{I})$ is a normalization factor so that the probabilities for all possible lattice hypotheses $\mathbf{h} \in \mathcal{H}$ add to one. The weights in this formulation are constant for every model.

5.5.2 Dynamic Combination - Phone Dependent Weights

We can combine the scores from the available information sources dynamically, within the simple form of an exponential model, by weighting each of the scores with different exponents, for different segments of a hypothesis.

We decided to use phone level segments and the weight for each segment depends on the

identity of the hypothesized phone (similar to [99]):

$$P(\mathbf{h}|\mathcal{I}) = \frac{1}{Z_\Lambda(\mathcal{I})} \left(\prod_{i=1}^{|\mathbf{h}|} P_{\mathrm{A}}(h_i)^{\lambda_{\mathrm{A},h_i}} P_{\mathrm{V}}(h_i)^{\lambda_{\mathrm{V},h_i}} \right) P_{\mathrm{LM}}(\mathbf{h})^{\lambda_{\mathrm{LM}}}, \tag{5.7}$$

where h_i is the ith phone in hypothesis $\mathbf{h}$.

The weights $\lambda_{\bullet,\bullet}$ can be tied for different classes of segments. For example, we can have the same weight for all the consonants and the same for all the vowels as was examined in [12]. In the case of the visual model we can examine the case of having one weight for each of the different visemic classes.

5.5.3 Optimization Issues

The above defined model is used to rescore the n-best lists and choose the *maximum-a-posteriori* (MAP) candidate. We train the parameters $\lambda_\bullet$ in (5.6) and $\lambda_{\bullet,\bullet}$ in (5.7), so that the empirical word error count induced by the model is minimized. Since the objective function is not smooth, gradient descend techniques are not appropriate for estimation. We use the simplex downhill method known as *amoeba search* [69] to minimize the word errors on a held out set [98].

5.5.4 Experimental Results

The above described techniques were used to combine the scores from the available audio and visual models, in the clean only audio case.

We used the clean audio lattices "Lat" for our experiments, the SI held-out data set in order to optimize the weights, and the SI test set for testing. For the purposes of the experiments, 2000 best hypotheses were obtained for each utterance using acoustic model scores provided by IBM and they were then rescored with the new acoustic and visual models created in the workshop using HTK.[2]

We performed three experiments:

- The audio and visual models are combined statically with one weight for each of the models.

[2]Due to this rescoring of the n-best hypotheses, the baseline obtained using only the HTK audio model is slightly better than the one obtained using this model directly in the decoder (ROVER effect [31]).

	SI held-out	SI test
Baseline acoustic	12.8	13.65
DMC: Static (acoustic + visual) weights	12.5	13.35
DMC: 1 acoustic + 13 visemic weights	12.2	13.22
DMC: 43 phonemic-acoustic + 13 visemic weights	11.8	12.95

Table 5.5: Discriminative model combination fusion WER (%) results in the clean audio case.

- One global weight is still used for the audio model scores, but we use 13 different weights for visual models corresponding to the each of the 13 visemic classes of Table 4.1.
- Different weights are used for each of the 43 audio phone-models and each of the 13 visemic classes.

The results are depicted in Table 5.5. Significant gains have been obtained in the clean audio case. The DMC technique has outperformed all other decision fusion techniques, albeit with the caveat of a lower audio-only baseline (see also Table 5.4).

5.6 Summary

In this chapter, a number of feature fusion and decision fusion techniques have been applied to the problem of large vocabulary continuous audio-visual speech recognition. Some of these techniques have been tried before in small vocabulary audio-visual ASR tasks, such as concatenative feature fusion, as well as state- and phone-level decision fusion by means of the multi-stream and product HMMs, respectively. However, none of these methods have been applied to the LVCSR domain before. Furthermore, new fusion techniques were introduced in the workshop: The hierarchical LDA feature fusion technique, an HNR-based, utterance dependent, stream exponent estimation algorithm, as well as the composite model joint maximum likelihood training based on bimodal observations. Finally, the discriminative model combination approach has never before been considered for audio-visual ASR.

We have conducted fusion experiments in both clean and noisy audio conditions. In both cases, we were able to obtain significant performance gains over state-of-the-art baseline audio systems, by incorporating the visual modality. Thus, we demonstrated for the first time that speaker independent audio-visual ASR in the large vocabulary continuous speech

	Clean audio	Noisy audio
Audio-only	14.44	48.10
AV-concat	16.00	40.00
AV-HiLDA	13.84	36.99
AV-MS-1	14.62	36.61
AV-MS-2	14.92	38.38
AV-MS-UTTER	13.47	35.27
AV-PROD	14.19	35.21
AV-DMC	12.95	–

Table 5.6: Audio-visual feature and decision fusion results in WER (%) on the SI test set in both clean and matched noisy audio conditions.

domain is beneficial.

A summary of all workshop audio-visual fusion results is depicted in Table 5.6. A novel and simple feature fusion technique, namely the hierarchical LDA approach, gave us significant gains in both audio conditions considered. More complicated decision fusion techniques, by means of the multi-stream HMM with utterance dependent stream exponents, the product HMM, and the discriminative model combination for rescoring n-best hypotheses, resulted in additional gains. Overall, we achieved up to a 7% WER relative reduction in the clean audio case, and 27% WER reduction in the noisy case.

It is worth noticing that the nature of lattice rescoring experiments places limits to these improvements. It is worth conducting full decoding experiments with some of the decision fusion techniques considered. Furthermore, it is of interest to consider local stream exponent estimation schemes at the frame level, in conjunction with multi-stream, as well as, product HMMs.

Chapter 6

Summary and Discussion

When we proposed audio-visual speech recognition as a workshop theme, our goal was to bring together leading researchers in the field of audio-visual speech recognition to advance the state-of-the-art by carrying out experiments on a first-of-a-kind audio-visual large vocabulary continuous speech recognition database collected and provided by IBM.

At the highest level, we were very successful in achieving these goals. In our view, this workshop is a significant milestone in audio-visual, large vocabulary continuous speech recognition. We demonstrated improvements for the first time in this domain in the clean audio environment case by adding visual information. Specifically, by conducting audio lattice rescoring experiments, we showed a 7% relative word error rate (WER) reduction in that condition. Furthermore, we demonstrated a significant improvement of 27% WER relative reduction over audio-only matched models at a 10 dB SNR with additive speech "babble" noise.

Our main focus was audio-visual integration. In that aspect, we pursued several interesting experimental threads. We investigated discriminant visual feature representations, visual modeling using visual relevant clustering schemes, a novel feature fusion technique based on a hierarchical linear discriminant analysis, state-synchronous and phone-synchronous decision fusion and local weighting schemes at the utterance level and speech unit levels.

There are fundamentally four research areas in audio-visual speech recognition:

- *Visual speech ROI extraction*: Once we define what is an appropriate ROI, it is important to come up with techniques that robustly extract the ROI under a variety of visually variable conditions (lighting, scene, etc). We did not focus on this aspect of the problem during the workshop. Given the nature of the IBM audio-visual data, we used face and mouth tracking algorithms developed at IBM [89].

- *Visual speech representation*: What portion of the face provides all the visually relevant speech information? A simple low-level, video pixel based approach representing a rectangular box around the subject mouth (baseline in our experiments) appears to take us a long way. However, perceptual [92] and other experiments [53] suggest that more of the face region (including the cheeks and the jaw) carry useful information. In our experiments, we did some preliminary investigation by using representations of the whole face (active appearance models, in section 3.2), with limited success. However, we believe that the results are preliminary and the jury is out on this thread of experimentation. Also, 3-D aspects of the face during speech production appear to provide additional information (in particular, for languages like French). Such 3-D representations could also provide a greater degree of pose-invariance. Thus, 3-D visual speech representations are a potential direction for future pursuit.

- *Visual modeling*: Modeling context dependence is a key element of the progress that has been made in audio-based speech recognition. Most of the speech community has converged on using triphone contexts, while others (including IBM) use pentaphone contexts. In both cases, it is essential to discover the most meaningful contexts. This is often done by automatically discovering (using decision trees, for instance) similar contexts by grouping together phonetic contexts that are similar along some acoustic dimension. Obviously, acoustic similarities are not the most appropriate for visemes. So, we explored ways to develop visually meaningful groupings (based on the place of articulation) of phones and their use in developing triphone contexts that are similar. Our preliminary results did not show any improvements due to visually meaningful modeling (section 4.1.4). However, the investigation is too preliminary to come to any conclusions.

- *Audio-visual integration.* This we believe, is a wide open area for research with implications transcending the audio-visual speech recognition problem [71]. In audio-visual speech recognition, the key questions are:

  - *What is the right granularity for combining the decisions between the audio and visual sources of information?* A useful source of information that influences the decision is the experimentally observed asynchrony between the two streams [9]. Being the easiest from an implementation point of view, synchronous feature level fusion was the baseline in our experiments. Feature level fusion (synchronous fusion) using discriminant joint representations (HiLDA, see section 5.1.2) bought

us most of the gains during the workshop. We experimented with state-level decision fusion. Although this framework does not allow for asynchrony between the audio and visual streams, it does allow for weighting the decisions independently (section 5.2). We did not see any improvements over discriminant feature fusion for clean speech (in fact, it was slightly worse). We partially modeled the asynchrony between the streams by creating HMM topologies that permit asynchrony within a phone (section 5.3). Although this does not adequately address the asynchrony at onset, the approach used in the workshop lays the foundation for more general asynchronous models (at word and utterance level). We did see some additional improvements over feature level fusion by using this approach. However, our ability to investigate this further was limited by what we could implement in HTK in 6 weeks. Carefully modeling the asynchrony between the two streams by taking into account the sampling rates and the timing of information-bearing events is an area of research with a lot of potential.

– *How do you measure the reliability of the audio and visual information sources to weight the influence of the decisions in the combination?* Reliability of the audio and visual streams can be obtained by measures of the signal (such as the amount of noise using SNR) or by knowledge-based (perceptual or linguistic) aspects of the two streams or by data-driven approaches (discriminative training). We pursued two different lines of investigation. The first was based on perceptual and acoustic-phonetic knowledge. We used the fact that voicing is only available in the audio stream to define an utterance level voicing estimator to determine the relative weights. We did see improvements (section 5.4.2). Although, we used utterance level weighting schemes, more local (at the frame level or unit level) weighting schemes may be more appropriate. The second approach was a data-driven approach where individual stream weights were estimated at the appropriate unit level (phones for audio and visemes for visual) using a discriminative technique. Small improvements of the order of 5% relative in clean were observed (section 5.5). A combination of the two approaches may be a fruitful direction.

Acknowledgements

We would like to acknowledge a number of people for contributions to this work: First and foremost, Michael Picheny and David Nahamoo (IBM) for encouragement and support of the proposal of including audio-visual ASR in the summer workshop; Giridharan Iyengar and Andrew Senior (IBM) for their help with face and mouth region detection for the IBM ViaVoice^{TM} audio-visual data; Eric Helmuth (IBM) for his help in data collection; Asela Gunawardana and Murat Saraclar (CLSP, Johns Hopkins University) for their help with the HTK software toolkit. We would like to thank Jie Zhou, Eugenio Culurciello, and Andreas Andreou (Johns Hopkins University) for help with some of the CNN data collection and setup. Further, we would like to thank the Center for Language and Speech Processing staff for their help with arrangements during the workshop. Finally, we would like to thank Frederick Jelinek, Sanjeev Khudanpur and Bill Byrne (CLSP, Johns Hopkins University) for continuing to carry on the tradition of hosting the summer workshops. We believe that the workshop is a unique and beneficial concept, and we hope that funding institutions such as the NSF, DARPA and others will continue supporting it.

Bibliography

[1] A. Adjoudani and C. Benoît. On the integration of auditory and visual parameters in an HMM-based ASR. In Stork and Hennecke [91], pages 461–471.

[2] S. Basu, C. Neti, N. Rajput, A. Senior, L. Subramaniam, and A. Verma. Audio-visual large vocabulary continuous speech recognition in the broadcast domain. In *Proc. IEEE 3rd Workshop on Multimedia Signal Processing*, pages 475–481, Copenhagen, 1999.

[3] S. Basu, N. Oliver, and A. Pentland. 3D modeling and tracking of human lip motions. In *Proc. International Conference on Computer Vision*, 1998.

[4] F. Berthommier and H. Glotin. A new SNR-feature mapping for robust multistream speech recognition. In *Proc. International Congress on Phonetic Sciences (ICPhS)*, volume 1, pages 711–715, San Francisco, 1999.

[5] P. Beyerlein. Discriminative model combination. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 481–484, Seattle, 1998.

[6] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:113–120, 1979.

[7] H. Bourlard and S. Dupont. A new ASR approach based on independent processing and recombination of partial frequency bands. In *Proc. International Conference on Spoken Language Processing*, volume 1, pages 426–429, Philadelphia, 1996.

[8] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 557–560, Minneapolis, 1993.

[9] C. Bregler and Y. Konig. 'Eigenlips' for robust speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 669–672, Adelaide, 1994.

[10] N. Brooke. Talking heads and speech recognizers that can see: The computer processing of visual speech signals. In Stork and Hennecke [91], pages 351–371.

[11] N. M. Brooke and S. D. Scott. PCA image coding schemes and visual speech intelligibility. *Proc. Institute of Acoustics*, 16(5):123–129, 1994.

[12] W. Byrne, P. Beyrlein, J. M. Huerta, S. Khudanpur, B. Marthi, J. Morgan, N. Pterek, J. Picone, and W. Wang. Towards language independent acoustic modeling. Technical report, Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, 1999.

[13] M. T. Chan, Y. Zhang, and T. S. Huang. Real-time lip tracking and bimodal continuous speech recognition. In *Proc. IEEE 2nd Workshop on Multimedia Signal Processing*, pages 65–70, Redondo Beach, 1998.

[14] C. Chatfield and A. J. Collins. *Introduction to Multivariate Analysis.* Chapman and Hall, London, 1991.

[15] C. C. Chibelushi, F. Deravi, and J. S. D. Mason. Survey of audio visual speech databases. Technical report, Department of Electrical and Electronic Engineering, University of Wales, Swansea, 1996.

[16] G. Chiou and J.-N. Hwang. Lipreading from color video. *IEEE Transactions on Image Processing*, 6(8):1192–1195, 1997.

[17] W. Chou, B.-H. Juang, C.-H. Lee, and F. Soong. A minimum error rate pattern recognition approach to speech recognition. *Journal of Pattern Recognition and Artificial Intelligence*, III:5–31, 1994.

[18] Y.-L. Chow. Maximum mutual information estimation of HMM parameters for continuous speech recognition using the N-best algorithm. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 701–704, Albuquerque, 1990.

[19] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. European Conference on Computer Vision*, pages 484–498, 1998.

[20] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In D. Hogg and R. Boyle, editors, *Proc. British Machine Vision Conference*, pages 9–18. BMVA Press, 1992.

[21] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[22] F. Davoine, H. Li, and R. Forchheimer. Video compression and person authentication. In J. Bigün, G. Chollet, and G. Borgefors, editors, *Audio- and Video-based Biometric Person Authentication*, pages 353–360, Berlin, 1997. Springer.

[23] P. De Cuetos, C. Neti, and A. Senior. Audio-visual intent to speak detection for human computer interaction. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, Istanbul, 2000.

[24] J. R. Deller, Jr., J. G. Proakis, and J. H. L. Hansen. *Discrete-Time Processing of Speech Signals.* Macmillan Publishing Company, Englewood Cliffs, 1993.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[26] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis.* Wiley, 1998.

[27] P. Duchnowski, M. Hunke, D. Büsching, U. Meier, and A. Waibel. Toward movement-invariant automatic lip-reading and speech recognition. In *Proc. International Conference on Spoken Language Processing*, pages 109–112, 1995.

[28] S. Dupont and J. Luettin. Using the multi-stream approach for continuous audio-visual speech recognition: Experiments on the M2VTS database. In *Proc. International Conference on Spoken Language Processing*, Sydney, 1998.

[29] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia*, 2(3):141–151, 2000.

[30] G. J. Edwards, T. F. Cootes, and C. J. Taylor. Face recognition using active appearance models. In *Proc. European Conference on Computer Vision*, pages 582–595, 1998.

[31] J. G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. Workshop on Automatic Speech Recognition and Understanding*, 1997.

[32] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, 1996.

[33] B. Fröba, C. Küblbeck, C. Rothe, and P. Plankensteiner. Multi-sensor biometric person recognition in an access control system. In *Proc. 2nd International Conference on Audio and Video-based Biometric Person Authentication (AVBPA)*, pages 55–59, Washington, 1999.

[34] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Morgan Kaufmann, 1990.

[35] M. J. F. Gales. 'Nice' model based compensation schemes for robust speech recognition. In *Proc. ESCA-NATO Workshop on Robust Speech Recognition for Unknown Communication Channels*, pages 55–59, Pont-a-Mousson, 1997.

[36] M. J. F. Gales and S. J. Young. An improved approach to hidden Markov model decomposition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 729–734, San Francisco, 1992.

[37] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994.

[38] O. Ghitza. Auditory nerve representation as a front end for speech recognition in noisy environments. *Computer, Speech and Language*, 1:109–130, 1986.

[39] H. Glotin. *Elaboration et etúde comparative d'un système adaptatif de reconnaissance robuste de la parole en sous-bandes: Incorporation d'indices primitifs F0 et ITD.* PhD thesis, Doctorat de l'Institut National Polytechnique de Grenoble, Grenoble, 2000.

[40] H. Glotin, F. Berthommier, E. Tessier, and H. Bourlard. Interfacing of CASA and multistream recognition. In *Proc. Text, Speech and Dialog International Workshop (TSD)*, pages 207–212, Brno, 1998.

[41] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1983.

[42] R. A. Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 661–664, Seattle, 1998.

[43] H. P. Graf, E. Cosatto, and G. Potamianos. Robust recognition of faces and facial features with a multi-modal system. In *Proc. International Conference on Systems, Man, and Cybernetics*, pages 2034–2039, Orlando, 1997.

[44] M. S. Gray, J. R. Movellan, and T. J. Sejnowski. Dynamic features for visual speechreading: A systematic comparison. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 751–757, Cambridge, 1997. MIT Press.

[45] M. E. Hennecke, D. G. Stork, and K. V. Prasad. Visionary speech: Looking ahead to practical speechreading systems. In Stork and Hennecke [91], pages 331–349.

[46] H. Hermansky and N. Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[47] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[48] P. Jourlin. Word dependent acoustic-labial weights in HMM-based speech recognition. In *Proc. European Tutorial Workshop on Audio-Visual Speech Processing (AVSP)*, pages 69–72, Rhodes, 1997.

[49] P. Jourlin, J. Luettin, D. Genoud, and H. Wassner. Acoustic-labial speaker verification. *Pattern Recognition Letters*, 18(9):853–858, 1997.

[50] B. H. Juang. Speech recognition in adverse environments. *Computer, Speech and Language*, 5:275–294, 1991.

[51] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[52] R. Kaucic, B. Dalton, and A. Blake. Real-time lip tracking for audio-visual speech recognition applications. In B. Buxton and R. Cipolla, editors, *Proc. European Conference on Computer Vision*, volume II of *Lecture Notes in Computer Science*, pages 376–387, Cambridge, 1996. Springer-Verlag.

[53] T. Kuratate, H. Yehia, and E. Vatiokotis-Bateson. Kinematics based synthesis of realistic talking faces. In *Proc. Workshop on Audio Visual Speech Processing*, Terrigal, 1998.

[54] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9:171–185, 1995.

[55] R. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22(1), 1997.

[56] F. Liu, R. Stern, X. Huang, and A. Acero. Efficient cepstral normalization for robust speech recognition. In *Proc. ARPA Human Language Technologies Workshop*, 1993.

[57] J. Luettin. Towards speaker independent continuous speechreading. In *Proc. of the European Conference on Speech Communication and Technology*, pages 1991–1994, Rhodes, 1997.

[58] J. Luettin. *Visual Speech and Speaker Recognition*. PhD thesis, University of Sheffield, 1997.

[59] J. Luettin and N. A. Thacker. Speechreading using probabilistic models. *Computer Vision and Image Understanding*, 65(2):163–178, 1997.

[60] J. Luettin, N. A. Thacker, and S. W. Beet. Active shape models for visual feature extraction. In Stork and Hennecke [91], pages 383–390.

[61] J. Luettin, N. A. Thacker, and S. W. Beet. Speechreading using shape and intensity information. In *Proc. International Conference on Spoken Language Processing*, volume 1, pages 58–61, 1996.

[62] D. W. Massaro and D. G. Stork. Speech recognition and sensory integration. *American Scientist*, 86(3):236–244, 1998.

[63] I. Matthews. *Features for Audio-Visual Speech Recognition*. PhD thesis, School of Information Systems, University of East Anglia, Norwich, 1998.

[64] I. Matthews, T. Cootes, S. Cox, R. Harvey, and J. A. Bangham. Lipreading using shape, shading and scale. In *Proc. Workshop on Audio Visual Speech Processing*, pages 73–78, Terrigal, 1998.

[65] H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–748, 1976.

[66] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTS: The extended M2VTS database. In *Proc. 2nd International Conference on Audio and Video-based Biometric Person Authentication (AVBPA)*, pages 72–76, Washington, 1999.

[67] J. R. Movellan and G. Chadderdon. Channel seperability in the audio visual integration of speech: A Bayesian approach. In Stork and Hennecke [91], pages 473–487.

[68] A. Nadas, D. Nahamoo, and M. Picheny. Speech recognition using noise adaptive prototypes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:1495–1503, 1989.

[69] J. A. Nelder and R. Mead. A simplex method for function minimisation. *Computing Journal*, 7(4):308–313, 1965.

[70] C. Neti. Neuromorphic speech processing for noisy environments. In *Proc. IEEE International Conference on Neural Networks*, pages 4425–4430, Orlando, 1994.

[71] C. Neti, G. Iyengar, G. Potamianos, A. Senior, and B. Maison. Perceptual interfaces for human computer interaction: Joint processing of audio and visual information for human computer interaction. In *Proc. International Conference on Spoken Language Processing*, Beijing, 2000.

[72] L. Neumeyer, A. Sankar, and V. Digalakis. A comparative study of speaker adaptation techniques. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1127–1130, Madrid, 1995.

[73] S. Okawa, T. Nakajima, and K. Shirai. A recombination strategy for multi-band speech recognition based on mutual information criterion. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, volume 2, pages 603–606, Budapest, 1999.

[74] E. D. Petajan. Automatic lipreading to enhance speech recognition. In *Proc. Global Telecommunications Conference (GLOBCOM)*, pages 265–272, Atlanta, 1984.

[75] G. Potamianos, E. Cosatto, H. P. Graf, and D. B. Roe. Speaker independent audio-visual database for bimodal ASR. In *Proc. European Tutorial Workshop on Audio-Visual Speech Processing (AVSP)*, pages 65–68, Rhodes, 1997.

[76] G. Potamianos and H. P. Graf. Discriminative training of HMM stream exponents for audio-visual speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3733–3736, Seattle, 1998.

[77] G. Potamianos and H. P. Graf. Linear discriminant analysis for speechreading. In *Proc. IEEE 2nd Workshop on Multimedia Signal Processing*, pages 221–226, Redondo Beach, 1998.

[78] G. Potamianos, H. P. Graf, and E. Cosatto. An image transform approach for HMM based automatic lipreading. In *Proc. IEEE International Conference on Image Processing*, volume I, pages 173–177, Chicago, 1998.

[79] G. Potamianos and A. Potamianos. Speaker adaptation for audio-visual speech recognition. In *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, volume 3, pages 1291–1294, Budapest, 1999.

[80] G. Potamianos, A. Verma, C. Neti, G. Iyengar, and S. Basu. A cascade image transform for speaker independent automatic speechreading. In *Proc. International Conference on Multimedia and Expo*, volume II, pages 1097–1100, New York, 2000.

[81] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1988.

[82] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, 1993.

[83] C. R. Rao. *Linear Statistical Inference and Its Applications*. John Wiley and Sons, New York, 1965.

[84] R. R. Rao and R. M. Mesereau. Lip modeling for visual speech recognition. In *28th Annual Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 587–590, 1994.

[85] A. Rogozan. *Etude de la fusion des données hétérogènes pour la reconnaissande automatique de la parole audiovisuelle*. PhD thesis, University of Orsay-Paris XI, Paris, 1999.

[86] A. Rogozan, P. Deléglise, and M. Alissali. Adaptive determination of audio and visual weights for automatic speech recognition. In *Proc. European Tutorial Workshop on Audio-Visual Speech Processing (AVSP)*, pages 61–64, Rhodes, 1997.

[87] M. U. R. Sánchez, J. Matas, and J. Kittler. Statistical chromaticity-based lip tracking with B-splines. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, Munich, 1997.

[88] S. Sclaroff and J. Isidoro. Active blobs. In *Proc. International Conference on Computer Vision*, 1998.

[89] A. W. Senior. Face and feature finding for a face recognition system. In *Proc. 2nd International Conference on Audio and Video-based Biometric Person Authentication (AVBPA)*, pages 154–159, Washington, 1999.

[90] P. L. Silsbee. Motion in deformable templates. In *Proc. IEEE International Conference on Image Processing*, volume 1, pages 323–327, 1994.

[91] D. G. Stork and M. E. Hennecke, editors. *Speechreading by Humans and Machines: Models, Systems and Applications*, volume 150 of *NATO ASI Series F: Computer and Systems Sciences*. Springer-Verlag, Berlin, 1996.

[92] A. Q. Summerfield. Some preliminaries to a comprehensive account of audio-visual speech perception. In B. Dodd and R. Campbell, editors, *Hearing by Eye: The Psychology of Lip-Reading*, pages 97–113, Hillside, 1987. Lawrence Erlbaum Associates.

[93] P. Teissier, J. Robert-Ribes, and J. L. Schwartz. Comparing models for audiovisual fusion in a noisy-vowel recognition task. *IEEE Transactions on Speech and Audio Processing*, 7(6):629–642, 1999.

[94] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[95] O. Vanegas, A. Tanaka, K. Tokuda, and T. Kitamura. HMM-based visual speech recognition using intensity and location normalization. In *Proc. International Conference on Spoken Language Processing*, pages 289–292, Sydney, 1998.

[96] P. Varga and R. K. Moore. Hidden Markov model decomposition of speech and noise. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 845–848, Albuquerque, 1990.

[97] D. Vergyri. *Integration of Multiple Knowledge Sources in Speech Recognition Using Minimum Error Training*. PhD thesis, Center for Speech and Language Processing, The Johns Hopkins University, Baltimore, 2000.

[98] D. Vergyri. Use of word level side information to improve speech recognition. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, Istanbul, 2000.

[99] D. Vergyri, S. Tsakalidis, and W. Byrne. Minimum risk acoustic clustering for multilingual acoustic model combination. In *Proc. International Conference on Spoken Language Processing*, Beijing, 2000.

[100] T. Wark and S. Sridharan. A syntactic approach to automatic lip feature extraction for speaker identification. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3693–3696, Seattle, 1998.

[101] J. J. Williams, J. C. Rutledge, D. C. Garstecki, and A. K. Katsaggelos. Frame rate and viseme analysis for multimedia applications. In *Proc. IEEE 1st Workshop on Multimedia Signal Processing*, pages 13–18, Princeton, 1997.

[102] M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, version 1.2*. Addison-Wesley, third edition, 1999.

[103] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book*. Entropic Ltd., Cambridge, 1999.

[104] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[105] E. Yumoto, W. J. Gould, and T. Baer. Harmonic to noise ratio as an index of the degree of hoarseness. *Journal of the Acoustical Society of America*, 1971:1544–1550, 1982.