TCD-TIMIT: A New Database for Audio-Visual Speech Recognition

A dissertation submitted to the University of Dublin
for the degree of Master of Science

Eoin Gillen
Trinity College Dublin, May 2014

Signal Processing and Media Applications
Department of Electronic and Electrical Engineering
Trinity College Dublin

To my family.

Abstract

Automatic audio-visual speech recognition currently lags behind its audio-only counterpart in terms of research. One of the reasons commonly cited by researchers is the scarcity of suitable research corpora. Motivated by this issue, this thesis details the creation of TCD-TIMIT, a new corpus designed for continuous audio-visual speech recognition research.

TCD-TIMIT's design choices were made with respect to the databases currently available and their perceived limitations. TCD-TIMIT consists of high-quality audio and video footage of 62 speakers reading a total of 6913 sentences. Three of the speakers are professionally-trained lipspeakers, recorded to test the hypothesis that lipspeakers may have an advantage over regular speakers in automatic visual speech recognition systems. Video footage was recorded from two angles: straight and 30°.

After recording the footage, it was processed to create video and audio clips for each sentence. Audio, visual and joint audio-visual baseline experiments were then run on the database using basic state-of-the-art techniques. Baseline results were seen as an important component to provide along with the database. Separate experiments were run on the lipspeaker and non-lipspeaker data, and the results were compared. Visual and audio-visual baseline results on the non-lipspeakers were low overall. Results on the lipspeakers were found to be significantly higher than the non-lipspeaker results.

It is hoped that TCD-TIMIT will now be used to further the state of audio-visual speech recognition research.

Declaration

I hereby declare that this thesis has not been submitted as an exercise for a degree at this or any other University and that it is entirely my own work.

I agree that the Library may lend or copy this thesis upon request.

Signed,

Eoin Gillen

May 14, 2014.

Acknowledgments

Thanks first and foremost to my supervisor, Dr. Naomi Harte, for giving me the opportunity to work on this project, and then for her guidance, encouragement and patience. Thanks to all the Sigmedia gang, particularly Dave, Frank, François, Gary, Marcin, Finnian, Ian and Brian for their input and help with equipment. I'm truly grateful to have had the chance to work with a group I respect so highly. Thanks to the staff of the Electronic Engineering Department, particularly Conor for the football matches. Thanks to Luca for his help all the way from London.

Thanks to every volunteer who participated in TCD-TIMIT, and DeafHear.ie and their fantastic lipspeakers. This project was funded by Science Foundation Ireland, I'm very grateful to them for their support. Finally, thanks to my family for their constant support and encouragement.

Contents

List of Acronyms

AVSR Audio-Visual Speech Recognition

AVCSR Audio-Visual Continuous Speech Recognition

ASR Acoustic Speech Recognition

VSR Visual Speech Recognition

LVCSR Large Vocabulary Continuous Speech Recognition

SR Speech Recognition

FPS Frames Per Second

IPA International Phonetic Alphabet

MFCC Mel Frequency Cepstral Coefficients

DCT Discrete Cosine Transform

HMM Hidden Markov Model

PER Phoneme Error Rate

WER Word Error Rate

PCA Principal Component Analysis

ROI Region Of Interest

AAM Active Appearance Model

SNR Signal to Noise Ratio

MKV MatrosKa Video

P2FA Penn Phonetics Lab Forced Aligner

MLF Master Label File

PTS Presentation Time Stamp

AWGN Additive White Gaussian Noise

1

Audio-Visual Speech Recognition (AVSR) Databases

1.1 The Benefits of AVSR

Automatic speech recognition is becoming ubiquitous. It is currently touted as a feature in smartphones, cars and game consoles, among other places [67]. Microsoft, Apple, Google, Samsung and Blackberry are among the companies now providing speech recognition capabilities in their products. Current commercial speech recognition implementations, e.g. Apple's "Siri", use only audio input from the speaker. This leaves them vulnerable to performance degradations if the audio channel is corrupted by noise (e.g. imagine trying to use a computer's speech recognition function while someone else is talking nearby).

In an attempt to solve this issue, some researchers are focusing on the fact that speech recognition by humans involves visual as well as audio processing [17], [82]. A well-known example of this is the McGurk effect, published by McGurk and MacDonald in 1976 [56]. They found that when shown a video of a person saying "ga", with synchronized audio of the sound "ba", subjects reported hearing "da". This was the first demonstrated example of the influence of vision on human speech recognition. The test showed that the subjects were processing visual speech information from the speaker. This skill, known as lipreading or speechreading, is subconsciously used by most people. Some deaf and hard-of-hearing people actively train their lipreading skill as an accompaniment to, or substitute for, hearing speech. Contrary to the name lipreading, humans interpret face, tongue and lip movements for visual speech information, as well as other more abstract cues such as emotion and context.

Lipreading is of interest to researchers as it is almost completely unaffected by noise in

the audio channel. For a computer to be capable of visual speech recognition, it must have video footage of the speaker's face. In current approaches, relevant features from the face are then parametrized and statistical models are trained to recognize distinctive classes of features. The first joint audio-visual speech recognition system was developed by Petajan in 1984 [66]. The recognizer improved speech recognition results on a single-speaker, 100-word task. Since then, researchers have demonstrated the benefits of AVSR over audio-only speech recognition in a variety of tasks [68]. However, as [68] goes on to explain, it is difficult to compare the many algorithms that have been suggested, as "they are rarely tested on a common audio-visual database".

1.2 Currently-available AVSR databases

The lack of suitable databases is a commonly-cited issue in AVSR research [68], [9], [28], [81], [24] [7], [63]. A list of commonly-used AVSR databases is given in Table 1.1.

Table 1.1 shows the wide variability in size and intended purpose in current AVSR databases. Only five of the databases in the table are suitable for medium or large-vocabulary continuous speech recognition: AV-TIMIT, GRID, VidTIMIT, IBM LVCSR and AusTalk. Of these five, only GRID and VidTIMIT are currently available (AV-TIMIT and IBM LVCSR have not been released, while AusTalk is not complete as of writing but a release is planned). Of the remaining two, GRID is larger than VidTIMIT (1000 sentences vs 430) and filmed at a higher resolution, but its vocabulary (51 words) is much smaller.

While developing the DUTAVSC Dutch AVSR corpus, Chitu and Rothkrantz [10] undertook a review of corpora designed for AVSR. They make a number of design recommendations for any new AVSR database. Among their findings, they list common limitations found in the databases they reviewed:

- The recordings contain only a small number of respondents.
- The pool of utterances is usually very limited.
- The quality of the recordings is often very poor.
- The datasets are not publicly available.

Gan's Ph.D. thesis [24] on audio-visual continuous speech recognition (AVCSR) lists the criteria he followed in choosing a database for his experiments:

- The data should be continuously spoken.
- The corpus should support speaker independent recognition.
- The size of the vocabulary should also be large enough in order to provide for a variety of spoken sentences.

Table 1.1: List of English-language AVSR Databases (some information taken from tables in [7], [16] and [50])

Database Acronym	Speakers # (# Female)	Content e.g. isolated words	Video Resolution, FPS	Stated Purpose (SR=Speech Recognition)	Alternative Views?
AMP/CMU [8]	10 (3 F)	78 isolated words	720x480	N/A	N/A
AVletters [54]	10 (5 F)	Alphabet set	80x60, 25fps	Letter recognition	No
AV-TIMIT [31]	223 (106 F)	TIMIT-SX sentences	720x480, 30fps	Continuous SR	Different lighting
AVICAR [44]	86 (40 F)	Digits, TIMIT sentences	720x480, 30fps	SR in a car	4 camera angles
AVOZES [28]	20 (10 F)	Digits, continuous speech	720x480, 30fps	Continuous SR	Stereo cameras (3D)
BANCA [2]	208 (104 F)	Numbers, names, addresses	720x576, 25fps	Speaker verification	Different environments
CUAVE [64]	36 (17 F)	Digits	720x480, 30fps	Speaker-independent digit recognition	Pairs of speakers
DAVID [53]	258 (126 F)	Digits, alphabet, syllables and phrases	560x480, 25fps	Speaker/SR	Different backgrounds
GRID [12]	36 (16 F)	Command sentences	720x576, 25fps	Small-vocabulary CSR	No
IBM LVCSR [55]	290	Continuous speech	740x480, 30fps	LVCSR	N/A
VidTIMIT [73]	43 (19 F)	TIMIT sentences	512x384, 25fps	AVCSR	No
Valid [20]	106	Digit strings + sentence	720x576, 25fps	Speaker/SR	Different environments
TULIPS1 [59]	12 (3 F)	First 4 English digits	100x75, 30fps	Isolated digit recognition	No
XM2VTS [58]	295	Digit strings + sentence	720x576, 25fps	Speaker/SR	No
QuLips [63]	N/A	Digit strings + sentence	720x576, 25fps	Pose-invariant lipreading	10 different angles
CMU-AVPFV [45]	10	150 isolated words	640x480, 30fps	Examine profile vs front view lip features	Profile view
HIT-AVDB-II [50]	30 (15 F)	Digits, English and Chinese phrases	N/A	View angle for speaker and SR	0, 30, 60, 90°
LiLIR [48]	1	200 sentences	N/A	View angle for SR	0, 30, 45, 60, 90°
WAPUSK20 [84]	20 (9 F)	Command sentences	640x480, 32fps	Stereo views for SR	Stereo cameras (3D)
BAVCD [22]	15	Connected digits	640x480, 20fps	Visual and depth feature examination	Kinect depth map
UNMC-VIER [86]	123 (49 F)	digits, TIMIT sentences	708x640, 25fps	Environments and SR	Profile, webcam view
AusTalk [6]	1000	Digits, isolated words, SCRIBE sentences	640x480, 48fps	Speaker/SR	Stereo cameras (3D)

- The corpus should be spoken in English and available to the public.

Based on these criteria, Gan chose to work with the GRID corpus [12] and VidTIMIT [73]. In a similar situation, Cappelletta listed his ideal features for an AVSR database in his master's thesis [7] on visual continuous speech recognition (He chose to work with the VidTIMIT corpus):

- Continuous speech with enhanced phonetic content.
- Accurate phonetic transcription (using the largest possible phoneme set (see Section 2.1.4)).
- High frame per second video rate (50FPS or more).
- Focus on the speaker's head. Neither shoulders nor large background are needed.
- As many speakers as possible. Gender balanced and with a variety of facial hair (beard, moustaches, shaved) and skin tones.
- Several English accents.
- Some non-native English speakers (a minor percentage of the database).

From these lists, as well as the motivations given for AVSR corpora produced in the last decade, there seems to be at least some consensus between researchers on desirable features. The recurring themes in the requests are:

- A large number of speakers.
- Continuous speech with good coverage of phonemes and visemes.
- Available to other researchers.
- High-quality recordings.
- Gender balanced speaker set.

1.3 Proposed Database

The requests listed in Section 1.2 are the catalyst for the new audio-visual continuous speech recognition (AVCSR) database introduced in this work. The database attempts to fulfil as many of these requests as possible, as illustrated in Table 1.2. The database is named TCD-TIMIT as its speech material is 2255 sentences from TIMIT [46], an audio-only speech recognition database created in the 1980s (see Section 2.1.4). The database contains audio and video footage (from two angles) of 59 volunteers, as well as 3 professional lipspeakers (see Section 2.4). Volunteers say 98 sentences each, while the lipspeakers say 377 sentences each. Audio, visual and joint audio-visual speech recognition baselines are also provided with the database. These are given and discussed in this work.

Table 1.2: Features of new database with respect to requests of Section 1.2

Request	TCD-TIMIT
A large number of speakers	62 speakers (3 lipspeakers (see Section 2.4))
Continuous speech with good coverage of phonemes and visemes	Speech material: TIMIT sentences (see Section 3.1.1)
Available to other researchers	Release planned upon completion
High-quality recordings	Video resolution: 1920x1080, video frame rate: 30FPS
Gender balanced speaker set	32 male, 30 female
Alternative views	30° view available

1.4 Report Outline

The report is structured as follows:

Chapter 2 gives a brief overview of audio-only, visual-only and joint audio-visual speech recognition. The concepts of phonemes and visemes are discussed, as well as probabilistic modelling and parametrization techniques. The TIMIT corpus is introduced and its approach to phonemic labelling is described.

Chapter 3 details the steps taken to construct the new AVSR database. The reasoning behind design choices such as speech material, volunteers and post-processing workflow is given. The methods used to obtain time-aligned transcriptions and baseline results for the database are discussed.

Chapter 4 presents and discusses the baseline results from the experiments set up in Chapter 3. Various methods used to confirm the validity of the baselines are given. The implications of the results are discussed.

Chapter 5 introduces the secondary "lipspeaker" part of the TCD-TIMIT database. The experiments from Chapter 4 are re-run on the lipspeaker data and the results are compared to those in Chapter 4.

Finally, Chapter 6 offers concluding remarks and ideas for future work using the database.

2

Audio-Visual Speech Recognition Theory

2.1 Acoustic Speech Recognition (ASR)

One of the first computers capable of ASR was built by Davis et al. [13] of Bell Laboratories in 1952. The computer was named Audrey. It was speaker-dependent and could recognize only the digits 0-9. In the years since, acoustic speech recognizers have become vastly more complex and powerful, yet the core concept seen in Audrey; matching features of an incoming snippet of speech against statistically-obtained models, is still the same. At the time of writing, a state-of-the-art recognizer designed by Abdel-Hamid et al. [1] has achieved a phoneme error rate (PER) of 20.07% on the core test set of TIMIT, a medium-vocabulary continuous speech corpus developed in 1988 for testing recognizers. A good overview of the evolution of the state of the art in terms of performance on TIMIT is given by Lopes and Perdigao [51]. Their table is reprinted in Figure 2.1 as a reference.

2.1.1 Phonemes

The recognizer built by Davis et al. [13] in 1952 worked by computing the correlation between the 1st and 2nd formant frequencies seen in an unknown vowel, and a set of reference models obtained statistically. Speakers were required to leave a silence of at least 350ms before saying a number. This is because the values of the formants over the whole length of the utterance were matched against the models. While this worked for a vocabulary of ten words, to recognize a larger vocabulary the recognizer would have to be trained for each new word introduced. Therefore, most speech recognition research focuses on recognizing sub-word units known as

Figure 2.1: TIMIT Performance Timeline [51]

Year	System	Speech Technology	%Corr	%Acc	Test Set
1989	(Lee & Hon, 1989)	HMM	73.80	66.08	160 utterances (TID7)
1991	(Robinson & Fallside, 1991)	Recurrent Error Propagation Network	76.4 76.5	68.9 69.8	160 utterances (TID7) Complete Set
1992	(Young, 1992)	HMM	73.7	59.9	160 utterances randomly selected
1993	(Lamel & Gauvain, 1993)	Triphone Continuous HMMs	77.5	72.9	Complete Set
1994	(Robinson, 1994)	RNN	78.6 77.5	75.0 73.9	Complete Set Core Set
1998	(Halberstadt & Glass, 1998)	Heterogeneous input features. SUMMIT. Broad classes	-	75.6	Core Set
2003	(Reynolds & Antoniou, 2003)	MLP, Broad Classes	-	75.8	1152 utterances
2006	(Sha & Saul, 2006)	GMMs trained as SVMs	-	69.9	Complete Set
2006	(Schwarz et al., 2006)	TRAPs and temporal context division	-	78.52	Complete Set
2007	(Deng & Yu, 2007)	Hidden Trajectory Models	78.40	75.17	Core Set
2007	(Rose & Momayyez, 2007)	TDNN, phonological features HMM		72.2	Complete Set
2007	(Scanlon et al., 2007)	MLP/HMM	-	74.2	Complete Set
2007	ASAT, (Bromberg et al., 2007)	MLP/HMM	73.39	69.52	-
2007	(Siniscalchi et al., 2007)	TRAPs, temporal context division + lattice rescoring	-	79.04	Complete Set
2008	(Morris & Fosler-Lussier, 2006)	MLP/CRF	- 74.76	70.74 71.49	Core Set 944 utterances
2009	(Hifny & Renals, 2009)	Augmented CRFs	-	77.0	Complete Set + SA
2010	(Mohamed & Hinton, 2010)	Boltzmann Machines	-	77.3	Core set
2011	(Mohamed et al., 2011)	Monophone Deep Belief Networks	-	79.3	Core set

phonemes. Phonemes have been described by Gimson [27] as "The smallest contrastive linguistic unit which may bring about a change of meaning". As explained by Hosom [36], a phoneme "differentiates one word from another", and as such provides "a description of the speech signal at a level of abstraction that is especially useful for word-level speech processing". Using a limited number of phonemes, every word in a vocabulary can be distinguished. This is why phonemes are used in large-vocabulary speech recognition.

Phonemes are not the same as phones, which are defined in the Merriam-Webster online dictionary as "a speech sound considered as a physical event without regard to its place in the sound system of a language" [57]. Phonemes are concerned with meaning, while phones are concerned with sounds. For example (this example is adapted from one given in "The Elements of English" by Branford [3]), the International Phonetic Alphabet (IPA) phones p^h (aspirated "p") and p (unaspirated "p") are both possible realizations of the phoneme /p/. p^h is commonly found in the pronunciation of the word "pin" and p in the word "spin". If a speaker was to use p^h in the word "spin", it might sound unusual, but the meaning of the word would not be lost for most English speakers. So, p^h and p are both members of the family of phones that make up the phoneme /p/. They are called allophones of /p/. However, if a speaker replaced p with k in the word "spin", it would of course make the word "skin", which has a completely different meaning for English speakers. Hence, the phone k is not a member of the /p/ phoneme family, it belongs to another phoneme, /k/.

2.1.2 Speech Parametrization

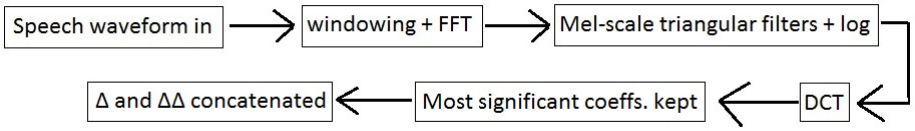
The theory behind parametrization of speech audio files is based on physiological and linguistic understandings of human speech production. A focus of early speech research was on attempting to use those understandings to identify phonemes. Jakobson et al. [39] proposed in 1963 that segmental phonemes could be characterized by 12 distinct features. The features were sets of opposite properties, so a phoneme either had one property or its opposite. The 12 features were:

Vocalic/Nonvocalic, Consonantal/Nonconsonantal, Compact/Diffuse, Tense/Lax, Voiced/Voiceless, Nasal/Oral, Discontinuous/Continuant, Strident/Mellow, Checked/Unchecked, Grave/Acute, Flat/Plain, Sharp/Plain.

In the years since, these feature definitions have been expanded and refined considerably by phoneticians to be more comprehensive. Huang et al. [37] mention that most phoneticians have converged on six features: high, low, front, back, round and tense. These form the basis of the International Phonetic Alphabet.

Furui [21] explains in his book how some of the features listed above can be distinguished by looking at the spectrogram and spectral envelope of an utterance. He shows that over "appropriately divided periods of 20-40ms", the spectrogram has roughly constant characteristics. Other features, as well as the phoneme boundaries themselves, are found by looking at the spectral variation graph. Periods of high spectral variation correspond to a change in phoneme. These

Figure 2.2: Typical MFCC Workflow



short-time variations can be examined by looking at the cepstrum of a signal, a concept first proposed in the realm of speech processing by Noll and Schroeder [61] in 1964 for detecting a speaker's pitch. Soon afterwards, according to Ganchev [25], short-time cepstral features became common in speech analysis. In brief, the method used for finding basic short-time cepstral features is:

Step 1) Window the speech signal (usually a Hamming window) to produce short segments (usually about 25ms in length).

Step 2) Perform a Fast Fourier Transform (FFT) on each segment.

Step 3) Get the logarithm of the FFT.

Step 4) Perform inverse FFT on the logarithm.

Essentially, what this does is give an insight into the rate of change of the main frequencies in the signal. Feature vectors can be extracted from the cepstrum. However, several methods of feature parametrization were originally proposed by various researchers. Some of the more popular methods were Real Cepstral Coefficients, Linear Prediction Coefficients, Linear Predictive Cepstral Coefficients and Mel Frequency Cepstral Coefficents (MFCCs). Eventually, in 1980 Davis and Mermelstein [14] showed MFCCs outperforming the other methods, which was later corroborated by other research. To this day, MFCCs are commonly used for parametrization. The method for extracting MFCCs is very similar to the method for extracting basic cepstral coefficients described above, with two differences. Between Step 2 and Step 3, the FFT powers obtained are mapped to the Mel frequency scale. Then, instead of the FFT in Step 4, a Discrete Cosine Transform (DCT) is applied to each vector, to decorrelate the features, and only the most significant coefficients (usually the first 12) are kept as the feature vector. Also, the 1st and 2nd order derivatives of these coefficients are often concatenated to the end of each feature vector to provide additional information about the vector's time variation characteristics. A typical MFCC workflow model can be seen in Figure 2.2.

2.1.2.1 Mel Frequency Scale

As explained by Huang et al. [37], the cochlea of a human ear is a spectrum analyzer, but in a complex and nonlinear way. Several frequency scales, such as the Bark and Mel scale, have been developed to try to mimic the weights assigned to different frequencies by the cochlea. The Mel

scale was first heuristically formulated by Stevens et al. [80] in 1937 after running an experiment on human frequency sensitivity with five volunteers. "Mel" is short for "melody". The scale was later revised, and revised again, in fact there are several versions of the scale in the literature. One of the more popular versions, mapping f (given in Hertz) to $B(f)$ (given in mel) is given as:

$$B(f) = 1125ln(1 + f/700)$$

2.1.3 Modelling Speech in Time - Hidden Markov Models

Sounds produced in speech are highly context dependent. The phenomenon of coarticulation - the effect of the previously-spoken sound on the next, means that the context cannot be ignored when trying to recognize phonemes. Coarticulation occurs between words and within words. For example, when said in a hurry, the word "handbag" can sound like "ham-bag". Without knowledge of this, a phoneme-level recognizer could not be used to recognize the word. At the level of natural, continuous speech recognition, humans form the context, the topic of conversation, from all previously-spoken words, and employ their knowledge of the topic to help identify the next word. For machines aiming to recognize at the phoneme level, however, the number of computations required to base decisions on all previously-seen phonemes would become prohibitively large after just a few words were spoken! It is for this reason that a simplified way of modelling this context dependence was proposed and became popular - the Hidden Markov Model (HMM).

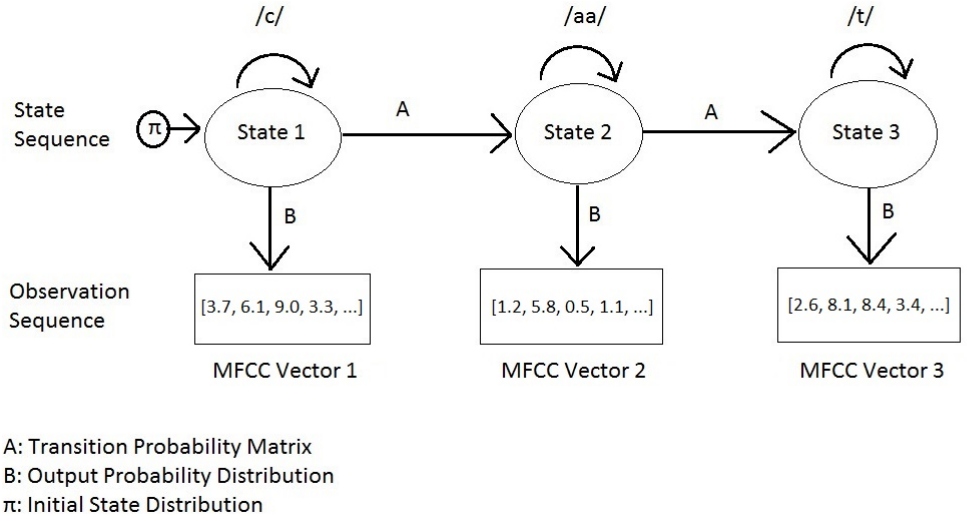
The theory behind HMMs for speech recognition was formulated by Rabiner [70]. The theory in this section is adapted from his tutorial paper and the tutorial by Young et al. [89] in the HTK manual. In a HMM, the Markov property is invoked, i.e. only the last state affects the probability of the next. Formally, with $S_1, S_2, S_3...S_t$ as a sequence of random variables, a Markov Chain is given as

$$Pr(S_{t+1} = s|S_1 = s_1, S_2 = s_2, ..., S_t = s_t) = Pr(S_{t+1} = s|S_t = s_t)$$

A Markov Chain typically comes with two parameters: the state transition matrix A, which contains the probabilities of transitioning from any state into another, and the initial distribution π, which gives the initial probability of being in each state.

A HMM is basically a Markov Chain in which each state transition also causes an associated probability distribution to output a value. In this way, each state and its associated probability distribution behave like another two-state Markov Chain, since the output depends on the state. The state itself at any given time is hidden to an observer, but the observer can see the output from the probability distributions. By looking at the outputs over time, the observer can make an educated guess as to the most likely state transitions, called the state sequence, that took place. For speech, the state sequence describes the progression from recognizable sound to recognizable sound over time. The output from the probability distributions is called

Figure 2.3: Simplified example of a HMM for the word "cat"



the observation sequence, and is a set of feature vectors (usually MFCCs) corresponding to the state sequence. A simplified example of a HMM for the word "cat" is given in Figure 2.3.

Right now, the "cat" HMM is set up to output the probability of an observation vector sequence O, given the parameters of the model $\lambda = [A, B, \pi]$, i.e.

$$Pr[O|\lambda] = \sum_{S} Pr[O|S, \lambda] Pr[S|\lambda].$$

Of course, a speech recognizer needs to compute the probability of a state sequence given an observation sequence, not the other way around, so Bayes' Rule is invoked to obtain

$$Pr[\lambda|O] = \frac{Pr[O|\lambda]Pr[\lambda]}{Pr[O]}.$$

This means that the speech recognizer needs to have knowledge of the parameters of the model before trying to compute a state sequence probability for a new observation sequence. This is done by "training" the model, feeding it observation sequences for which the corresponding state sequences are already known, and adapting its parameters to fit that data. In short, the training is meant to solve Rabiner's third problem. Rabiner [70] identified three problems that have to be solved when trying to obtain $Pr[\lambda|O]$:

Problem 1) Given the observation sequence $O = o_1, o_2, ..., o_t$, and the model λ, how do we efficiently compute $Pr[O|\lambda]$?

Problem 2) Given the observation sequence $O = o_1, o_2, ..., o_t$, and the model λ, how do we choose a corresponding state sequence $S = s_1, s_2, ..., s_t$ which is optimal in some meaningful sense (i.e. best "explains" the observations)?

Problem 3) Given the model λ, how do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximise $Pr[O|\lambda]$?

Rabiner's proposed solutions to problems 1, 2 and 3 are the Forward procedure, the Viterbi algorithm and the Baum-Welch method respectively. They are described in his tutorial paper [70]. Once the model's parameters have been adjusted using the training data, it can be used to output state sequence probabilities for new observation sequences.

2.1.4 Phonemes of the TIMIT Speech Corpus

As mentioned in Section 2.1, TIMIT is a frequently-used corpus for continuous speech recognition. The TIMIT speech corpus contains 6300 sentences spoken by 630 speakers (10 sentences each) from 8 regions of the USA. The database was jointly developed by Texas Instruments (TI) and MIT (hence the name). The sentences in the database were chosen to include as many phoneme pairs as possible. Each sentence clip comes with a transcription transcribed by a phonetician. The recording process, and the difficulties in phonetically transcribing continuous speech, are described in a report by Lamel et al. [46]. In particular, the report mentions problems in deciding phoneme labels and boundaries in cases of "severe coarticulation effects". Nevertheless, the labels are used as ground truth when training an acoustic speech recognizer.

For a given TIMIT audio file (e.g. SI1269.wav), its corresponding phoneme label file can be found in the same directory with the extension ".PHN" (e.g. SI1269.PHN). An example label file can be seen in Figure 2.4. The boundaries are given in units of samples/second. TIMIT was recorded at 16kHz. Also worth noting is that TIMIT also provides word-level label files, with the extension ".WRD". These contain the start and end time (in samples/second) of each word in the given sentence.

TIMIT is labelled using 61 phonemes. 52 of these phonemes are given in Table 2.1. The remaining 9 not shown are "pau", to indicate pauses, "epi" to indicate epenthetic silence, "h#" and "#h" to indicate silence at the start and end of a sentence respectively, and finally, vowel variants which are distinguished from their unstressed counterparts. For example, "AO1", the "1" indicating primary stress, is treated as a separate vowel to "AO", which is unstressed.

As Lee and Hon [49] (whose results can be seen at the top of Figure 2.1) found in 1989, there are a few issues with the phonemes of Table 2.1 from the perspective of an acoustic speech recognizer. Some of the phonemes rarely occur in the TIMIT sentences, leaving very little data to train accurate models for them. Others sound too similar to be distinguishable. Their solution was to reduce the 61 phonemes to a 39-phoneme set, a decision which was adopted by most subsequent experimenters. The reduction rules they applied are given in Table 2.2, and were used in this work also.

Figure 2.4: Phoneme-level label file for TIMIT sentence SI1269. The sentence transcribed in this case is "Resistance thermometers".

SI1269.PHN

```
1   0 2370 h#
2   2370 3007 r
3   3007 4120 ax-h
4   4120 5710 s
5   5710 6970 ih
6   6970 7924 s
7   7924 8271 tcl
8   8271 9090 t
9   9090 10850 en
10  10850 13430 s
11  13430 14830 th
12  14830 15456 axr
13  15456 16200 m
14  16200 18280 aa
15  18280 19007 m
16  19007 19880 ax
17  19880 20440 dx
18  20440 22990 er
19  22990 25700 s
20  25700 27680 h#
```

2.1.4.1 Phonemes or Phones?

At this point there is a small but confusing semantic issue. TIMIT's documentation actually states that it uses a 61-*phone* set, but this arguably stretches the definition of the word phone given in Section 2.1.1. But while TIMIT stretched the definition, the reduced set given by Lee and Hon [49] (seen in Table 2.2) are far more phonemes than phones, because allophones have been mapped into single classes. In the paper, Lee and Hon refer to the reduced set as a "list of the phones", but they are later referred to as phonemes by Young [88] and other researchers. Also, Hosom [36] uses the term phonemes to refer to TIMIT's original 61 sub-word units. Thus, the term phoneme will be used in this report to describe TIMIT's original 61 sub-word units and the reduced set in Table 2.2.

2.1.5 Evaluating Recognition Performance on Phonemes

When recognizing speech at the phoneme level, a method is needed to evaluate the recognition results that takes into account hits, substitutions S, insertions I and deletions D. The usual method, described in the HTK manual by Young et al. [88] is to define the correctness score C as

$$C = \frac{N - D - S}{N} \times 100\% \tag{2.1}$$

Table 2.1: Original TIMIT Phoneme Set [26]

Symbol	Example Word	Transcription	Symbol	Example Word	Transcription
b	bee	BCL B iy	d	day	DCL D ey
g	gay	GCL G ey	p	pea	PCL P iy
t	tea	TCL T i	k	key	KCL K iy
dx	muddy	m ah DX iy	q	bat	bcl b ae Q
jh	joke	DCL JH ow kcl k	ch	choke	TCL CH ow kcl k
s	sea	S iy	sh	she	SH iy
z	zone	Z ow n	zh	azure	ae ZH er
f	fin	F ih n	th	thin	TH ih n
v	van	V ae n	dh	then	DH e n
m	mom	M aa M	n	noon	N uw N
ng	sing	s ih NG	em	bottom	b aa tcl t EM
en	button	b aa q EN	eng	washing	w aa sh ENG
nx	winner	w ih NX axr	l	lay	L ey
r	ray	R ey	w	way	W ey
y	yacht	Y aa tcl t	hh	hay	HH ey
hv	ahead	ax HV eh dcl d	el	bottle	bcl b aa tcl t EL
iy	beet	bcl b IY tcl t	ih	bit	bcl b IH tcl t
eh	bet	bcl b EH tcl t	ey	bait	bcl b EY tcl t
ae	bat	bcl b AE tcl t	aa	bott	bcl b AA tcl t
aw	bout	bcl b AW tcl t	ay	bite	bcl b AY tcl t
ah	but	bcl b AH tcl t	ao	bought	bcl b AO tcl t
oy	boy	bcl b OY	ow	boat	bcl b OW tcl t
uh	book	bcl b UH kcl k	uw	boot	bcl b UW tcl t
ux	toot	tcl t UX tcl t	er	bird	bcl b ER dcl d
ax	about	AX bcl b aw tcl t	ix	debit	dcl d eh bcl b IX tcl t
axr	butter	bcl b ah dx AXR	ax-h	suspect	s AX-H s pcl p eh kcl k tcl t

and the accuracy score A as

$$A = \frac{N - D - S - I}{N} \times 100\%, \tag{2.2}$$

where N is the total number of phonemes in the reference transcription. Since the accuracy score takes insertions into account, it is often a better indicator of the recognizer's true performance, but bear in mind that if the number of insertions is high enough, it can return a negative score. The Phoneme Error Rate (PER) mentioned above in Figure 2.1, is another name for $1 - A$. The Word Error Rate (WER) is $1 - A$ when recognizing at the word level.

Table 2.2: Reduced Phoneme Set of Lee and Hon [49]

/Original Phoneme/	/Final Phoneme/
/ao/	/aa/
/ux/	/uw/
/axr/	/er/
/hv/	/hh/
/ix/	/ih/
/el/	/l/
/em/	/m/
/zh/	/sh/
/eng/	/ng/
/en/, /nx/	/n/
/ax/, /ax-h/	/ah/
/pcl/, /tcl/, /kcl/, /bcl/, /dcl/, /gcl/, /h#/, /#h/, /pau/, /epi/	/sil/
/q/	none

2.2 Visual Speech Recognition (VSR)

2.2.1 Visemes

Visual speech recognition relies on an analogue to the phonemes of Section 2.1, called visemes. Unfortunately, as Cappelletta [7] explains in his thesis, visemes are not as well defined as phonemes. In fact, there are two competing definitions, with neither clearly superior.

The first definition (data-driven approach) is based on articulatory gestures: “Visemes can be thought of in terms of articulatory gestures, such as the lips closing or rounding, teeth exposure, jaw movement etc., without a link to the uttered phoneme”. Going by this definition, joint audio-visual recognition becomes more complicated, as there is no link between the phonemes and visemes.

The second definition (linguistic approach) is based on the corresponding phonemes: “Visemes are derived from groups of phonemes having the same visual appearance”. This definition allows for phoneme-to-viseme maps, which are many-to-one maps, as some phonemes are visually indistinguishable. Thus, some information is lost straight away.

Note that neither definition suggests that visemes distinguish words, as phonemes do. In fact, according to the second definition, some words definitely will have the exact same viseme sequence. The second definition is more widely used than the first [7].

2.2.2 Phoneme-to-Viseme Maps

According to Cappelletta [7], the most-used phoneme-to-viseme map in the literature is the one developed by Neti et al. [68]. The map is given in Table 2.3.

Table 2.3: Neti Map

Viseme	TIMIT Phonemes	Description
/V1	/ao/ /ah/ /aa/ /er/ /oy/ /aw/ /hh/	Lip-rounding based vowels
/V2	/uw/ /uh/ /ow/	”
/V3	/ae/ /eh/ /ey/ /ay/	”
/V4	/ih/ /iy/ /ax/	”
/A	/l/ /el/ /r/ /y/	Alveolar-semivowels
/B	/s/ /z/	Alveolar-fricatives
/C	/t/ /d/ /n/ /en/	Alveolar
/D	/sh/ /zh/ /ch/ /jh/	Palato-alveolar
/E	/p/ /b/ /m/	Bilabial
/F	/th/ /dh/	Dental
/G	/f/ /v/	Labio-dental
/H	/ng/ /g/ /k/ /w/	Velar
/S	/sil/ /sp/	Silence

Part of Cappelletta's thesis involved comparing the Neti map with four other phoneme-to-viseme maps used in the literature. Using the VidTIMIT audiovisual speech corpus mentioned in Table 1.1, he trained a visual-only speech recognizer for each map. He used the maps to create a set of label files for each recognizer so he could compare their performance. Although there was very little training data, and the maps have different numbers of visemes, he found that, of the five maps, a map designed by Jeffers and Barley [40] produced the highest correctness and accuracy scores for three different feature types and both 3 and 4-state HMMs. The map is given for TIMIT's phoneme set in Table 2.4.

Two TIMIT phonemes are not present in the map: /hh/ and /hv/. Cappelletta explains that these two phonemes do not have any viseme associated with them, as a speaker produces these phonemes while forming the next viseme with their mouth. As a result, when these phonemes appear in a transcription they are mapped to the next viseme in the sequence.

The main performance difference between the Jeffers map (Table 2.4) and the Neti map (Table 2.3) was found to stem from the fact that the Jeffers map has two large vowel visemes (B and I) and two small ones (D and G). In contrast, the vowel visemes of the Neti map are all roughly the same size. To test this, Cappelletta created hybrid maps in which vowel visemes were not based on corresponding phonemes, but classified using Self-Organizing Maps (SOM). New visual-only recognizers were then trained. Compared to the original Jeffers map, correctness

Table 2.4: Jeffers and Barley Map

Viseme	TIMIT Phonemes	Description	Visibility Rank	Occurrence [%]
/A	/f/ /v/	Lip to Teeth	1	3.15
/B	/er/ /ow/ /r/ /q/ /w/ /uh/ /uw/ /axr/ /ux/	Lips Puckered	2	15.49
/C	/b/ /p/ /m/ /em/	Lip Together	3	5.88
/D	/aw/	Lips Relaxed-Moderate Opening to Lips Puckered-Narrow	4	0.7
/E	/dh/ /th/	Tongue Between Teeth	5	2.9
/F	/ch/ /jh/ /sh/ /zh/	Lips Forward	6	1.2
/G	/oy/ /ao/	Lips Rounded	7	1.81
/H	/s/ /z/	Teeth Approximated	8	4.36
/I	/aa/ /ae/ /ah/ /ay/ /ey/ /ih/ /iy/ /y/ /eh/ /ax-h/ /ax/ /ix/	Lips Relaxed Narrow Opening	9	31.46
/J	/d/ /l/ /n/ /t/ /el/ /nx/ /en/ /dx/	Tongue Up or Down	10	21.1
/K	/g/ /k/ /ng/ /eng/	Tongue Back	11	4.84
/S	/sil/ /pcl/ /tcl/ /kcl/ /bcl/ /dcl/ /gcl/ /h#/ /#h/ /pau/ /epi/	Silence	-	-

(a) Phoneme /aa/, Viseme /I **(b)** Phoneme /sh/, Viseme /F **(c)** Phoneme /uw/, Viseme /B

Figure 2.5: Viseme ROI Examples

and accuracy both fell (using 4-state HMMs) using the Jeffers hybrid map. On the other hand, the recognizer built using the hybrid Neti map was more accurate than the original.

2.2.3 Feature Extraction for Visemes

There are two main schools of thought regarding feature extraction for visemes. One school believes that features describing visemes can be found by modelling the shapes made by the

mouth, obtaining parameters like curvature of the lips, mouth width, area between upper and lower lip etc. Features from this school of thought are called shape-based features. The other school of thought believes that viseme features can be obtained from the pixel values directly. The sub-image around the mouth is extracted and parametrized using a technique like Principal Component Analysis (PCA), Discrete Wavelet Transform (DWT) or Discrete Cosine Transform (DCT). These features are called appearance-based features.

Shape-based features contain more information about the mouth shape, but require precise lip-tracking. In comparison, the bounding box around the mouth needed for appearance-based features (called the Region of Interest (ROI)) is easier to extract. In 2001, Matthews et al. [55] compared DCT, DWT, PCA and a combined shape and appearance model called an Active Appearance Model (AAM). They used IBM's LVCSR database to obtain speaker-independent, large-vocabulary, continuous audio-visual recognition word error rates (WER). Of the four models, DCT was found to produce the lowest WER. In 2008, Seymour et al. [76] compared four appearance-based feature extraction methods: DCT, PCA, Linear Discriminant Analysis (LDA) and Fast Discrete Curvelet Transform (FDCT). The task was speaker-independent digit recognition. They found that DCT produced the lowest WER. In 2009, Lan et al. [47] compared appearance and shape-based features for an isolated word recognition task on the GRID database. They found that appearance-based features performed better, and that an AAM offered only slightly better performance than appearance-based features alone.

Part of Cappelletta's thesis [7] involved comparing the performance of PCA, DCT and another appearance-based method called Optical Flow (based on the mouth's movement vectors from video frame to frame). The task was large-vocabulary continuous viseme recognition. He found that PCA gave the most correct hits, but that DCT was the most accurate.

Based on these findings, DCT was chosen as the feature extraction method for visemes in this work. DCT feature extraction involves two steps: extracting the ROI around the mouth, and applying the DCT to the ROI.

2.2.3.1 ROI Extraction

The ROI extraction method developed by Cappelletta [7] was used. The method is fully described in his thesis, so a short overview will be given here.

The method works by identifying and tracking the nostrils of the speaker. For a given video clip, the initial position (in frame 1) of the center of the nose is also required as an input. From this, the nostril positions are identified by searching for the darkest "blobs" in the surrounding region. From the nostril positions, the bounding box for the mouth is defined, extracted, rotated (if necessary) and output as a smaller image. The nostril positions are tracked throughout the remaining frames.

The method sometimes fails when nostrils are occluded (e.g if a speaker lowers their head), when the speaker has facial hair or when the speaker's skin colour is dark. In these cases, the

speaker's nostril positions in each frame have to be manually specified to the ROI-extraction part of the method.

After ROI extraction, the DCT is applied to the output images.

2.2.3.2 Discrete Cosine Transform (DCT)

The discrete cosine transform is related to the Discrete Fourier Transform (DFT) in that it also tries to express a vector in terms of frequencies. The DCT is often used in image compression as it attempts to reduce the dimensionality of an image while maintaining as much of the information, or energy, as possible. For an input $M \times N$-dimensional image A_{mn}, the discrete cosine transform produces the output $P \times Q$-dimensional image B_{pq} with the following transformation [38]:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} cos \left[\frac{\pi(2m+1)p}{2M} \right] cos \left[\frac{\pi(2n+1)q}{2N} \right]$$

where

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, \, p = 0 \\ \sqrt{\frac{2}{M}}, \, 1 \leq p \leq M-1 \end{cases}$$

and

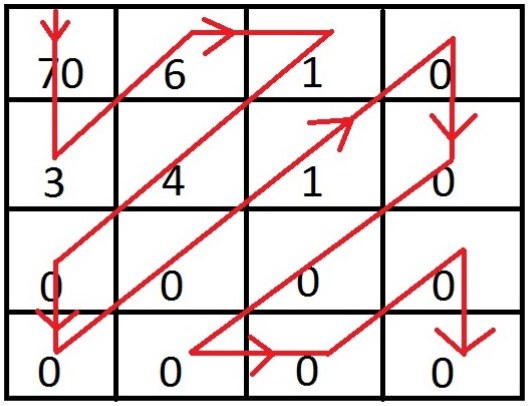
$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, \, q = 0 \\ \sqrt{\frac{2}{N}}, \, 1 \leq q \leq N-1 \end{cases}$$

After B_{pq} has been found, the vector size is reduced by retaining only the most significant coefficients. After the DCT, these coefficients are contained in the top right-hand corner of B_{pq} and are extracted using a "zig-zag" pattern as shown in Figure 2.6. Usually the first coefficient is not used, as it is just the average image intensity.

2.2.4 Visemes and HMMs

Coarticulation affects visemes as well as phonemes. Thus, HMMs are also used to account for the effect of the previous viseme on the current one. After feature vectors have been extracted and "ground-truth" viseme label files have been made from the phoneme label files and a phoneme-to-viseme map, a HMM can be trained to recognize visemes. The HMM is trained on a set of already-known viseme sequences and label files, and its parameters are changed to approximately solve Rabiner's third problem, described in Section 2.1.3. This leads, in theory, to a recognizer capable of VSR, sometimes called automatic or computer lipreading. The recognizer could then be combined with an audio recognizer to create a joint audio-visual speech recognizer. The various methods of creating joint audio-visual speech recognizers are discussed in Section 2.3.

Figure 2.6: DCT "Zig-Zag" Coefficient Extraction Pattern



Output vector:
[70, 3, 6, 1, 4, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]

2.2.5 Angled Views for Lipreading

The ideal angle for viewing a speaker's face for lipreading has not been settled upon. A frontal view offers information about mouth width, but a profile view offers information about lip protrusion. Angles in between have also been tested in an attempt to find a "sweet spot" angle, offering the best of both worlds. The debate also takes practical use cases into account. For example, a recognizer designed to be used with laptop webcams should be trained on frontal views. A recognizer to be used by car drivers, however, might have to work with angled views, as the camera cannot block the driver's view.

An early investigation into the performance of profile versus frontal views was undertaken by Lucey and Potamianos [52] in 2006. Using DCT features, they found that frontal views significantly outperformed profile views on a visual-only word-recognition task. Kumar et al. [45] later compared profile and frontal views using shape-based features, and found that profile views gave the highest performance. In 2010, Pass et al. [63] investigated the viability of pose-invariant visual only speech recognition on a speaker-dependent, isolated-digit database using DCT features. Using recorded angles from 0° - 90°, they found that the best set of DCT coefficients to retain to be recognized by a frontal-view trained recognizer were angle-dependent. In 2012, Lan et al. [48] put together a small, single-speaker database with camera angles of 0, 30, 45, 60 and 90 degrees. Using an AAM, they found that the view from 30 degrees gave the highest performance. Based on these results, it was decided to record speakers in the TCD-TIMIT database from two views: frontal and 30 degrees.

2.3 Audio-Visual Speech Recognition (AVSR)

As mentioned previously, certain words cannot be distinguished by lipreading. For example, the words "mat" and "bat" look visually identical. This is why VSR is usually talked about as an aide to ASR. Visual features are not affected by a decrease in the audio channel's signal-to-noise ratio (SNR), so they can help a recognizer be more robust in noisy environments. Methods of integrating an audio and visual speech recognizer are divided into three categories: early, intermediate and late-stage integration. These categories have been explained by Potamianos et al. [69].

2.3.1 Early Integration

Early integration is also known as feature fusion because extracted audio feature vectors are "fused" with extracted visual feature vectors. A single multi-stream HMM is trained on the resulting vectors. This kind of integration requires the audio and visual feature vectors to be synchronous, i.e. for each audio vector extracted at time t there is a corresponding visual vector also extracted at time t.

2.3.2 Intermediate Integration

Intermediate integration allows audio and visual states to be asynchronous within a model, but forces them to be synchronous at the model's boundaries. Composite HMMs called product or coupled HMMs are used. The feature vectors can either be "fused" audio-visual feature vectors or separate vectors. Intermediate integration allows for the natural asynchrony present between phonemes and visemes. For example, the /g/ phoneme in the word "segment" has no clear viseme associated with it, and depending on the speed of pronunciation, several different visual features may be seen in the duration of the phoneme.

2.3.3 Late Integration

Late integration is also known as decision fusion. The audio and visual HMMs are kept separate, as are the observed feature vectors. Then, a set of n-best output sequences is returned from the audio and visual recognizers. Using appropriate weighting, a combined likelihood of each output sequence is found, and the overall most likely sequence is chosen as the correct one.

2.4 Human Lip Reading

Lip reading is a skill subconsciously used by most humans, especially when it is difficult to hear a speaker. Actually, the term "speech reading" is preferred by experts [79] as many other details about the speaker (e.g. eyes, mood, context etc) are also used by speech readers. Some deaf societies offer classes teaching people how to improve their speech reading. There are also

organizations offering professional training to people to make them easier to lipread. Individuals who have completed this training are called lipspeakers, and can be used as translators for speech readers [15]. An information leaflet published by the Victoria Deaf Society [79] lists some of the things that their lipspeakers are trained to do to make their visemes more distinctive:

1. Increase the duration of the sound /m/ to distinguish it from /p/, /b/
2. Increase the duration of the sound /n/ to distinguish it from /t/, /d/
3. Place the tongue between the upper and lower teeth to clarify /th/
4. Spread the lips, clench the teeth firmly and grin to indicate /s/, /z/
5. Bite the lower lip with the upper teeth to indicate /f/, /v/
6. Move the jaw down briefly while producing /k/, /g/
7. Shrug the shoulders briefly during the inhalation preceding /h/
8. Increase or decrease height/width of the lip opening while producing vowels

From the list, it appears that rather than exaggerating mouth movements, lipspeakers attempt to make their mouth movements more distinctive. Since the purpose of the TCD-TIMIT database is to help investigate visual features for VSR, it was decided to include some lipspeakers in the database along with non-trained speakers. The lipspeaker data is introduced in Chapter 5.

2.5 Summary

This chapter looks at the concepts behind audio and visual speech recognizers. To design a database suitable for training an audio, visual or joint audiovisual speech recognizer, it was important to first review these concepts. Also, the methods used to obtain the TCD-TIMIT baselines were chosen based on the background in this section.

For audio-only recognizers, the sub-word unit known as the phoneme is described in Section 2.1.1. Audio parametrization techniques for speech are described in Section 2.1.2. Examples of how phonemes are defined and used in tests on the TIMIT speech corpus are looked at in Section 2.1.4.

For visual speech recognizers, visemes are discussed in Section 2.2.1, and phoneme-to-viseme maps in Section 2.2.2. Feature extraction techniques for video of speech are described in Section 2.2.3. The use of angled views of a speaker for visual speech recognition is discussed in Section 2.2.5. Based on this information, speakers in the TCD-TIMIT database were recorded from an angle as well as from the front (see Section 3.1.3).

For joint audiovisual speech recognizers, integration methods are discussed in Section 2.3. Finally, a short introduction is given to professional "lipspeakers", teachers who teach people how to visually recognize speech (i.e. "lipread") in Section 2.4. Three such lipspeakers were recorded as a secondary part of the TCD-TIMIT database. The recording of the database, post-processing and obtaining of baselines are discussed in Chapter 3.

3

The TCD-TIMIT AVSR Database

3.1 The Recording Process

3.1.1 Speaker Scripts

From the beginning, the TCD-TIMIT corpus introduced in this work was intended for Audio-Visual Continuous Speech Recognition (AVCSR). The first decision to be taken when constructing an AVCSR corpus is on the speech material to use. Four of the databases in Table 1.1 have used groups of sentences from TIMIT, the audio-only CSR database discussed in Section 2.1.4, as the TIMIT sentences were designed to include as many phoneme pairs as possible [46]. TIMIT's sentence list consists of two "SA" sentences, designed to highlight a speaker's accent, 450 "SX" sentences, hand-designed to include as many different pairs of phonemes as possible, and 1890 "SI" sentences, picked from playwrights' books to include phoneme pairs in "unusual" contexts. The 630 speakers in TIMIT said 10 sentences each: the two SA sentences, five SX sentences and three SI sentences. The SI sentences were unique to each speaker (1890/3 = 630), but each group of five SX sentences was spoken by seven speakers (450/5 = 90, 630/90 = 7). The TIMIT sentences were chosen as the speech material for the TCD-TIMIT corpus. The intention for TCD-TIMIT was to have speakers say much more than 10 TIMIT sentences each. This motivated an examination of the phonetic balance of groups of sentences picked from TIMIT. If a TCD-TIMIT speaker was given some number of sentences from TIMIT to say, would they adequately cover all phonemes and visemes? To check this, a script originally written by Andrew Hines [34] was modified to find the number of occurrences of all phonemes in the full

TIMIT corpus. Phoneme occurrence rates for a number of different groupings of sentences were then checked against these statistics. For example, to give each TCD-TIMIT speaker 12 TIMIT speakers' scripts to say, TIMIT's 630 speaker scripts were first split into groups of 12. Each group's phoneme statistics were then checked against the overall occurrence rates to make sure that no group had far too many or too few instances of a phoneme.

Eventually, after trial runs with early volunteers and script sizes of 50 to 200, a script size of 98 sentences per speaker was decided upon. Feedback from the early volunteers suggested that this was the most sentences potential volunteers could be asked to say without discouraging them from volunteering. To make scripts of 98 sentences, the 630 TIMIT speaker scripts were split into groups of 12. While it is true that each TIMIT speaker says 10 sentences, the first two sentences are always the same (SA1 and SA2) so these were only included once in the 98-sentence scripts. This left 8 sentences to be contributed by each of the twelve TIMIT speakers (96 sentences), plus SA1 and SA2.

Splitting the TIMIT speakers into groups of 12 involved more than just grouping them randomly. Since the SX sentences were all said by 7 speakers each, if the speakers were grouped randomly it was likely that there would be duplicate SX sentences in some of the groups. Luckily, the SX sentence distributions in TIMIT follow some sort of order, with a few exceptions. They were split into groups of 5, so (in most cases) if a speaker said one sentence from a group, they would also have said the others. For example, the group of SX sentences SX4, SX94, SX184, SX274 and SX364 were all said by speakers MDRB0, FEDW0, MGLB0, MJDH0, FJEM0, MRJS0 and MTDT0. So, any grouping of 12 TIMIT speakers should only have one of these speakers in it. Using this knowledge, the TIMIT speakers were arranged in a spreadsheet in order of lowest SX sentence (SX3, SX4 etc etc), where the groups of 7 were easy to see. Then, to get unique groups of 12, this list was traversed repeatedly, removing one speaker from each group until 50 groups of 12 speakers had been made. This left a remainder of 30 unused TIMIT speakers.

Unfortunately, TIMIT didn't always abide by the groups of SX sentences. For example, the SX group containing sentences SX17, SX107, SX197, SX287 and SX377 was said by 6 speakers, but the 7th speaker said SX17, SX77, SX167, SX257 and SX437. There are numerous examples of this, especially around the groups at the bottom of the list, and there is no obvious reason why. As a result, once the groups of 12 had been extracted, they were then individually checked for duplicate sentences. If a duplicate was found, the TIMIT speaker responsible was removed from the group, returned back to the list, and the next as-of-yet unused speaker was chosen instead. Eventually, 50 scripts of 98 unique sentences were made.

After creating the scripts, the next stage was to choose the recording location, equipment and find volunteers to record.

3.1.2 Equipment

The research group already owns a pair of Sony PMW-EX3 cameras, so these were used to record the database. The PMW-EX3s can be synchronized, making time-alignment of the two camera views much easier after recording. They also have inputs for recording external microphones, meaning that the audio and video streams were also synchronized. The cameras record 1920x1080-pixel frames at 30fps. The cameras can record at 50fps, but this is only available for 1280x720-pixel frames. A study by Saitoh and Konishi [72] found that a higher frame rate did not lead to additional improvements in recognition scores. As a result, higher resolution frames were chosen over a higher frame rate.

For the microphone, a wireless clip-on electret mic was used. The mic was a Shure PG185, with a PG1 transmitter and a PG4 receiver. The room was not soundproof, so other mics were picking up noise from outside. The clip-on had some high-frequency hiss, but picked up the least outside noise. The volunteers clipped the mic on themselves, with the instruction that it be below the chin, close to the mouth and angled towards the mouth. Audio levels were then checked. Another mic, a shotgun-type condenser, was also recording, but its audio contained a lot more noise and was not used.

3.1.3 Recording Setup

The setup of the room is shown in Figures 3.1, 3.2 and 3.3. One camera recorded the speaker from directly in front, while the other recorded at an angle of 30° to the speaker's right. Both cameras were zoomed in to contain only the speaker's head, shoulders and the greenscreen in the shot. The light used was a 500W tungsten photoflood with a 60x60cm softbox. The light was placed directly behind and above the front-facing camera and angled slightly downwards (about 15 degrees) at the subject. This position was chosen to get even lighting across the subject's face, illuminate as much of the face as possible (without blinding the subject!) and eliminate shadows on the greenscreen behind the subject.

3.1.3.1 Greenscreening (Chroma Keying)

Leaving the opportunity to greenscreen subjects at a later date was seen as an easily-obtainable extra feature for the database for only a little extra work. Most professional greenscreening involves lighting the subject and the screen separately, but with a diffuse enough light source and enough distance between the subject and the screen, it can be done to a reasonable standard using the same light source for both. Due mainly to space limitations, this was the approach used. Another common technique used in TV-standard greenscreening is to use a blue screen if the subject has blonde hair, but hanging a different screen was far too impractical for the small additional benefit. Volunteers were asked not to wear green clothing to recording sessions.

Figure 3.1: View of setup from behind cameras

3.1.4 Volunteer Recruitment

It takes roughly 25 minutes for a volunteer to record 98 TIMIT sentences. Once that recording session length had been settled upon, it was decided to compensate the volunteers for their time. Volunteers were given €5, refreshments and were entered into a prize draw for an Amazon Kindle or a set of Sennheiser headphones. This greatly helped the recruitment process. Posters were put up around campus advertising the rewards. An advertisement was put on TCD's online bulletin board. Several lecturers were kind enough to allow a short presentation about the database to be given at the start of one of their lectures, after which a sign-up sheet was passed around the lecture hall (this was the most successful recruitment tactic). Advertising was also done on several TCD-related Facebook groups (with the permission of the group owners). Volunteers were given a contact email address, and suitable dates and times for their recording sessions were worked out by email.

To recruit lipspeakers for the second part of the database, DeafHear.ie was an invaluable contact. DeafHear.ie is an organization which provides services for the deaf in Ireland and promotes deaf awareness. They referred three professional lipspeakers who agreed to be part of the database.

Figure 3.2: View of setup from behind speaker

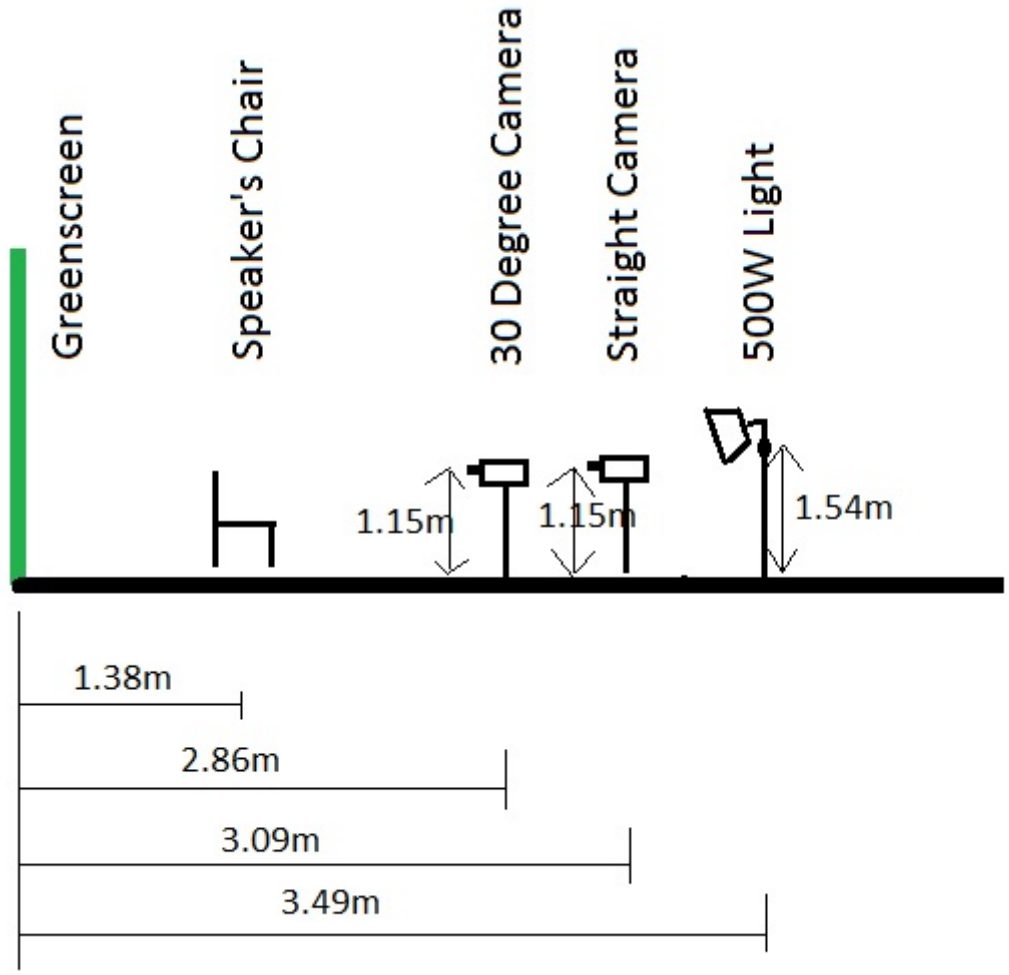
3.1.5 Typical Recording Session

A typical recording session played out as follows: The speaker would arrive and be given a short overview of what was involved. They would be seated, mic'd and the cameras would be set up. Audio levels, lighting and camera parameters would be checked. The speaker would be told to try their best to start and end each sentence with their mouth closed and to leave two seconds of silence between each sentence. If the speaker made a mistake during a sentence attempt, they were directed to leave two seconds of silence before attempting it again. The only people present during the recording session were the speaker and the recorder. It was the recorder's job to advance the speaker's "teleprompter" (an external monitor hooked up to a laptop controlled by the recorder). Since the recorder could also see the sentences, it was their job to catch any pronunciation mistakes made by the speaker. After the recording process, the volunteers signed a consent form. The consent form is given in Appendix C.

3.2 Post-Processing

The Sony PMW-EX3s record video in MPEG-2 Long GoP format, and audio in stereo PCM, onto SxS cards, Sony-designed flash memory cards compliant with the ExpressCard standard.

Figure 3.3: Equipment Positional Measurements



The video and audio are both wrapped in an MP4 container. The cameras cannot record single files larger than 4GB, so long recordings are technically split into two or more MP4 files connected by a third file explaining the structure. With a downloadable Sony driver, the SxS cards can be recognized through a laptop's ExpressCard slot.

A minute of footage from one of the cameras is roughly 260MB. The average footage length was roughly 15 minutes. After recording 59 speakers, 450GB of raw footage had been recorded.

The goal of post-processing was to end up with a set of video and audio clips, one for each speaker sentence, preferably in the exact same format as that taken from the cameras, or at the very least with no loss of quality. It was not desirable to clip the footage manually due to the time and potential for error involved. Also, while Adobe Premiere (the editor that would have been used to clip the footage manually) had no problem reading in the Sony footage and recombining the two or more MP4 files that made up a long recording session, it could not output it in the same format. Several lossless codecs were tried, but the output file sizes were

too large to be practical. Attention then turned to ffmpeg [18], the command-line video editing tool. Since it was called from the command line, it could be automated, but unfortunately it could not process the Sony MP4 files correctly, due to an inability to wrap PCM audio into an MP4 container. However, it could wrap MPEG-2 video and PCM audio into a Matroska container (MKV), an open-source, free container format. Matroska videos can also be clipped by ffmpeg. Thus MKV became the eventual file format used. The only remaining problem was that only Adobe Premiere and Sony's own "Content Browser" software could recombine the multiple MP4 files from long recording sessions. In these cases, the "Content Browser" software was used to combine the MP4s and export them as large Material eXchange Format (MXF) files, which could then be converted to MKV files for clipping. An additional benefit of MKV files is that they can be read in by Matlab's VideoReader function.

3.2.1 Clipping the Footage

The purpose of clipping the footage was to create a clip from the straight and 30° camera for each sentence. Since the clip-on mic channel was the best-quality audio channel recorded, it was used as the audio for the 30° camera's footage as well. This meant synchronizing the 30° camera's footage with that of the straight camera. In addition to being able to read Sony's MP4 files, Adobe Premiere can read synchronization information between them, but as explained, cannot output the footage without transcoding. The workaround solution was to synchronize the videos in Premiere, export the synchronized audio, then use ffmpeg to combine that audio and the video from the 30° camera in an MKV file. In this way, the 30° camera's footage could be clipped in the same way as the straight camera. This means that each sentence clip from the straight camera has a corresponding clip from the 30° camera shot at the exact same time.

Once the audio was synchronized and extracted, a very basic "speech detector" function, based on energy thresholding, was used to find the beginning and end of speaker sentences in the audio. Since the speakers had been instructed to leave at least a second of silence between each sentence attempt, the detector's algorithm for finding speech events went as follows:

Step 1) Threshold the samples to make every sample below a certain amplitude equal to 0.

Step 2) Go through the samples, and the first time a non-zero sample is seen, treat it as the start of a speech event. Record its index.

Step 3) Now look for a run of a certain length of zeros (500ms was usually used), indicating that the speech event is over. Record the index of the last zero in the run.

Step 3) Check that the distance (in samples) between the speech start index and the last zero's index is greater than a certain amount of time (the shortest possible sentence was just over 1s long, so 1s was usually used). Also check whether there were at least a certain amount of nonzero values between the two indices (in case the "speech event" was just a few loud isolated noises).

Step 4) If both checks in Step 3 are positive, record the indices as sentence start and end times. Otherwise, discard them. Add 500ms to the start time to account for the 1st viseme being visible before the 1st phoneme is heard. Unless the end of the samples has been reached, return to Step 1.

The video files were then clipped with ffmpeg using the beginning and end times found by the speech detector. Unfortunately, the speech detector was not perfect. Failed sentence attempts as well as successful attempts were classed as speech events and clipped. If a speaker left a long pause at a comma, the sentence would be split into two clips. If the speaker spoke quietly and the sentence was short, sometimes the sentence was missed. Hence, every clip had to be manually checked to see if it was a valid sentence. Missing or incomplete sentences were manually re-clipped. The speech detector created about 90% of the clips correctly. Once all of the correct sentence clips had been identified and created, they were given their TIMIT codes. A corresponding audio clip was created for each video clip (using ffmpeg) for acoustic speech recognition tests. Also, the word-level and phoneme-level TIMIT label files (explained in Section 2.1.4) for each sentence were added to the directories with the sentence clips. These label files would later be used to create force-aligned label files for TCD-TIMIT (see Section 3.3). For SA and SX sentences, there were multiple TIMIT label files available to choose from, since more than one TIMIT speaker said these sentences. In these cases, the label file was arbitrarily chosen, since there was no clear way of knowing which label file would be "best" to use.

3.2.2 Database Structure

The volunteers' identification code format was chosen to make it as easy as possible to iterate through speakers. Initially, the speakers were simply going to have a double-digit number as their ID code, from 01-99 (it was not expected that more than 99 speakers would be recorded), but afterwards, it became useful to also have a way of quickly identifying whether a speaker was male or female, so the letter "M" or "F" was appended to each speaker's code to indicate this. This is similar to TIMIT's approach, which prepends an "M" or "F" to its speaker codes. One of the reasons TIMIT's naming convention ("M" or "F", followed by 3 characters of speaker's initials, followed by a digit to distinguish between otherwise duplicate codes) was not followed is that there are fewer speakers in TCD-TIMIT, so a shorter, digit-based code was easier to search through. Lipspeakers (discussed in Chapter 5) use a different format to distinguish them. The reason TIMIT's speaker codes were not re-used, as they are in VidTIMIT, is that TCD-TIMIT speakers say more than one TIMIT speaker's sentences, so there is no direct relationship.

3.2.3 TCD-TIMIT Speakers

In total, there are 62 speakers in TCD-TIMIT. The main "volunteer" part of TCD-TIMIT consists of 59 regular speakers (i.e. non-lipspeakers). Of these, 32 are male and 27 are female. 56 of the volunteers are between the ages of 18 and 29 (the remaining 3 are between the ages of 51

and 57). The secondary "lipspeaker" part of TCD-TIMIT consists of 3 professional lipspeakers. All three lipspeakers are female. The average age of the lipspeakers is 60, with a variance of 2.

There are 4 non-Irish accents in TCD-TIMIT. Speaker 27M has a Spanish accent, while 35M, 53M and Lipspeaker 3 have British accents. The rest of the accents are various Irish accents, the majority being "neutral" Dublin accents. Speakers were allowed to wear glasses, piercings and any hairstyle, the only request was that nothing green be near the head or shoulders. Speakers were instructed to speak in the most natural way possible, i.e. they had control over speed, pauses, intonation etc. The only difference between the recording sessions for the volunteers and lipspeakers was the length. The 3 lipspeakers said almost 4 times as many sentences as the volunteers.

3.3 Audio-Only Baseline

3.3.1 Phoneme-Level Label Files

Phoneme-level label files contain the list of phonemes in a sentence's audio clip, along with their start and end times. They are used as ground-truth for training and testing an acoustic speech recognizer. Using a phoneme-to-viseme map, they can also be converted into viseme-level label files and used to train and test a visual speech recognizer. TIMIT comes with word-level and phoneme-level label files for its sentences. However, as explained in Section 2.1.4, the original phoneme set was too large, so newer label files, whose phonemes had been mapped down to the reduced set in Table 2.2, were used here. The first method of obtaining label files for the TCD-TIMIT clips was to create them by performing forced alignment using the TIMIT label files.

3.3.2 Forced Alignment

TIMIT's time-aligned phoneme label files were created by hand, by phoneticians. Ideally, TCD-TIMIT's phoneme label files would be created in a similar manner, but this is painstaking, time-consuming work. To overcome this problem, researchers commonly make use of a technique called forced alignment to obtain time-aligned label files of acceptable quality ([4], [65], [78], [83], [35]). The two most widely-used toolkits for automatic speech recognition, HTK and CMU Sphinx, provide forced alignment capabilities.

The idea behind forced alignment is that if there are well-trained phoneme HMMs available, they can be arranged in such a way as to provide the maximum likelihood of producing some non-aligned observation sequence. The phoneme boundaries can then be taken as the boundaries between the HMMs. The key here is that the phonemes in the sequence, and their order, are known. Hence the HMMs to use, and order in which to arrange them, is known. The only missing information is the boundaries between the phonemes. An example of forced alignment can be seen in Figures 3.4, 3.5 and 3.6.

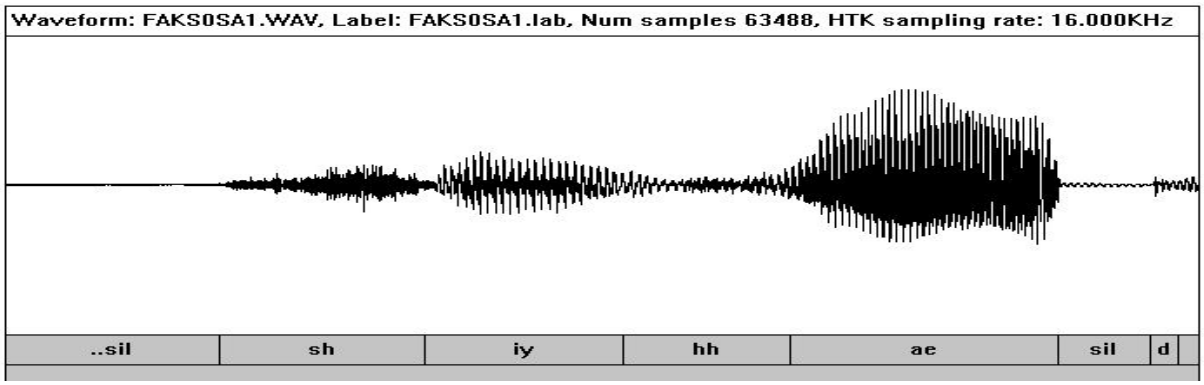


Figure 3.4: Snippet of a TIMIT speaker saying "She had", annotated with TIMIT's hand-aligned labels.

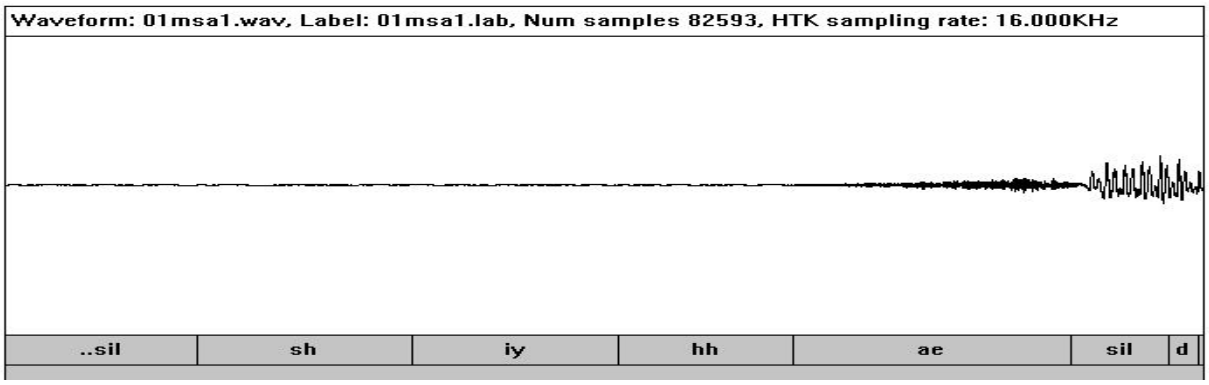


Figure 3.5: Applying the labels of Figure 3.4 to a TCD-TIMIT speaker's waveform of the same sentence. The order of the labels is still correct, but they no longer correctly mark the phonemes. Forced alignment attempts to remedy this.

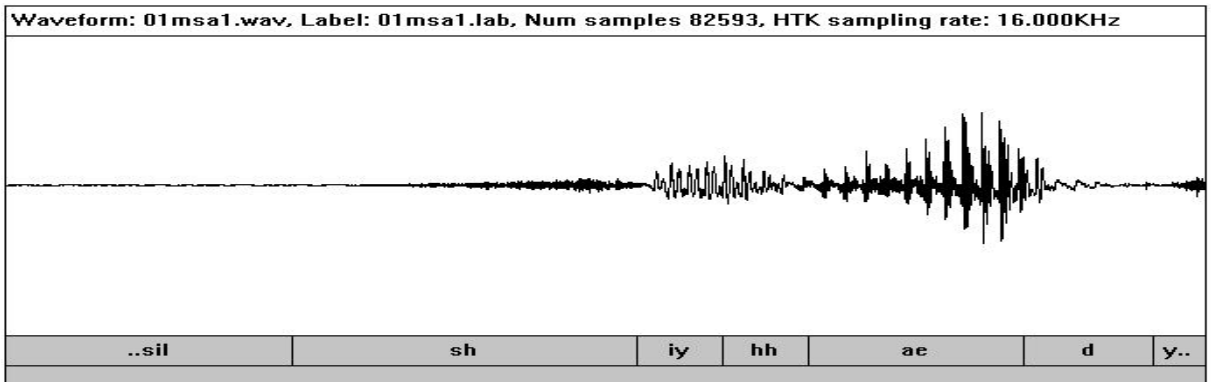


Figure 3.6: The resulting labels applied to the TCD-TIMIT clip after forced alignment. The labels now correctly mark the starts and ends of the phonemes.

Since the TCD-TIMIT speakers said sentences from the TIMIT corpus, TIMIT's phoneme-level label files for these sentences can be used for forced alignment. The only remaining requirement for forced alignment is a set of well-trained phoneme HMMs. To obtain these, the TIMIT corpus was used. A set of monophone HMMs was trained on TIMIT using its recommended training set (the 462 speakers from TIMIT's "TRAIN" section). After this, the HMMs were tested on TIMIT's recommended test set. This was done for two reasons: to make sure the models were adequately trained and ready to use for forced alignment, and also to have an acoustic baseline to compare to the TCD-TIMIT baseline later. The scores obtained on the test set of TIMIT were compared to monophone TIMIT scores in the literature (see Section 4.1.2) to check whether the models were adequately trained.

From these comparisons, the models were deemed to be adequately trained for forced alignment. The next step was to adapt the models to the TCD-TIMIT audio. This is done by using several passes of embedded re-estimation. During embedded re-estimation, the time information in phoneme label files is ignored. Only the order of the phonemes in the corresponding observation sequence is needed. For a given observation sequence, the embedded re-estimation procedure lines up the HMMs in the order given by the label file, and then adjusts their parameters to maximise the probability of outputting that sequence. A suitably-weighted fraction of the adjustments is then applied to the HMMs. The reasoning behind adapting before attempting to force-align using the HMMs is that the TCD-TIMIT audio does not sound like the TIMIT audio. It was recorded in a different environment, with different equipment and different speakers. Adapting the HMMs allows them an opportunity to account for these differences before forced alignment.

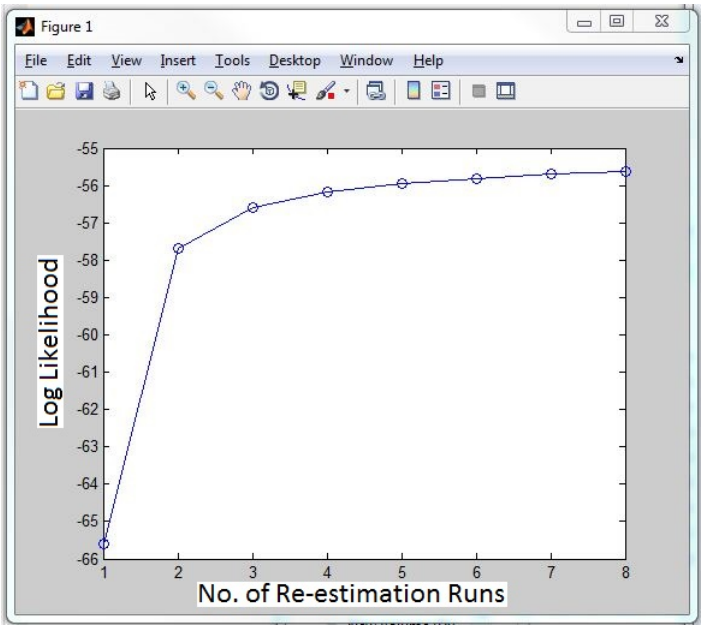
Adaptation was performed iteratively, with forced alignment attempted after each iteration. The overall log-likelihood of the alignments was recorded each time. As the HMMs became more adapted to the TCD-TIMIT audio, the log-likelihoods approached an upper limit. This can be seen in Figure 3.7. After getting close to the upper bound, the HMMs run the risk of being overfitted. For this reason, the force-aligned label files produced after 7 iterations of embedded re-estimation were chosen. Tests began on TCD-TIMIT using these label files.

HTK was used to perform forced alignment, as well as obtain the baselines discussed in Chapter 4. The specifics of how HTK was used to do this are described in Appendix B, so a quick run-through of the important HTK settings used in this chapter is given in Section 3.3.3.

3.3.3 HTK Settings for Forced Alignment and Testing

TIMIT's full "TRAIN" set (SA, SX and SI sentences) was used to initially train HMMs for forced alignment. When obtaining TIMIT baselines on the HMMs, TIMIT's full "TEST" set (SA, SX and SI sentences) was used. There are 4620 sentences in TIMIT's "TRAIN" set and 1680 sentences in TIMIT's "TEST" set. This is roughly a 73-27 train-test split. For the tests performed on TCD-TIMIT in this chapter, the main train-test split used is given in Table 3.1.

Figure 3.7: HMM Adaptation: Log Likelihood Versus Re-estimation Runs



This is a 66-34 split. Another train-test split was occasionally used to verify results. The reason 27M, 35M and 53M were not used is due to their non-Irish accents.

For the audio-only recognizers in this chapter, the audio files used were mono WAV files sampled at 16kHz. The feature files created from these WAVs were MFCC files with 12 coefficients as well as their 1st and 2nd derivatives. The configuration file used to create these files and the prototype HMM used to set up the phoneme HMMs are given in Appendix B. Models based on the prototype start off with one Gaussian mixture per state, but during training this was increased to 31 mixtures per state, with five embedded re-estimation runs after each increase. These numbers were high in order to be thorough and consistent with the training given to each recognizer, so they could be easily compared. The trained recognizers were always evaluated on monophone recognition performance.

Table 3.1: Main TCD-TIMIT train-test split used

TRAIN	01M, 02M, 03F, 04M, 05F, 06M, 07F, 08F, 09F, 10M, 11F, 12M, 14M, 16M, 17F, 18M, 19M, 20M, 22M, 26M, 32F, 33F, 38F, 39M, 40F, 41M, 42M, 44F, 45F, 46F, 48M, 49F, 50F, 52M, 56M, 58F, 59F
TEST	13F, 15F, 21M, 23M, 24M, 25M, 28M, 29M, 30F, 31F, 34M, 36F, 37F, 43F, 47M, 51F, 54M, 55F, 57M

3.3.4 Performance of Force-Aligned TIMIT Label Files

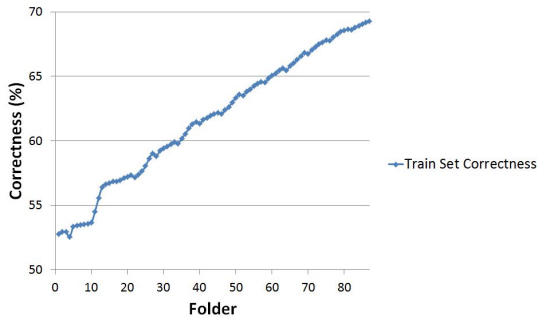
As explained in Section 3.3.2, a set of force-aligned label files for TCD-TIMIT was obtained from the TIMIT label files. However, this approach was not perfect. The time information had been changed to fit the TCD-TIMIT data, but the phonemes never changed. TIMIT used 630 American English speakers, some with very strong accents. This meant that the phonetic transcriptions were sometimes incorrect when used for the TCD-TIMIT speakers. This could be seen by manually inspecting a TCD-TIMIT audio file and its label file in HTK's "HSLab" program.

In an attempt to investigate whether the force-aligned label files were adequate, their performance was evaluated on a speaker-independent monophone recognition task. A recognizer was trained and tested on TCD-TIMIT's audio (train-test split given in Section 3.3.3), using the force-aligned files. The correctness and accuracy scores were recorded. Another recognizer was trained and tested on TIMIT (train-test split given in Section 3.3.3), and its correctness and accuracy scores were recorded for comparison. TCD-TIMIT's scores themselves were not expected to be higher than or even the same as TIMIT's, but if they were within roughly 10% and exhibited the same characteristics, it would suggest that the label files were adequate. The results from both recognizers are given in Table 3.2.

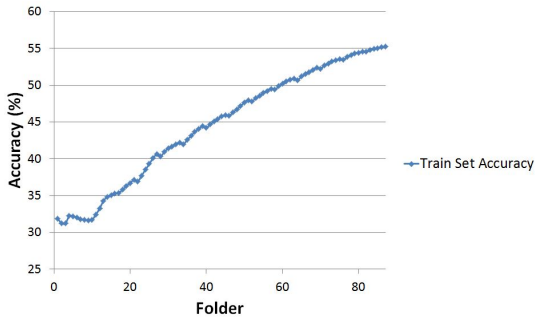
Table 3.2: Monophone recognition results from TIMIT and TCD-TIMIT using force-aligned TIMIT labels

	TIMIT		TCD-TIMIT	
	train set	test set	train set	test set
%correct	76.54	72.53	69.29	61.53
%accuracy	64.04	57.94	55.26	44.50

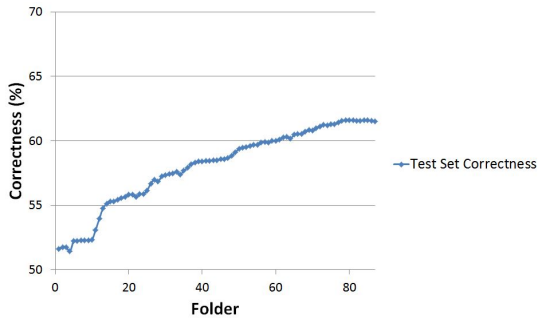
The TCD-TIMIT results in Table 3.2 were lower than expected, over 10% lower than the TIMIT results. More importantly, the gap between TCD-TIMIT's train and test set results is twice as wide as TIMIT's, suggesting that the recognizer was not as robust to new data. It was possible that the TCD-TIMIT recognizer had been overtrained or had passed its optimal number of mixtures. To rule this out, recognition scores were recorded after each change to the HMMs, and graphed. Ideally, the graphs should show the scores increasing towards an upper bound as the HMMs are trained. The graphs can be seen in Figure 3.8. The "Folder" parameter in the graphs represents each incremental change made to the HMMs. When the number of mixtures was increased or a re-estimation was performed, the resulting HMMs were stored in the next highest folder. Small dips can be seen in the graphs (e.g. at folder 22) each time more mixtures were added to the HMMs. Each graph shows correctness and accuracy increasing towards an upper bound, as expected. The graphs show that the HMMs were not overtrained. Audio quality had been assessed as adequate by manual inspection, so the next most likely cause for



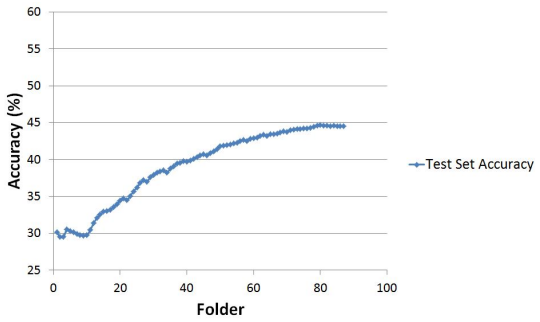
(a) Train Set Correctness (Y axis) vs HMM Modification (X axis, represented by "Folder")



(b) Train Set Accuracy (Y axis) vs HMM Modification (X axis, represented by "Folder")



(c) Test Set Correctness (Y axis) vs HMM Modification (X axis, represented by "Folder")



(d) Test Set Accuracy (Y axis) vs HMM Modification (X axis, represented by "Folder")

Figure 3.8: TCD-TIMIT train and test-set scores increasing as the HMMs are refined (through mixture increases and embedded re-estimation runs).

the low scores was thought to be either the relatively small number of TCD-TIMIT speakers or the accuracy of the force-aligned label files. The number of speakers could not be changed, so attempts were made to improve the accuracy of the force-aligned label files.

3.3.5 Forced Alignment with P2FA

The "Penn Phonetics Lab Forced Aligner" (P2FA) is a tool developed by Yuan and Liberman [90] during their research into identifying speakers in the SCOTUS (Supreme Court of the United States) corpus. The corpus contains over 9000 hours of oral arguments from the Supreme Courts. With 25.5 hours of manually-aligned training data, they built a recognizer capable of force-aligning long (1hr+) segments of speech. The majority of the differences between the manually and force-aligned boundaries were under 50ms. With this in mind, it was decided to try obtaining force-aligned files for TCD-TIMIT using P2FA to see if they were more accurate than the force-aligned TIMIT files.

The P2FA tool consists of a set of HTK-compatible models and other files, and a Python

script which sets up and calls HTK in forced-alignment mode. The Python script produces a force-aligned phoneme-level file for a given speech clip and its word-level label file (it disregards time information). It does this by using a pronouncing dictionary, specifically the CMU pronouncing dictionary [11], which contains transcriptions for over 125000 words. The pronouncing dictionary is used to get the transcriptions for the words in the word-level file. The phonemes are then force-aligned as they were with the TIMIT files. The major difference is that since the pronouncing dictionary can have multiple transcriptions for the same word, HTK tries all of these and picks the most accurate one. This means that different pronunciations can be accounted for by having multiple transcriptions for words. In contrast, using the force-aligned TIMIT files meant that the original TIMIT speaker's pronunciations decided the phonemes present, and there was no easy way to change them.

Table 3.3: TIMIT Phoneme Set (of Lee and Hon [49]) vs CMU Phoneme Set

TIMIT	CMU	TIMIT	CMU
aa	AA0, AA1, AA2	ae	AE0, AE1, AE2
ax-h, ah	AH1, AH2	ax	AH0
ao	AO0, AO1, AO2	eh	EH0, EH1, EH2
ix, ih	IH0, IH1, IH2	ey	EY0, EY1, EY2
ay	AY0, AY1, AY2	iy	IY0, IY1, IY2
oy	OY0, OY1, OY2	aw	AW0, AW1, AW2
ow	OW0, OW1, OW2	ux, uw	UW0, UW1, UW2
uh	UH0, UH1, UH2	axr, er	ER0, ER1, ER2
eng, ng	NG	sh	SH
ch	CH	jh	JH
zh	ZH	y	Y
dh	DH	p	P
b	B	em, m	M
dx, t	T	d	D
en, nx, n	N	k	K
g	G	s	S
z	Z	f	F
v	V	w	W
el, l	L	r	R
th	TH	hv, hh	HH
#h, h#, epi,		q	removed and time added to following phoneme
pau, kcl, tcl,	sil		
pcl, gcl, dcl, bcl			

A very small number of words in the TIMIT sentences were not present in the CMU dictionary. Since there were only a few of them, the missing words were manually given CMU-compatible transcriptions, following the rules in Table 3.3. In cases where stress indicators were needed, similar words were consulted for a suitable indicator to apply. Finally, the newly-transcribed words were inserted into the CMU dictionary.

P2FA's Python script needed a few slight modifications to run in Windows, as it was developed for Linux machines. Calls to Unix utilities cp, rm and cat were changed to their Windows equivalents. Calls to the SoX sound processing program were replaced with calls to ffmpeg. Finally, the script was ready to run. Since it only processes one sound clip at a time, a second wrapper script was written to run it for each TCD-TIMIT clip. Each transcription was added to one large text file.

After the transcriptions had been obtained, they needed to be mapped back to TIMIT-compatible phonemes using the reduced TIMIT phoneme set of Table 2.2. This was to make it easier to compare results using P2FA transcriptions to results using TIMIT label files, and also to make it easier to use the phoneme-to-viseme maps, which were based on the reduced TIMIT phoneme set. The map of Table 3.3 was used for this purpose. The only TIMIT phoneme that did not have an equivalent in the CMU dictionary was /dx/, so this phoneme was ignored during the reverse mapping and not used afterwards. Other than that, the only other issue was the stress indicators in the CMU vowels, which were simply discarded. After the reverse mapping, the file containing the transcriptions was turned into a HTK-compatible MLF (Master Label File).

Another recognizer was then trained using the train-test split of Table 3.1. Table 3.4 shows the results.

Table 3.4: TCD-TIMIT recognition results on force-aligned TIMIT labels and P2FA labels

	Force-aligned TIMIT labels		P2FA labels	
	train set	test set	train set	test set
%correct	69.29	61.53	73.22	63.39
%accuracy	55.26	44.5	58.56	45.24

As can be seen in Table 3.4, the gap between the train and test set results widened using the P2FA transcriptions. The training set's accuracy improved by 3%, but the test set scores did not improve significantly. This may be due to the fact that the phonemes that are labelled incorrectly in the force-aligned TIMIT transcriptions are consistently incorrect. With the start and end times seemingly accurate in both sets of transcriptions (this was verified by looking at a sample set of files manually), this would simply mean that the phonemes are just being called different names in the TIMIT transcriptions. Also, most words have the same transcription in

Figure 3.9: Similarity between force-aligned TIMIT and P2FA transcriptions

```
====================== HTK Results Analysis =======================
  Date: Fri Aug 02 11:58:41 2013
  Ref : P2FA_TCD_nodub.mlf
  Rec : TCDAVSRresults_r96.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.17 [H=10, S=5772, N=5782]
WORD: %Corr=80.58, Acc=68.52 [H=165020, D=10710, S=29069, I=24700, N=204799]
===================================================================
```

the CMU dictionary and TIMIT. To check this, HTK's results checking program was used to compare the force-aligned TIMIT files to P2FA output, to see how similar they were. The result is given in Figure 3.9.

According to Figure 3.9, both sets of transcriptions are indeed quite similar, with 68.52% of phonemes the same. Another idea was formed: the CMU dictionary had multiple pronunciations for some words, but they were all based in American English. There was no allowance for Hiberno-English, particularly Dublin accents. If Hiberno-English-influenced transcriptions were added to the dictionary, perhaps it would make the transcriptions more accurate and bring the recognition scores closer to those obtained on TIMIT (see Figure 3.2).

Appendix A describes some of the phonetic traits of Hiberno-English. Based on these traits, a set of rules was created and applied to the CMU dictionary to add Hiberno-English transcriptions of words that fell under one or more of the rules. After this process, over 29000 extra transcriptions had been added to the CMU dictionary. The P2FA script was run again and new transcriptions were created. To see how many cases this influenced, the new transcriptions were compared to the old ones via HTK's results analyser. The results of the comparison are given in Figure 3.10. The results from a recognizer trained on the new transcriptions are given in Table 3.5.

Table 3.5: Monophone recognition results using Pre and Post Hiberno-English P2FA transcriptions

	Without Hiberno-English		With Hiberno-English	
	train set	test set	train set	test set
%correct	73.22	63.39	73.36	63.63
%accuracy	58.56	45.24	58.66	45.46

Figure 3.10: Similarity between P2FA transcriptions Pre and Post Hiberno-English additions

```
====================== HTK Results Analysis =======================
  Date: Fri Aug 02 11:08:49 2013
  Ref : P2FA_TCD_nodub.mlf
  Rec : P2FA_TCD_wdub.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=43.27 [H=2502, S=3280, N=5782]
WORD: %Corr=97.57, Acc=97.33 [H=199827, D=1672, S=3300, I=497, N=204799]
===================================================================
```

Figure 3.10 shows that only a small number of new pronunciations were introduced. The results in Table 3.5 support this. At this point however, a new trend was noticed. Looking at speaker-by-speaker recognition results from HTK, there seemed to be a correlation between a speaker's amplitude and their recognition score (higher=better). Until this point, no normalization had been applied to the volume of the speaker clips. To be thorough in investigating all possible explanations for the low recognition results, the audio clips had to be normalized.

3.3.5.1 Normalizing the Audio Clips

There are several ways of normalizing audio clips, each with advantages and drawbacks. One way is to normalize based on the highest sample peak in the clip (peak normalization). Since it is usually not desirable to have any peak exceed (clip) the new threshold, enough gain is applied to the highest peak to bring it to the threshold. This same gain is then applied to the rest of the samples. Depending on the difference between the amplitude of the highest peak and the average sample, this might not have the desired effect. In that case, another normalization method is loudness normalization. This is meant to bring the average loudness to a certain level. For example, Root Mean Square (RMS) normalization measures the average power in the clip, and applies gain to the samples to bring the RMS value to a certain level. Unlike peak normalization, this is a nonlinear operation.

While RMS normalization would be preferable to peak normalization, it does not take the frequencies contributing to the power into account. If some clips have more noise, or less silence, their power will be higher and so their gain will be lower. For this reason, a different metric for comparing the loudness of the clips was used - a metric developed by the International Telecommunications Union (ITU) and known as Loudness Units relative to Full Scale (LUFS) (sometimes called Loudness K-weighted relative to Full Scale (LKFS)). The LUFS scale takes human perception of loudness, periods of silence and other factors into account. Its implementation is described in the ITU's recommendation paper BS.1770 [5]. LUFS have been adopted by the European Broadcasting Union (EBU) as a standard.

Adobe Audition has an implementation of the LUFS standard which allows clips to be analyzed and normalized to a certain LUFS level. Using Audition, the average loudness of the original TIMIT clips was found to be -21.3LUFS. The TCD-TIMIT audio clips were then normalized to this level.

After the clips had been normalized, a recognizer was trained and tested on the force-aligned TIMIT label files again. The results are given in Table 3.6. There was no significant improvement in the results from normalization. At this point, the impression was that the limit of a speaker-independent monophone recognizer trained on TCD-TIMIT may have been reached. The 56 speakers used for training and testing meant that 5488 sentences were used. While TIMIT only has 6300 sentences, these come from 630 speakers, which gives it an advantage when training a speaker-independent recognizer. The purpose of obtaining speaker-independent audio-only recognition results on TCD-TIMIT was mainly to have an easily-verifiable baseline to go with the database, and this was done. The purpose of comparing the results to those obtained on TIMIT was to compare to an "ideal baseline". Since TIMIT's label files are hand-aligned, TCD-TIMIT scores of a relatively similar nature would have supported the case that the force-aligned transcriptions were similarly accurate. Unfortunately, this cannot be claimed, but based on the results obtained, and manual checking of a sample set using HTK's "HSLab" tool, the transcription results were deemed acceptable, and work could commence on obtaining the visual-only and audio-visual baselines.

Table 3.6: Monophone recognition results from pre- and post-normalized TCD-TIMIT

	Pre-Normalization		Post-Normalization	
	train set	test set	train set	test set
%correct	73.36	63.63	72.48	63.06
%accuracy	58.66	45.46	57.94	45.06

3.4 Visual-Only Baseline

With the transcription files from P2FA and the video clips, a visual-only baseline could now be obtained. The first step was to use a phoneme-to-viseme map to convert the phoneme-level P2FA transcriptions to viseme-level transcriptions. The map used was the map by Jeffers and Barley, given in Table 2.4. The next step was to get visual feature files from the video clips. These were extracted using Cappelletta's method [7], briefly described in Section 2.2.3.1. Using slightly modified versions of Cappelletta's scripts, DCT feature files were created for each clip.

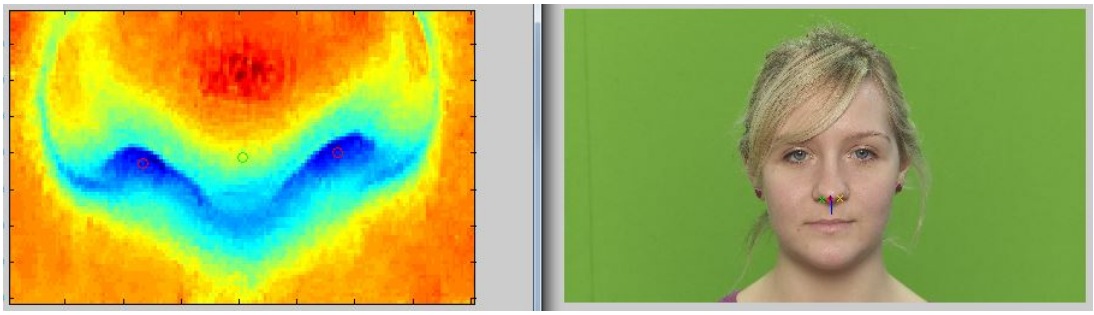


Figure 3.11: Example of correctly-tracked nostrils. The green "x" should mark the left nostril and the yellow "x" should mark the right nostril. The red dot should mark the halfway point between the nostrils.

3.4.1 Region of Interest (ROI) Extraction

VidTIMIT, the database Cappelletta worked with in his thesis [7], has 430 clips, each clip containing on average 120 512x384-pixel frames at 25FPS. The volunteer section of TCD-TIMIT, on the other hand, contains 5782 clips, each clip containing on average 150 1920x1080-pixel frames at 30FPS. Some hard-coded parameters in Cappelletta's scripts had to be changed to account for the different frame sizes. Extracting ROI subimages from the TCD-TIMIT clips using Cappelletta's scripts took roughly 45 seconds per clip, i.e. 72 hours for the 59 speakers.

On running his scripts on VidTIMIT, Cappelletta noted that the nostril tracking stage of the algorithm worked for 74.2% of his footage, and that the mouth detection stage always succeeded if the nostril tracking was successful. The most common problems he found were nostril occlusions, dark skin and moustaches. With this in mind, care was taken when recording TCD-TIMIT to avoid nostril occlusions wherever possible. As a result, nostril tracking worked on about 85% of the TCD-TIMIT footage. Unfortunately, unlike Cappelletta's finding, mouth detection did not work on all of the clips where nostril tracking was successful. The vast majority of the nostril and mouth failures were due to similar problems and contained within a small group of speakers: 13F, 25M, 26M, 29M, 30F, 34M, 40F, 52M, 56M and 57M.

When a clip failed for Cappelletta, he resorted to manually tracking the nostrils himself for each frame in the clip, and using those points for the mouth detection stage. Even though there was a lower percentage of failures in the TCD-TIMIT footage, manually tracking each clip would have meant tracking around 900 clips. Since most of these clips were from a small group of speakers, it was deemed quicker and more practical to modify the algorithm for problematic speakers. The modifications employed are given in Table 3.7.

Using these modifications, the eventual number of clips that had to be tracked manually was reduced to just 7. In each of the 7 clips, the speaker had moved their head rapidly at some point, losing the nostril detector. It took roughly 5 minutes to track a clip manually.

With the ROIs for each clip extracted, DCT feature files were created. Following the results

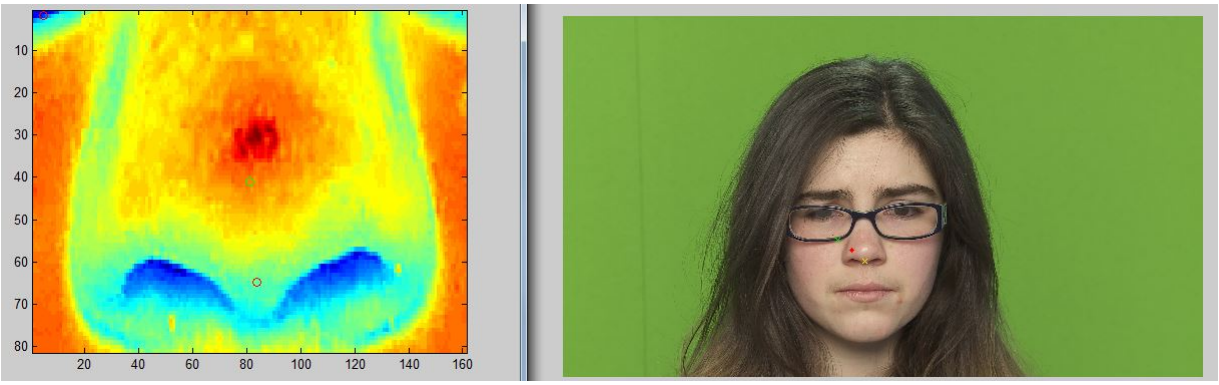


Figure 3.12: Nostril tracking failure due to dark glasses. The left nostril's marker has been placed on the glasses.

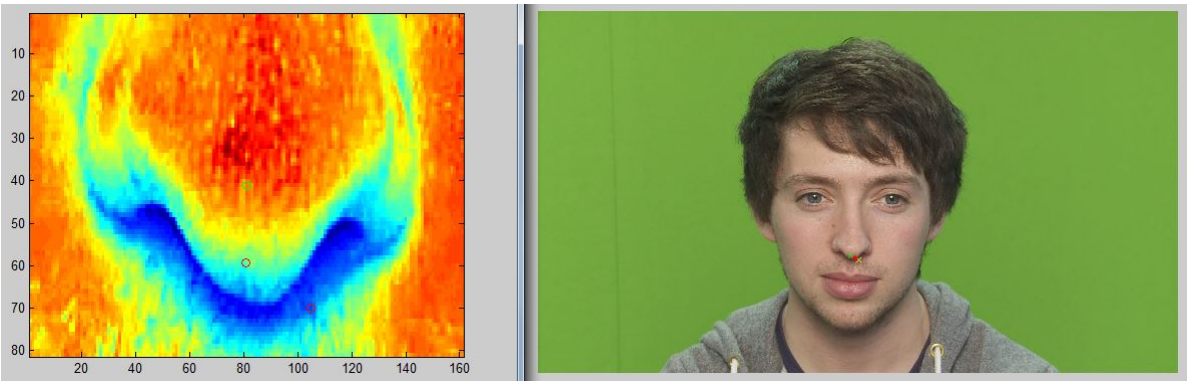


Figure 3.13: Nostril tracking failure due to shadows under nose. The left nostril's marker has been placed under the bridge of the nose.

Figure 3.14: Mouth detection failure due to facial hair. The speaker's "soul patch" has been chosen as the mouth.

Table 3.7: Algorithm Modifications for Problematic Speakers

Area	Problem	Modification	Relevant Speakers
Nostrils	Dark glasses	Smaller nostril search region	13F, 30F, 56M
Nostrils	Shadows under nose	Ignore small region under middle of nose	26M, 40F, 57M
Mouth	Chin facial hair detected as mouth	Score regions based on distance from nose (farther = lower)	25M, 29M, 34M, 52M

on page 66 of Cappelletta's thesis [7], an original feature length of 27 was initially used for testing, with 1st and 2nd derivatives concatenated for a final vector length of 81. The HTK parameters were 4-state viseme HMMs with 20 mixtures after training. The feature vectors were upsampled from 30 to 60 per second using linear interpolation. HTK was used to train and test a visual-only recognizer using the DCT files and the viseme-level label transcriptions.

Initial visual-only scores obtained with the recognizer were much lower than expected, indicating a problem with the training. After checking the viseme label files against the ROI frames they supposedly labelled, the two were found to be out of sync for every clip by varying amounts. This additional complication is discussed in Section 3.4.2.

3.4.2 Finding the Offsets Between Audio and Video

The 5782 MKV clips that were used to create the visual feature files all play back with no issues. The audio and video are perfectly synchronized. The audio was extracted from the clips and used to create phoneme-level label files. These were then converted to viseme-level label files. To create the corresponding visual feature files, the clips were read into Matlab using its VideoReader tool. However, when a clip was read into Matlab, the frames did not match up to the visemes in the corresponding label file. Since there is no equivalent to HTK's HSLab tool for video, this was checked by reading the label file into Matlab and labelling each frame with a viseme based on the time information in the label file and the fact that the video was shot at 30FPS. Frame 1 was assumed to start at time 0. This check revealed that the labels and frames were out of sync. Even worse, the offset varied from clip to clip, and deciding whether a clip was in-sync or not meant going through the clip frame-by-frame, using best judgement to decide whether a certain mouth position accurately represented its label.

The problem was found to stem from the way that ffmpeg treats video clipping requests. The ffmpeg command used to originally create the clips was:

```
# ffmpeg -i fullvid.mkv -vcodec copy -acodec copy -ss 00:xx:xx.xxxx -t 00:xx:xx.xxxx clipN.mkv
```

This command specifies the start time and duration of the clip, and also requests that ffmpeg not transcode the audio or video. As a result, ffmpeg may not be able to clip at those exact points, at least for the video. MPEG-2 Long GoP video contains I-frames, P-frames and B-

```
packet|codec_type=audio|stream_index=1|pts=2|pts_time=0.002000|dt
packet|codec_type=audio|stream_index=1|pts=44|pts_time=0.044000|d
packet|codec_type=audio|stream_index=1|pts=87|pts_time=0.087000|d
packet|codec_type=audio|stream_index=1|pts=130|pts_time=0.130000|
packet|codec_type=audio|stream_index=1|pts=172|pts_time=0.172000|
packet|codec_type=audio|stream_index=1|pts=215|pts_time=0.215000|
packet|codec_type=audio|stream_index=1|pts=258|pts_time=0.258000|
packet|codec_type=audio|stream_index=1|pts=300|pts_time=0.300000|
packet|codec_type=audio|stream_index=1|pts=343|pts_time=0.343000|
packet|codec_type=audio|stream_index=1|pts=386|pts_time=0.386000|
packet|codec_type=audio|stream_index=1|pts=428|pts_time=0.428000|
packet|codec_type=audio|stream_index=1|pts=471|pts_time=0.471000|
packet|codec_type=video|stream_index=0|pts=584|pts_time=0.584000|
packet|codec_type=audio|stream_index=1|pts=514|pts_time=0.514000|
packet|codec_type=video|stream_index=0|pts=517|pts_time=0.517000|
packet|codec_type=video|stream_index=0|pts=551|pts_time=0.551000|
packet|codec_type=audio|stream_index=1|pts=556|pts_time=0.556000|
packet|codec_type=video|stream_index=0|pts=684|pts_time=0.684000|
```

Figure 3.15: Packet structure of an MKV clip. The packets are presented in order of "pts". In this case the offset between the audio and video is 5170000 - 20000. MediaInfo was calculating the offset from the first video packet in the list, i.e. 5840000 - 20000.

frames (a good primer on MPEG-2 video compression can be found at [85]). ffmpeg can only split a video at a keyframe (an I-frame), so when a start time is specified, ffmpeg finds the first I-frame after that point and uses it as the start point of the clipped video. It does the same for the end point. For the audio, however, ffmpeg can and does cut at the closest packet (the audio is broken up into packets). This means that the clipped audio and video are of different lengths. To keep the audio and video in the resulting clip in-sync during playback, a video player will check the PTS (Presentation Time Stamps) of each packet and frame. An example of a clip's eventual packet structure is given in Figure 3.15. This is why there was no synchronization issue visible when playing back the clips. When extracting the audio or video frames alone, this PTS information is lost. This is why the extracted frames were out of sync with the extracted audio's label files.

To resolve this issue, it was decided to try and calculate the offset for each clip and apply this to its viseme label file. At first, a program called MediaInfo was tried. This program returned, as part of its information about a clip, an offset between the audio and video. However, when this offset was applied to the label files, manual inspection indicated that the frames were still out of sync with the labels, though by smaller amounts. This problem was resolved with the help of Thierry Focu of Google, who, after looking at the packet and frame information for a clip, recognized it as having GoP format BBIBBPBBPBBPBBP, which meant that the first frame stored in the container is not the first displayed frame, but actually the third, since the first frame displayed must be an I-frame. This is visible in Figure 3.15. MediaInfo was calculating its offset from the PTS of the first stored frame, where it should have been calculating the offset from the first I-frame. Since MediaInfo could not be used, ffmpeg's "ffprobe" tool was used

to output clips' packet info to a text file, where it was parsed by a script to find the correct offsets. These offsets were then applied to the viseme label files. Manual inspection of the results appeared to show the frames finally in sync with their labels.

With the label files finally matching their corresponding visual feature files, the visual-only baselines could be found. The baselines are given and discussed in Chapter 4.

3.5 Audio-Visual Baseline

Section 2.3 explained the various methods of integration that can be chosen to create an audio-visual speech recognizer. For TCD-TIMIT's audio-visual baseline, early integration was chosen. This meant concatenating the MFCC audio feature vectors with their DCT visual feature vector counterparts.

To concatenate audio and video feature vectors, their framerates must be equal. One way to accomplish this is to upsample the video feature vectors to the audio framerate using linear interpolation ([60], [91], [30], [24]). This method was used to upsample the DCT vectors to 100FPS, the audio sample rate. Ideally, if the audio and video clip are the exact same length, this would create the same number of DCT vectors as MFCC vectors. However, as discussed in Section 3.4.2, this was not the case. There were consistently more MFCC vectors than DCT vectors after upsampling. The solution employed was to find the difference between the two, and delete this number of MFCC vectors. This left the same number of MFCC and DCT vectors, which were then concatenated to form audio-visual feature vectors. Offset phoneme-level label files (offsets applied using the method of Section 3.4.2) could then be used as the audio-visual label files.

3.6 Summary

This chapter describes the creation of TCD-TIMIT, a new database for AVCSR. Specifically, this chapter is concerned with the recording, post-processing and setup of baseline experiments run on the main (volunteer) and secondary (lipspeaker) parts of the new database.

The choice of TIMIT sentences as the reading material for TCD-TIMIT speakers, and the method used to assign sentences to speakers, is discussed in Section 3.1.1. The recording equipment, setup and volunteer recruitment methods are described in Section 3.1. Post-recording, the method of obtaining individual clips for each sentence is described in Section 3.2.1. General information about the TCD-TIMIT speakers is given in Section 3.2.3.

Section 3.3 describes the process of obtaining an audio-only CSR baseline on TCD-TIMIT. The method of forced alignment for creating accurate phoneme label files is introduced in 3.3.2, followed by discussion of the suitability of force-aligned files obtained from original TIMIT label files in Section 3.3.4. Based on these results, a new forced alignment tool, Yuan and Liberman's P2FA [90] is introduced in Section 3.3.5. The audio-only baseline was finally obtained using

force-aligned files created using P2FA.

The steps involved in obtaining TCD-TIMIT's visual CSR baseline are discussed in Section 3.4. Mouth ROIs were extracted and visual feature files created from these ROIs using work originally done by Cappelletta [7]. This is discussed in Section 3.4.1. The detection and correction of a synchronization issue between the phoneme and viseme label files is discussed in Section 3.4.2. Finally, the method used to obtain an audiovisual baseline on TCD-TIMIT is described in Section 3.5. All baseline results (on the main "volunteer" section of TCD-TIMIT) are discussed in Chapter 4.

4

Database Baselines

4.1 Audio Baseline

As discussed in Section 3.3, the audio-only baseline for TCD-TIMIT was obtained after creating force-aligned phoneme-level label files using P2FA. Once the label files were deemed satisfactory, a recognizer was trained using the settings described in Section 3.3.3. To make it slightly easier to compare the results to those obtained on TIMIT, where a 73-27 train-test split was used (Section 3.3.3), a 70-30 split was used on TCD-TIMIT. Also, the exact same settings were used when building recognizers for each dataset (see Section 3.3.3). The speakers in the train and test subsets are given in Table 4.1 and the results are given in Table 4.2. Note that the three non-native English speakers in the TCD-TIMIT database (27M, 35M and 53M), were not used in any experiments.

Table 4.1: Main TCD-TIMIT Train-Test Split

TRAIN	24M	04M	26M	02M	32F	47M	06M	50F	59F	23M
	19M	05F	31F	22M	01M	39M	46F	11F	42M	57M
	43F	29M	17F	37F	21M	12M	38F	48M	16M	52M
	40F	13F	14M	03F	20M	51F	30F	10M	07F	
TEST	28M	55F	25M	56M	49F	44F	33F	09F	18M	54M
	45F	36F	34M	15F	58F	08F	41M			

In discussing the results in Table 4.2, it is important to note the following differences between

Table 4.2: Speaker-independent audio-only monophone recognition results on TIMIT and TCD-TIMIT

	TIMIT		TCD-TIMIT	
	train set	test set	train set	test set
%correct	77.93	73.87	72.53	65.47
%accuracy	65.51	59.38	57.82	47.63

TIMIT and TCD-TIMIT:

1. The number of TCD-TIMIT speakers is over 10 times smaller than the number of TIMIT speakers.
2. Each TCD-TIMIT speaker said almost 10 times as many sentences as each TIMIT speaker.
3. The majority of the TCD-TIMIT speakers come from the same region of Ireland and have similar accents. The three non-Irish TCD-TIMIT speakers were not used in obtaining the audio baseline (56 of 59 speakers used). In contrast, the TIMIT speakers are semi-evenly distributed between 8 different regions of the U.S.
4. The /dx/ phoneme was not used in the TCD-TIMIT label files (see Section 3.3.5). Instead, it was mapped to /t/.
5. None of the SX or SI sentences said by speakers in TIMIT's training set are said by speakers in the test set. In contrast, there was no train-test distinction used when assigning sentences to TCD-TIMIT speakers, so some SX and SI sentences are present in the training and test sets, although said by different speakers.

The TCD-TIMIT audio baseline is not supposed to be equal to or higher than TIMIT's baseline. Nevertheless, it is useful to compare them to see whether the TCD-TIMIT baseline makes sense. From the results, it can be seen that correctness and accuracy scores for TIMIT's train and test set are higher than their TCD-TIMIT counterparts. Looking at the jump in correctness and accuracy from the test to the training sets, TIMIT's training set scores are only 4.06% (correctness) and 6.13% (accuracy) higher than its test set. On the other hand, TCD-TIMIT's training set scores are 7.06% (correctness) and 10.19% (accuracy) higher than its test set. This suggests that the recognizer trained on TCD-TIMIT audio was either less robust to unseen speakers, i.e. less speaker-independent than TIMIT, or, less robust to unseen data in general, i.e. not as well-trained. To test for evidence of the first case (speaker dependence of TCD-TIMIT), a speaker-dependent audio baseline was next obtained on TCD-TIMIT. The results from two different speaker-dependent train-test splits are given in Table 4.3.

The speaker-dependent results in Table 4.3 show virtually no change in the training set's correctness or accuracy scores compared to the speaker-independent results of Figure 4.2. However,

Table 4.3: Audio-only recognition results from two speaker-dependent TCD-TIMIT train-test splits

	Split 1		Split 2	
	train set	test set	train set	test set
%correct	72.32	66.81	72.83	67.62
%accuracy	57.23	49.84	57.82	50.63

there is a slight increase (˜2%) in the test set scores, which indicates that speaker dependence is at least partially responsible for the scores in Figure 4.2.

To investigate the performance of individual TCD-TIMIT speakers versus TIMIT speakers, HTK was used to output speaker-by-speaker correctness and accuracy scores for both datasets. The variance in speaker performance was then found. These variances are given in Table 4.4.

Table 4.4: Inter-speaker Performance Variance in TIMIT and TCD-TIMIT Train/Test Subsets

Subset	Variance in Correctness (%)	Variance in Accuracy (%)
TIMIT Train Set	14.6	25.4
TIMIT Test Set	15.8	29.8
TCD Train Set	10.6	14.6
TCD Test Set	11.4	15.8

The main point emphasized by Table 4.4 is that due to much more data being available per TCD-TIMIT speaker, the inter-speaker performance variance is lower than that of TIMIT. However, the low TCD-TIMIT variances also suggest that there are no "bad-apple" individual speaker performances which would explain the low overall recognition results. This lends additional credence to the theory that the TCD-TIMIT recognizer was less robust than the TIMIT recognizer. It is also possible that the TCD-TIMIT speakers are more similar to each other than the TIMIT speakers. TIMIT has a wider range of accents than TCD-TIMIT, and TIMIT is less gender-balanced (70%/30%) than TCD-TIMIT (52%/48%).

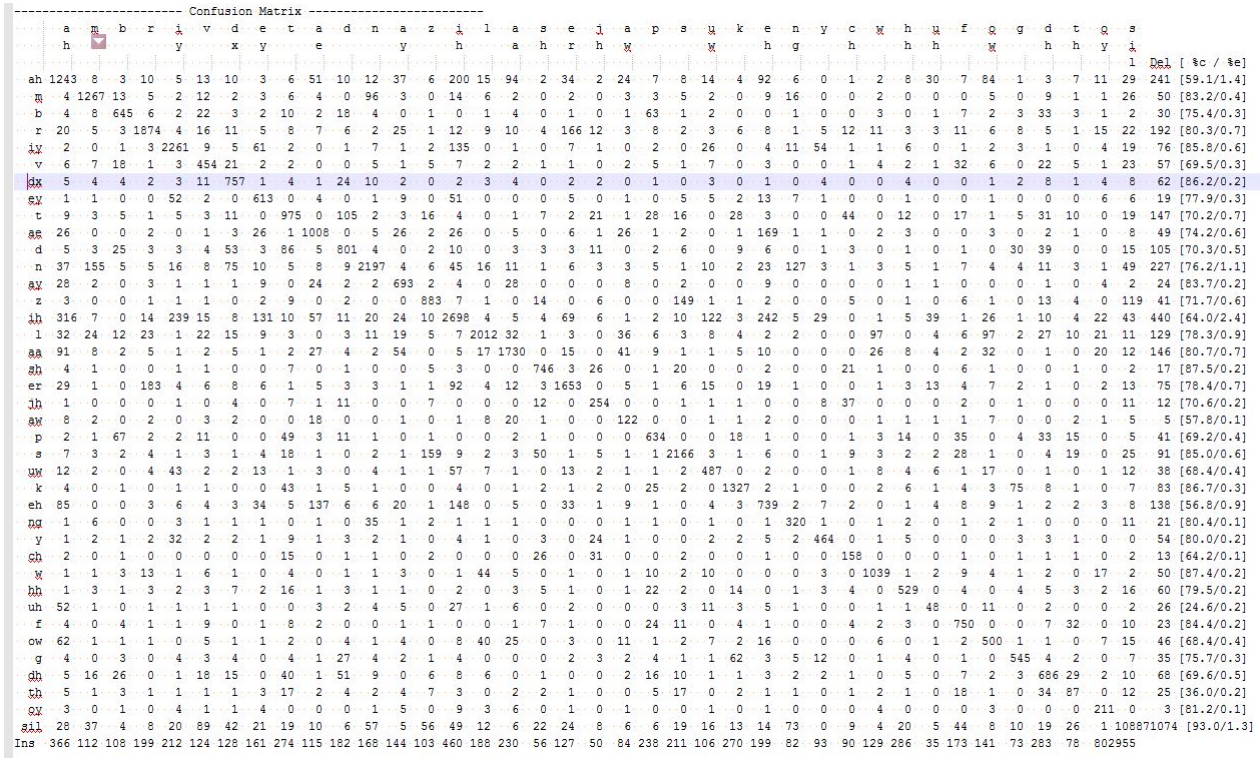
4.1.1 Phoneme Performance Between TIMIT and TCD-TIMIT

To see which phonemes were recognized most correctly and incorrectly in TIMIT and TCD-TIMIT, HTK was used to generate confusion matrices for their test subsets. These are given in Figures 4.1 and 4.2.

Comparing Figures 4.1 and 4.2:

- /t/ is confused mostly with /d/ in both TIMIT and TCD-TIMIT. However, it is confused with /d/ more in TCD-TIMIT, and also deleted more. The most likely reason for this

Figure 4.1: TIMIT Test Set Confusion Matrix



	a	m	b	r	i	v	d	e	t	a	d	n	a	z	i	l	a	s	e	j	a	p	s	u	k	e	n	y	c	w	h	u	f	o	g	d	t	o	s		
	h				y		x	y		e			y		h		a	h	r	h	w			w		h	g		h		h	h		w			h	h	y	i		
																																								l	Del	[%c / %e]
ah	1243	8	3	10	5	13	10	3	6	51	10	12	37	6	200	15	94	2	34	2	24	7	8	14	4	92	6	0	1	2	8	30	7	84	1	3	7	11	29	241	[59.1/1.4]
m	4	1267	13	5	2	12	2	3	6	4	0	96	3	0	14	6	2	0	2	0	3	3	5	2	0	9	16	0	0	2	0	0	0	5	0	9	1	1	26	50	[83.2/0.4]
b	4	8	645	6	2	22	3	2	10	2	18	4	0	1	0	1	4	0	1	0	1	63	1	2	0	0	1	0	0	3	0	1	7	2	3	33	3	1	2	30	[75.4/0.3]
r	20	5	3	1874	4	16	11	5	8	7	6	2	25	1	12	9	10	4	166	12	3	8	2	3	6	8	1	5	12	11	3	3	11	6	8	5	1	15	22	192	[80.3/0.7]
iy	2	0	1	3	2261	9	5	61	2	0	1	7	1	2	135	0	1	0	7	1	0	2	0	26	0	4	11	54	1	1	6	0	1	2	3	1	0	4	19	76	[85.8/0.6]
v	6	7	18	1	3	454	21	2	2	0	0	5	1	5	7	2	2	1	1	0	2	5	1	7	0	3	0	0	1	4	2	1	32	6	0	22	5	1	23	57	[69.5/0.3]
dx	5	4	4	2	3	11	757	1	4	1	24	10	2	0	2	3	4	0	2	2	0	1	0	3	0	1	0	4	0	0	4	0	0	1	2	8	1	4	8	62	[86.2/0.2]
ey	1	1	0	0	52	2	0	613	0	4	0	1	9	0	51	0	0	0	5	0	1	0	5	5	2	13	7	1	0	0	1	0	0	1	0	0	0	6	6	19	[77.9/0.3]
t	9	3	5	1	5	3	11	0	975	0	105	2	3	16	4	0	1	7	2	21	1	28	16	0	28	3	0	0	44	0	12	0	17	1	5	31	10	0	19	147	[70.2/0.7]
ae	26	0	0	2	0	1	3	26	1	1008	0	5	26	2	26	0	5	0	6	1	26	1	2	0	1	169	1	1	0	2	3	0	0	3	0	2	1	0	8	49	[74.2/0.6]
d	5	3	25	3	3	4	53	3	86	5	801	4	0	2	10	0	3	3	3	11	0	2	6	0	9	6	0	1	3	0	1	0	1	0	30	39	0	0	15	105	[70.3/0.5]
n	37	155	5	5	16	8	75	10	5	8	9	2197	4	6	45	16	11	1	6	3	3	5	1	10	2	23	127	3	1	3	5	1	7	4	4	11	3	1	49	227	[76.2/1.1]
ay	28	2	0	3	1	1	1	9	0	24	2	2	693	2	4	0	28	0	0	0	8	0	2	0	0	9	0	0	0	0	1	1	0	0	0	1	0	4	2	24	[83.7/0.2]
z	3	0	0	1	1	1	0	2	9	0	2	0	0	883	7	1	0	14	0	6	0	0	149	1	1	2	0	0	5	0	1	0	6	1	0	13	4	0	119	41	[71.7/0.6]
ih	316	7	0	14	239	15	8	131	10	57	11	20	24	10	2698	4	5	4	69	6	1	2	10	122	3	242	5	29	0	1	5	39	1	26	1	10	4	22	43	440	[64.0/2.4]
l	32	24	12	23	1	22	15	9	3	0	3	11	19	5	7	2012	32	1	3	0	36	6	3	8	4	2	2	0	0	97	0	4	6	97	2	27	10	21	11	129	[78.3/0.9]
aa	91	8	2	5	1	2	5	1	2	27	4	2	54	0	5	17	1730	0	15	0	41	9	1	1	5	10	0	0	0	26	8	4	2	32	0	1	0	20	12	146	[80.7/0.7]
sh	4	1	0	0	1	1	0	0	7	0	1	0	0	5	3	0	0	746	3	26	0	1	20	0	0	2	0	0	21	1	0	0	6	1	0	0	1	0	2	17	[87.5/0.2]
er	29	1	0	183	4	6	8	6	1	5	3	3	1	1	92	4	12	3	1653	0	5	1	6	15	0	19	1	0	0	1	3	13	4	7	2	1	0	2	13	75	[78.4/0.7]
th	1	0	0	0	1	0	4	0	7	1	11	0	0	7	0	0	0	12	0	254	0	0	1	1	1	0	0	8	37	0	0	0	2	0	1	0	0	0	11	12	[70.6/0.2]
aw	8	2	0	2	0	3	2	0	0	18	0	0	1	0	1	8	20	1	0	0	122	0	0	1	1	2	0	0	0	1	1	1	1	7	0	0	2	1	5	5	[57.8/0.1]
p	2	1	67	2	2	11	0	0	49	3	11	1	0	1	0	0	2	1	0	0	0	634	0	0	18	1	0	0	1	3	14	0	35	0	4	33	15	0	5	41	[69.2/0.4]
s	7	3	2	4	1	3	1	4	18	1	0	2	1	159	9	2	3	50	1	5	1	1	2166	3	1	6	0	1	9	3	2	2	28	1	0	4	19	0	25	91	[85.0/0.6]
uw	12	2	0	4	43	2	2	13	1	3	0	4	1	1	57	7	1	0	13	2	1	1	2	487	0	2	0	0	1	8	4	6	1	17	0	1	0	1	12	38	[68.4/0.4]
k	4	0	1	0	1	1	0	0	43	1	5	1	0	0	4	0	1	2	1	2	0	25	2	0	1327	2	1	0	0	2	6	1	4	3	75	8	1	0	7	83	[86.7/0.3]
eh	85	0	0	3	6	4	3	34	5	137	6	6	20	1	148	0	5	0	33	1	9	1	0	4	3	739	2	7	2	0	1	4	8	9	1	2	2	3	8	138	[56.8/0.9]
ng	1	6	0	0	3	1	1	1	0	1	0	35	1	2	1	1	1	0	0	0	1	1	0	1	0	1	320	1	0	1	2	0	1	2	1	0	0	0	11	21	[80.4/0.1]
y	1	2	1	2	32	2	2	1	9	1	3	2	1	0	4	1	0	3	0	24	1	0	0	2	2	5	2	464	0	1	5	0	0	0	3	3	1	0	0	54	[80.0/0.2]
ch	2	0	1	0	0	0	0	0	15	0	1	1	0	2	0	0	0	26	0	31	0	0	2	0	0	1	0	0	158	0	0	0	1	0	1	1	1	0	2	13	[64.2/0.1]
w	1	1	3	13	1	6	1	0	4	0	1	1	3	0	1	44	5	0	1	0	1	10	2	10	0	0	0	3	0	1039	1	2	9	4	1	2	0	17	2	50	[87.4/0.2]
hh	1	3	1	3	2	3	7	2	16	1	3	1	1	0	2	0	3	5	1	0	1	22	2	0	14	0	1	3	4	0	529	0	4	0	4	5	3	2	16	60	[79.5/0.2]
uh	52	1	0	1	1	1	1	0	0	3	2	4	5	0	27	1	6	0	2	0	0	0	3	11	3	5	1	0	0	1	1	48	0	11	0	2	0	0	2	26	[24.6/0.2]
f	4	0	4	1	1	9	0	1	8	2	0	0	1	1	0	0	1	7	1	0	0	24	11	0	4	1	0	0	4	2	3	0	750	0	0	7	32	0	10	23	[84.4/0.2]
ow	62	1	1	1	0	5	1	1	2	0	4	1	4	0	8	40	25	0	3	0	11	1	2	7	2	16	0	0	0	6	0	1	2	500	1	1	0	7	15	46	[68.4/0.4]
g	4	0	3	0	4	3	4	0	4	1	27	4	2	1	4	0	0	0	2	3	2	4	1	1	62	3	5	12	0	1	4	0	1	0	545	4	2	0	7	35	[75.7/0.3]
dh	5	16	26	0	1	18	15	0	40	1	51	9	0	6	8	6	0	1	0	0	2	16	10	1	1	3	2	2	1	0	5	0	7	2	3	686	29	2	10	68	[69.6/0.5]
th	5	1	3	1	1	1	1	3	17	2	4	2	4	7	3	0	2	2	1	0	0	5	17	0	2	1	1	0	1	2	1	0	18	1	0	34	87	0	12	25	[36.0/0.2]
oy	3	0	1	0	4	1	1	4	0	0	0	1	5	0	9	3	6	0	1	0	1	0	0	1	0	1	0	0	0	4	0	0	0	3	0	0	0	211	0	3	[81.2/0.1]
sil	28	37	4	8	20	89	42	21	19	10	6	57	5	56	49	12	6	22	24	8	6	6	19	16	13	14	73	0	9	4	20	5	44	8	10	19	26	1	108871074	[93.0/1.3]	
Ins	366	112	108	199	212	124	128	161	274	115	182	168	144	103	460	188	230	56	127	50	84	238	211	106	270	199	82	93	90	129	286	35	173	141	73	283	78	802955			

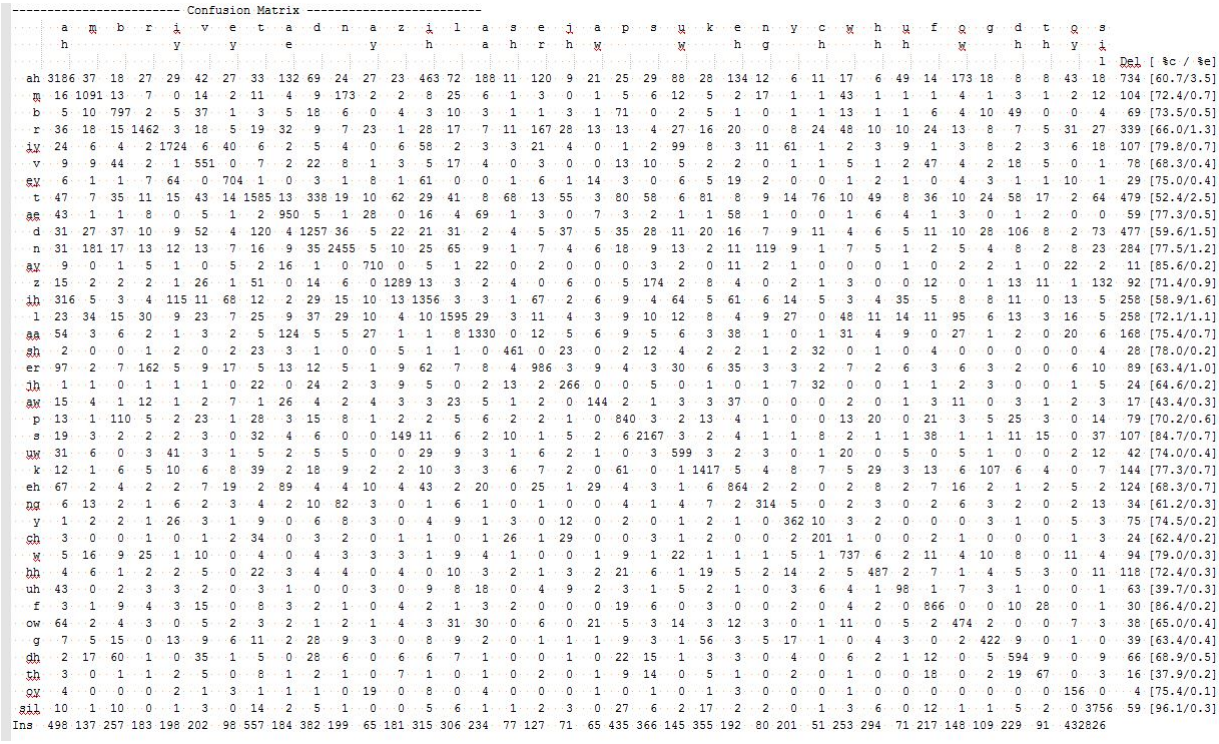
is that the /dx/ phoneme was folded into /t/ in the TCD-TIMIT transcriptions, where perhaps some of its instances should have been folded into /d/. It is also possible that the placement or performance of the clip-on mic makes TCD-TIMIT speakers' /t/s sound more nasal.

- /r/ is confused more (mostly with /er/) and deleted more in TCD-TIMIT. This may be due to inexact boundaries between preceding vowels and /r/ created by P2FA. Some evidence of this was found while manually inspecting files.

- The phonemes /th/ and /uh/ are evident by their bad performance in both TIMIT and TCD-TIMIT. Both phonemes are correctly recognized less than 40% of the time. /th/ is easily confused with /dh/ and /t/, while /uh/ is easily confused with /ah/ and /ih/. /th/ is the most infrequent phoneme in TCD-TIMIT (4th most infrequent in TIMIT) and /uh/ is the most infrequent phoneme in TIMIT (3rd most infrequent in TCD-TIMIT), so the relative scarcity of training data likely also plays a part in their poor performances.

- /aw/ is confused a lot more with /eh/ in TCD-TIMIT's test set. One possible explanation (formed after listening to the audio clips) is that these two phonemes may be more similar-sounding in Hiberno-English.

Figure 4.2: TCD Test Set Confusion Matrix



- Lip-based plosives /b/ and /p/ have similar performance in both TIMIT and TCD-TIMIT, while tongue-body based plosives /g/ and /k/ are confused roughly 10% more in TCD-TIMIT. This may be due to P2FA creating inexact boundaries for /g/ and /k/.
- Nasal /m/ is confused with /n/ more in TCD-TIMIT.
- /er/ is confused with /ah/ more in TCD-TIMIT.

Looking at insertions in both confusion matrices, /t/ was the most wrongly-inserted phoneme in TCD-TIMIT, while /ih/ was the most wrongly-inserted phoneme in TIMIT. /ah/ was a close second in both cases. /ih/ would mostly be associated with breathing noises, while /t/ would mostly be associated with high-frequency "click"-type noise. In a small number of clips, /t/s were found to have been inserted before the speaker had started speaking. Inspecting the video clip of the speaker saying the sentence, they can be seen opening their mouth to begin the sentence, bringing their tongue down from the roof of their mouth and making a very small amount of noise. Evidently, this was picked up by the clip-on mic underneath the chin. The time and duration of these noises matches the time and duration of the phantom /t/s output by the recognizer. This only accounts for a small number of the incorrectly inserted /t/s, but other noise events like this may account for other insertions. TCD-TIMIT has a larger number of insertions with respect to total number of phonemes than TIMIT (17.8% vs 14.5%).

4.1.2 Comparisons with TIMIT Baselines in the Literature

In this section, the TIMIT baseline results obtained on the recognizer detailed in Section 3.3.3 (results in Figure 4.2) are compared to other published baselines with similar parameters. The purpose of this was to verify that the baselines in Figure 4.2 were consistent with results of similar recognizers, which would provide evidence that the recognizer had been trained correctly. Due to the large number of variables involved when creating and testing a recognizer, finding published baselines with the exact same setup proved difficult. The key criteria for a comparable baseline are:

- The recognizer was trained and tested on TIMIT.
- Monophone HMMs only were used to model the data.
- MFCCs were used as features.

The 5 most similar systems found, their parameters and their results, are summarized in Tables 4.5 and 4.6.

Table 4.5: Baseline Monophone HMM Results Reported on TIMIT - Part 1

Paper	Year	No. of HMMs	States	Mixtures	Feature Type	MFCC Order	Preemphasis
This work	2013	39	3	31	MFCC_D_A_Z	12	yes
Young [87]	1992	48	3	10	MFCC_E_D	12	yes
Kelly [43]	2009	39	3	31	MFCC_D_A_Z	12	yes
Siniscalchi et al. [77]	2007	39	3	16	MFCC_E_D_A	12	N/A
Kapadia et al. [41]	1992	39	3	16	MFCC_E_D_A	12	N/A
Rose and Momayyez [71]	2007	N/A	3	1	MFCC_D_A	12	N/A

Notes: Fields with no information available are marked with "N/A". In the Feature Type column, "_E" "_D" and "_A" mean append log Energy, Deltas and Acceleration respectively. "_Z" means use cepstral mean normalization.

There are a few additional notes about the baselines in Tables 4.5 and 4.6. Young [87] followed Hon and Lee [49] in using a set of 48 phonemes during training. He mapped down to the 39-phoneme set of Table 2.2 during recognition. The rest of the baselines used the phoneme set of Table 2.2 from the beginning, except for Siniscalchi et al. [77]. Their reduced set had a slight difference: instead of mapping closures like /bcl/ and /tcl/ to /sil/ (as in Table 2.2), they mapped these closures to the following phoneme, so for example /bcl/ /b/ became /b/.

From the table, the average accuracy score is 61.2%, with a variance of 28. The lowest baseline (Rose and Momayyez [71]) is the most difficult to compare to the rest, since the HMMs only had 1 mixture each, and the number of HMMs used was not specified in the paper. 51.7% is a high accuracy score for single-mix monophone HMMs. The equivalent score on the recognizer

Table 4.6: Baseline Monophone HMM Results Reported on TIMIT - Part 2

Paper	FPS	Window Size	Training Set	Test Set	Corr %	Acc %	Bigram?
This work	100	25ms	Full TRAIN set	Full TEST set	73.87	59.38	yes
Young [87]	100	16ms	All SI+SX sentences	160 random SI/SX sentences	71.9	62.8	N/A
Kelly [43]	100	25ms	Full TRAIN set	Full TEST set	72.11	58.25	yes
Siniscalchi et al. [77]	N/A	N/A	412 TRAIN speakers (SI+SX)	162 TEST speakers (SI+SX)	N/A	62.73	N/A
Kapadia et al. [41]	100	16ms	All SI+SX sentences	336 random SI/SX sentences	69.68	66.07	yes
Rose and Momayyez [71]	N/A	N/A	Full TRAIN set (SI+SX)	Full TEST set (SI+SX)	N/A	51.7	yes

Notes: Fields with no information available are marked with "N/A". SA, SX and SI sentences are explained in Section 3.1.1.

in this work, for one mixture, was 41%. The highest score, by Kapadia et al. [41], is also difficult to compare, since it seems the authors used every SX and SI sentence in TIMIT to train, and then picked 336 of those previously-seen sentences for testing, i.e. they tested on previously-seen data. If this is the case, the most comparable score from the recognizer in this work is probably the one obtained on the training set at 17 mixtures, 58.87%. Kapadia et al. also mention that they squared probabilities from their bigram (an empirical decision) and forbade sequences of identical phonemes during testing, and that "comparison with other systems is difficult" as a result. This might explain the difference in scores.

The most suitable baselines to compare to the TIMIT results reported in this work are those of Young [87], Kelly [43] and Siniscalchi et al. [77]. The variance in these four baselines is much lower (5.4). The results obtained in this work are closest to those obtained by Kelly [43] in his undergraduate thesis. This is not surprising, as his training method was followed. Young's HMMs consisted of only 10 states and his features only made use of 1st derivatives, yet his score is high. Young seems to have tested on previously-seen data, stating that "the training set consisted of all SI and SX sentences and the test set consisted of 160 SI and SX sentences chosen at random" [87]. The score obtained on the recognizer in this work at 9 states was 54.44%, using 1st and 2nd derivatives. Young's monophone HMMs were created from triphones which had undergone a number of re-estimation runs, so it is difficult to follow the training regime. However, since he used Lee and Hon's set of 48 phonemes until recognition, rather than using 39 from the beginning, it is also possible that he avoided the introduction of some amount of error.

Siniscalchi et al. [77] trained and tested on different data. Next to Kelly's setup, theirs is the most similar to the one in this work. At 16 Gaussian mixtures, their accuracy was 62.73%. The most comparable score (at 17 mixtures) from the recognizer in this work is 55.3%. While a few details of their system are unknown, these are relatively minor and it is assumed that they used the most common settings. One thing that is known is that their reduced phoneme set used different mappings for closures. It is unclear how this would affect results. They did not use SA sentences. They used only 412 of the TRAIN-set speakers for training, using the other 50 for cross-validation. They increased mixtures in their HMMs "until the saturation of PER (Phoneme Error Rate) on the CV (Cross-Validation) development subset was observed". Since they do not print their correctness score, it is possible that their parameters were set to maximise the accuracy score (e.g. apply a high penalty for insertions).

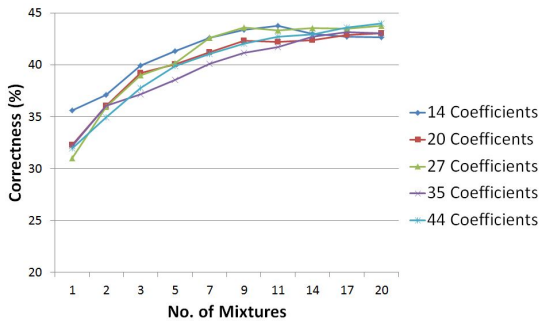
Based on the table, the TIMIT baseline obtained in this work was deemed to be sensible. Since the TIMIT-trained recognizer seems to be correctly trained, and since the exact same parameters and training regime were used when training a recognizer on TCD-TIMIT audio, the implication is that the TCD-TIMIT trained recognizer is also trained correctly, and that its baseline scores are also sensible.

4.2 Visual Baseline

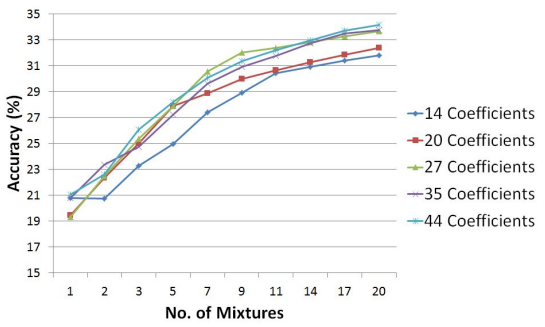
4.2.1 Speaker-dependent Visual Baseline

Section 3.4 details the initial preparations undertaken for the visual baseline experiments. The final parameters to be decided were the number of DCT coefficients and HMM states to use. To decide this, a speaker-dependent recognizer was trained for every combination of 3, 4, and 5-state HMMs and 14, 20, 27, 35 and 44-length DCT coefficient vectors. These are the same coefficient vector lengths used by Cappelletta [7] during his visual-only DCT recognition experiments on VidTIMIT. They were chosen to make comparisons between the experiments easier. Also following Cappelletta's methodology, the DCT vectors were upsampled from 30 to 60fps and then concatenated with their 1st and 2nd derivatives, leading to final vector lengths of 42, 60, 81, 105 and 132. Recognition results for each vector and HMM state length are given in Figure 4.3.

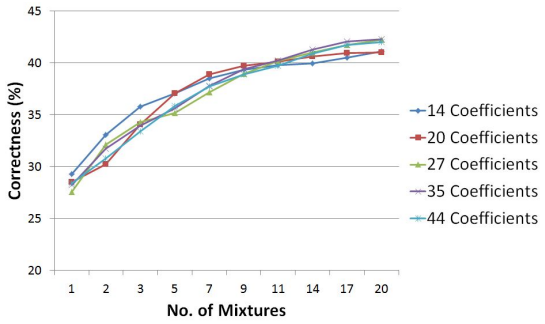
The results in Figure 4.3 are low overall, with correctness and accuracy averaging 41% and 32% respectively at the final mixture count of 20. The highest correctness scores occur with 3-state HMMs (Figure 4.3a), while the highest accuracy scores occur with 4-state HMMs (Figure 4.3d). The performance of the different vector lengths is very similar as the number of mixtures and states are increased. However, the 14 and 20-coefficient vector scores are consistently below the others at 20 mixtures. This is consistent with results found by others. Heckmann et al. [32] found no decrease in word error rate from varying the number of DCT coefficients between 20 and 100 on a single-speaker isolated digit recognition task. Seymour et



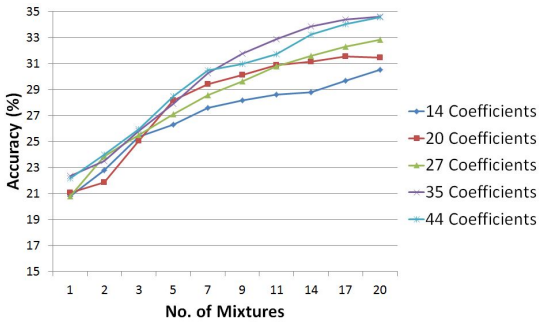
(a) Correctness vs No. of Mixtures (3-state HMMs)



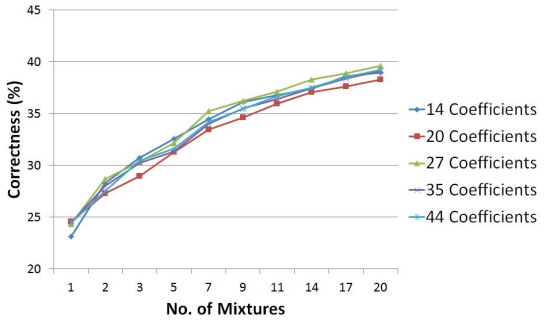
(b) Accuracy vs No. of Mixtures (3-state HMMs)



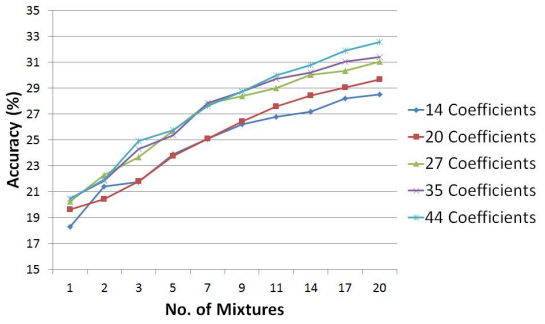
(c) Correctness vs No. of Mixtures (4-state HMMs)



(d) Accuracy vs No. of Mixtures (4-state HMMs)



(e) Correctness vs No. of Mixtures (5-state HMMs)



(f) Accuracy vs No. of Mixtures (5-state HMMs)

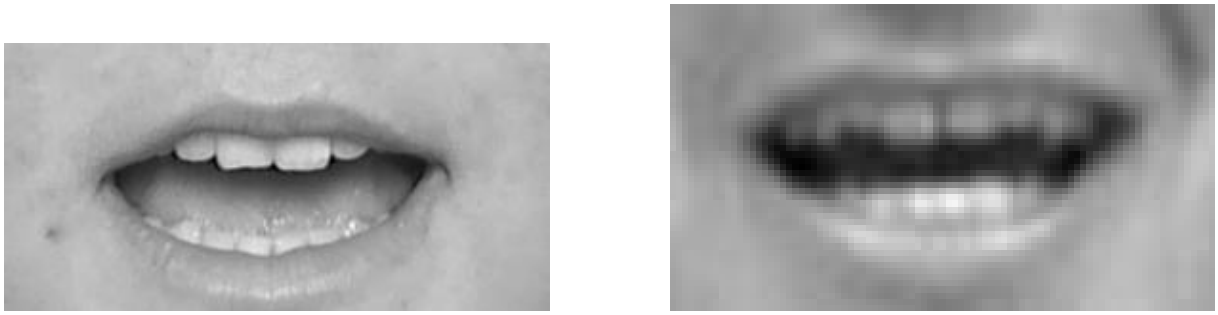
Figure 4.3: Monoviseme recognition results using different combinations of DCT vector and HMM state lengths. The graphs show each combination converging at the final mixture count of 20. The highest correctness result was 44% for 44 coefficients and 3-state HMMs. The highest accuracy result was 34.58% for 35 coefficients and 4-state HMMs. The 5-state results are lower then their 3 and 4-state counterparts.

al. [76] tested two different methods of extracting the most relevant DCT features on a speaker-independent isolated digit recognition task. They found the performance of both methods to be similar beyond 40 coefficients. Scanlon et al. [74] found that of vectors containing 15, 28 and 36 DCT coefficients and their deltas, the 28-coefficient vectors gave the highest performance on an isolated-word recognition task. Their deltas were not calculated between adjacent frames but between frames a certain distance apart (distance depending on the length of the utterance). They also found that using 28 deltas alone led to higher word-recognition accuracy than any combination of DCT coefficients and deltas. They note that "increasing the number of transform coefficients increases the visual recognition accuracy" but also note that "the size of training data available limits the possible feature vector dimensions for good recognition". Cappelletta [7] used the same DCT coefficient sizes as in Figure 4.3 in viseme recognition experiments on continuous speech. He did not use 1st or 2nd derivatives during these experiments. He found that "recognition results are not highly related to the DCT vector length". After that result, he chose to use 27-length vectors for the remainder of his experiments.

The findings of these researchers and the results of Figure 4.3 suggest that the 27, 35 and 44-length DCT coefficient vectors are all adequate for representing the ROIs. Since the 44-length vector has consistently high accuracy, it was chosen as the vector length for further visual-only and audio-visual experiments. For the choice of HMM state length, Figure 4.3 shows 4-state HMMs returning the highest accuracy results. Hence, 4-state HMMs were chosen for further experiments. All subsequent experiments in this section use 4-state HMMs and 44-coefficient DCT vectors (plus 1st and 2nd derivatives) unless otherwise specified.

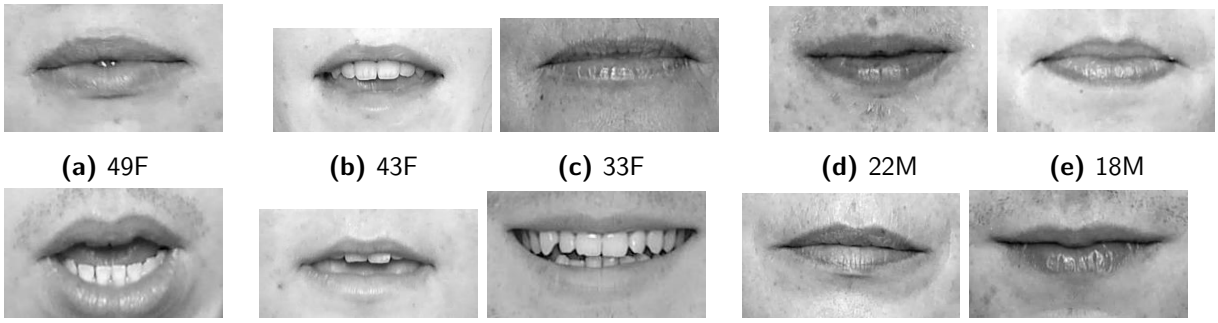
The results of Figure 4.3 were lower than expected. Results similar to those reported by Heckmann et al. [32] or Seymour et al. [76] were not expected, since isolated digit recognition is much simpler than viseme recognition in continuous speech. Cappelletta [7], however, did run a speaker-dependent viseme recognition experiment on VidTIMIT. He used a very similar setup (27 DCT coefficients, 4-state HMMs), and reported results of 44.89% correctness and 42.25% accuracy. The equivalent results on TCD-TIMIT, read from Figures 4.3c and 4.3d, are 42.19% correctness and 32.81% accuracy. Assuming the training regime was the same in both experiments (a reasonable assumption, since Cappelletta's training method was followed in these experiments), the main difference is the database used (VidTIMIT vs TCD-TIMIT). The ROIs extracted from TCD-TIMIT (using Cappelletta's method, see Section 2.2.3.1) were all checked manually and found to be similar to those Cappelletta extracted from VidTIMIT. A comparison between typical ROIs from both databases is given in Figure 4.4. The viseme-level label files used were also deemed adequate after manually inspecting a sample set (see Section 3.4.2).

To investigate the reliability of the results of Figure 4.3, the 4-state, 44-coefficient experiment was performed again using a different train-test split. Results for both splits are given in Table 4.7. The results show that performance does not vary with different training and test data. The individual speaker results were then checked for significantly below-average performances. The inter-speaker performance variances were found to be high, with a correctness variance of



(a) Typical TCD-TIMIT ROI (286x140 pixels) **(b)** Typical VidTIMIT ROI (89x50 pixels)

Figure 4.4: Comparison between a typical TCD-TIMIT and VidTIMIT ROI, both extracted using Cappelletta's method [7]



(a) 49F **(b)** 43F **(c)** 33F **(d)** 22M **(e)** 18M

(f) 20M **(g)** 03F **(h)** 39M **(i)** 44F **(j)** 12M

Figure 4.5: Sample ROIs from the top 10 speakers of Table 4.8

54.08 and an accuracy variance of 25.67. The scores of the 10 best and worst speakers are given in Table 4.8, and example ROIs from these speakers are given in Figures 4.5 and 4.6 for comparison. The full scores of every speaker can be found in Appendix D.

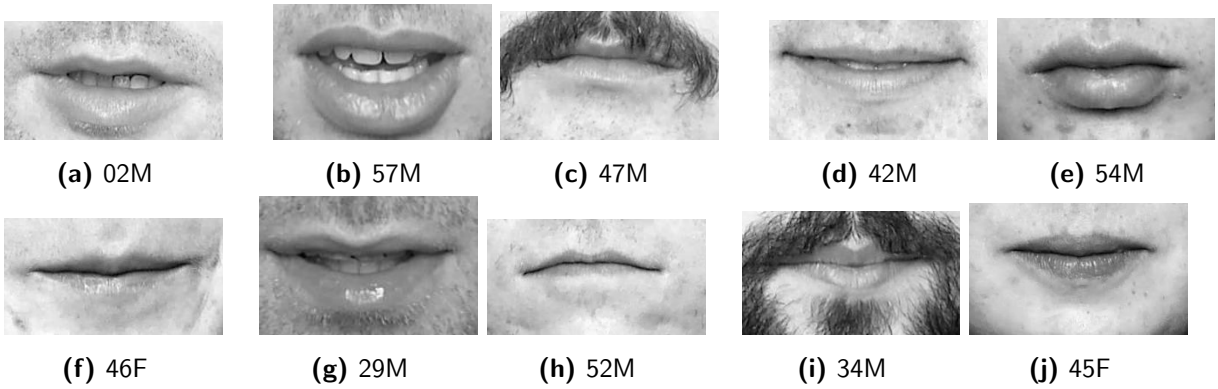
Table 4.7: Visual-only recognition results from two speaker-dependent TCD-TIMIT train-test splits

	Split 1		Split 2	
	Train set	Test set	Train set	Test set
%correct (%)	42.69	41.98	42.19	41.89
%accuracy (%)	36.05	34.54	36.01	34.82

The images of Figures 4.5 and 4.6 do not provide any especially obvious clues to explain the performance variance between the best and worst speakers. It is interesting to note that the two volunteers with the most facial hair are in the bottom 10. Also worth noting is that the top 10 speakers contain an equal number of males and females, while the bottom 10 consists of

Table 4.8: The 10 best and worst speakers from the 4-state, 44-coefficient experiment of Figure 4.3

Top 10 Speakers			Bottom 10 Speakers		
Speaker	Correctness	Accuracy	Speaker	Correctness	Accuracy
12M:	51.9	39.67	02M:	16.68	16.49
44F:	45.21	39.71	57M:	24.92	22.89
39M:	49.56	39.86	47M:	23.83	23.24
03F:	50.87	40.1	42M:	28.6	25.9
20M:	44.88	40.24	54M:	32.97	26.86
18M:	45.96	40.37	46F:	33.3	28.69
22M:	50.44	40.39	29M:	29.61	28.77
33F:	46.02	40.99	52M:	33.39	28.78
43F:	47.1	41.1	34M:	30.8	29.22
49F:	49.62	41.9	45F:	35.85	30.14



(a) 02M **(b)** 57M **(c)** 47M **(d)** 42M **(e)** 54M

(f) 46F **(g)** 29M **(h)** 52M **(i)** 34M **(j)** 45F

Figure 4.6: Sample ROIs from the bottom 10 speakers of Table 4.8

8 males and 2 females. The average accuracy of the 29 male speakers was 32.93%, while the average accuracy of the 27 females was 36.21%. One theory behind the disparity between the top and bottom 10 is that most of the top 10 speakers were quite expressive, whereas most of the bottom 10 moved their mouths less while speaking. To examine how detrimental the bottom 10 speakers had been to the trained HMMs, a new recognizer was trained and tested without them. The results from this recognizer are compared to the original results in Table 4.9. The results show that the removal of the 10 worst speakers only improved the accuracy of the recognizer by 3%. The average (test-set) accuracy between the top 46 speakers in the original recognizer is 36.34%, so the 37.31% score indicates that the recognizer did not become significantly more accurate with the 10 worst speakers removed.

Finally, the confusion matrices output by HTK offer an insight into each viseme HMM's

Table 4.9: Speaker-dependent viseme recognition results with and without 10 worst speakers (4-state HMMs, 44 DCT coefficients)

	Original 56 volunteers		Without 10 worst speakers	
	train set	test set	train set	test set
%correct	42.69	41.98	45.21	44.84
%accuracy	36.05	34.54	38.54	37.31

performance. For the 4-state, 44-coefficient case in Figure 4.3d, the confusion matrix is given in Figure 4.7. The phonemes which were mapped to these visemes are given in Table 2.4. Table 2.4 also shows the visibility ranking which Jeffers and Barley [40] assigned to each viseme. Comparing the performance of the visemes in the confusion matrix to their visibility rankings (ignoring the silence viseme /S/), most perform correspondingly, with a few exceptions. The correctness score of viseme /B/ is only the 6th highest, while Jeffers and Barley rank it as the 2nd most visible viseme. The correctness score of /G/, which consists only of the phonemes /oy/ and /ao/, places it in 4th, compared to the Jeffers and Barley ranking of 7th. /E/, which consists of phonemes /dh/ and /th/ and is ranked 5th by Jeffers and Barley, has a correctness score of only 58.3%, the 3rd lowest score (ranking it 9th).

4.2.2 Speaker-independent Visual Baseline

A speaker-independent visual baseline was also obtained on the ROIs. The results from two train-test splits are given in Table 4.10. The first split was the same as the first audio-only speaker independent split, given in Table 4.1.

Table 4.10: Visual-only recognition results from two speaker-independent TCD-TIMIT train-test splits

	Split 1 (Table 4.1)		Split 2	
	Train set	Test set	Train set	Test set
%correct	42.33	46.78	41.18	46.97
%accuracy	36.50	34.77	35.53	35.61

The speaker-independent results in Table 4.10 are within 1% of their speaker-dependent counterparts (4.9). Furthermore, the jump in accuracy from training to test set is roughly the same (between 0-2% each time). This suggests that the benefit to the recognizer in having previously seen the test-set speakers may have been cancelled out by having trained on more speakers overall. This theory is supported by Figure 4.8, which compares the speaker-independent and dependent test set accuracy results as the mixtures in their recognizers were increased. The graph shows the accuracy score of both sets converging at the final mixture count of 20.

Figure 4.7: Test Set Confusion Matrix for Figure 4.3d (44-coefficient case)

```
====================== HTK Results Analysis =======================
  Date: Thu Sep 19 14:12:52 2013
  Ref : TCDTIMITjeffersVisemesOffset.mlf
  Rec : Results\splitlResults\TCDTestRecognition\TCDTestRec60.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=1736, N=1736]
WORD: %Corr=41.98, Acc=34.54 [H=23619, D=19858, S=12780, I=4186, N=56257]
------------------------ Confusion Matrix -------------------------
       I    C    S    D    B    A    J    H    F    G    K    E  Del [ %c / %e]
   I 6461  500  143  681  475  314  210  441  345  338  445  375 6818 [60.2/7.6]
   C   15 2903   13   31   77   64   33   40   40   25   48   59  655 [86.7/0.8]
   S   38   17 3540   29   26   29   12   25   18   17   29   19  381 [93.2/0.5]
   D    9    6    1  191    7    2    2    7    2    2    7   10   92 [77.6/0.1]
   B  100  253   47  155 2560  217  107  144  130  156  176  120 2555 [61.5/2.9]
   A    7   40   10   31   22 1327   19   21   12   13   37   22  410 [85.0/0.4]
   J  116  368   90  259  292  346 2749  449  415  205  430  322 4724 [45.5/5.9]
   H   48  122   51   67  125  128  142 1753  187   62  133  114 1712 [59.8/2.1]
   F    6   44    9   17   28   22   32   66  629   18   25   31  480 [67.9/0.5]
   G    3    4    1    6    8    3    3    2    2  110    4    0   86 [75.3/0.1]
   K   33   87   24   83   87  106   82  104   81   59 1001   82 1456 [54.7/1.5]
   E   11   42   16   11   39   35   23   41   23   12   29  395  489 [58.3/0.5]
Ins   251  402  240  293  471  337  265  428  320  212  434  533
====================================================================
```

Figure 4.8: Speaker Dependent and Independent Test Set Accuracy (with respect to number of mixtures)

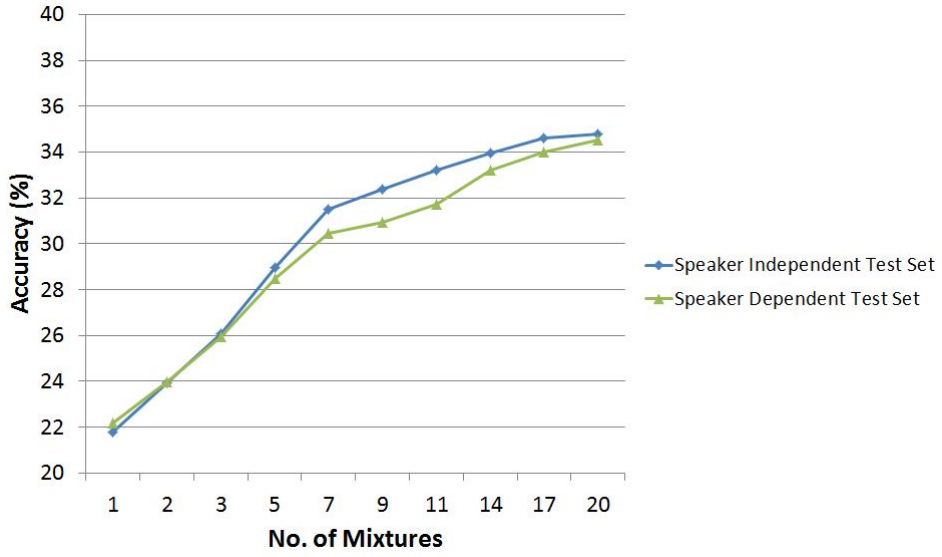


Figure 4.9: Speaker Independent Test Set Confusion Matrix

```
====================== HTK Results Analysis =======================
  Date: Mon Oct 21 01:53:12 2013
  Ref : TCDTIMITjeffersVisemesOffset.mlf
  Rec : Results\split1Results\TCDTestRecognition\TCDTestRec60.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=1666, N=1666]
WORD: %Corr=46.78, Acc=34.77 [H=25134, D=15240, S=13351, I=6453, N=53725]
------------------------ Confusion Matrix -------------------------
      I   C   S   D   B   A   J   H   F   G   K   E  Del [ %c / %e]
  I 7862 315 339 261 715 247 343 440 433 174 432 309 4820 [66.2/7.5]
  C  35 2247  68  39 116 116  75  68  72  14  83  75  802 [74.7/1.4]
  S  21  12 3475  10  35  19  30  25  29   5  30  15  250 [93.8/0.4]
  D  22  10   8 123  21  12   8   5   6   2  16   3  113 [52.1/0.2]
  B 129 155 132  65 3108 188 164 170 207  58 143 125 1822 [66.9/2.9]
  A  31  38  64  21  67 904  51  51  64   6  47  41  501 [65.3/0.9]
  J 159 269 304 102 429 237 4022 371 435  79 389 301 3190 [56.7/5.7]
  H  95 102 123  35 180  89 212 1513 241  38 151 123 1561 [52.1/2.6]
  F  13  35  38  13  51  29  70  69 557   7  45  38  405 [57.7/0.8]
  G  13   8   4   3  14   3   9   6   3  64  10   5   71 [45.1/0.1]
  K  55  69  75  40 134  79 123 109 108  21 958  93 1242 [51.4/1.7]
  E  16  36  21  13  53  38  53  41  46   6  42 301  463 [45.2/0.7]
Ins 441 567 554 226 865 450 578 686 602 169 736 579
====================================================================
```

Comparing the confusion matrix of the speaker-independent recognizer (Figure 4.9) to its speaker-dependent counterpart (Figure 4.7), it performs considerably worse on infrequently-used visemes such as /D/ (25% worse), /A/ (20% worse) and /G/ (30% worse). This may indicate more variation in how speakers express infrequently-used visemes. However, these lower scores are matched by higher scores on the most frequent visemes I (6% better), B (5% better) and J (11% better), leaving it with a very similar overall accuracy score.

Comparing the individual speaker performances between the speaker-independent and dependent recognizers is difficult, as speakers are treated differently in both. Taking only the performances of the 5 best and worst test set speakers (Figure 4.11), 3 of the speakers in the bottom 5 (34M, 54M and 45F) are also in the bottom 10 of Table 4.8, while 3 of the speakers in the top 5 (18M, 49F and 33F) are also in the top 10 of Table 4.8.

4.2.2.1 Brightness Normalization in ROIs

In training recognizers thus far, the brightness of the ROIs had not been normalized. While checking the ROIs to ensure their accuracy, it was felt that there may be enough variance in the brightness (probably due to a visually imperceptible flicker from ceiling lights in the recording room) to warrant performing brightness normalization. This was done with the help of Dr. François Pitié, an image-processing expert in Sigmedia. He recommended a simple clip-

Table 4.11: The 5 best and worst speakers from the speaker-independent visual results

Top 5 Speakers			Bottom 5 Speakers		
Speaker	Correctness	Accuracy	Speaker	Correctness	Accuracy
15F:	47.33	38.94	34M:	24.32	21.33
33F:	48.61	39.72	54M:	51.1	25.98
55F:	52.01	39.97	25M:	46.63	30.52
49F:	48.73	40.37	45F:	39.68	31.09
18M:	47.63	41.47	56M:	47.24	32.1

specific normalization whereby each frame in a clip is normalized with respect to the first frame. Normalizing each ROI with respect to the entire set of ROIs was not considered necessary, as lighting conditions were consistent during recordings. A recognizer trained on the normalized ROIs (using the same train-test split as Figure 4.3) gave results of 46.21% correctness and 34.99% accuracy on the test set. These results offer virtually no improvement on the results of Table 4.10. Hence, the brightness-normalized ROIs were not used in further experiments.

4.3 Audio-Visual Baseline

4.3.1 Visual Component Selection

Section 3.5 describes how the feature vectors for the audio-visual baseline experiments were set up. As that section explains, the joint audio-visual feature vectors consisted of 12 MFCC audio coefficients and their 1st and 2nd derivatives, concatenated with the corresponding visual DCT vector. The vector sampling rate was 100FPS (the visual vectors were upsampled). Since 44-coefficient DCT vectors had been found to perform well in Section 4.2.1, these were chosen as the DCT vectors to concatenate with the MFCCs. However, the full length of these DCT vectors was 132, since they also contained the 1st and 2nd derivatives of the coefficients. Concatenating these vectors with the MFCCs would have produced 168-length vectors, most likely too long for the amount of training data available.

To shorten the vectors, tests were undertaken to find the most useful components of the DCT vectors. Visual-only recognizers were trained and tested on several combinations of original coefficients, 1st and 2nd derivatives, and their results were compared. These results are given in Table 4.12. From the table, the closest performance to the original 132-length vectors was obtained on vectors consisting of the 1st and 2nd derivatives of the DCT coefficients. Hence, these were chosen as the visual component to concatenate with the audio MFCCs, leading to a final audio-visual vector length of 124.

Table 4.12: Visual-only viseme recognition results for vectors composed of different DCT components

DCT Components	Length	Corr	Acc
Original Coefficients	44	20.76	19.69
1st Derivatives	44	42.95	28.82
2nd Derivatives	44	43.75	28
1st and 2nd Derivatives	88	48.32	30.78
Original Coefficients, 1st and 2nd Derivatives	132	41.98	34.54

4.3.2 Audio-Visual Speech Recognition In Noise

Audio-visual speech recognition is usually proposed as a solution when noise affects the audio channel. Therefore, the most common experiment run on audio-visual speech recognizers is to evaluate their performance as the signal-to-noise ratio (SNR) in the audio component of the signal is lowered [23], [62], [91], [74], [76]. To run this experiment, several sets of audio-visual vectors were created. Each set contained audio which had been corrupted by a different amount of additive white Gaussian noise (AWGN). A speaker-dependent recognizer was trained on the clean audio-visual data and then tested on the increasingly noisy data. For comparison, an audio-only recognizer was trained on clean audio and then tested on the increasingly noisy audio data. The results are graphed in Figures 4.10 and 4.11.

Figures 4.10 and 4.11 show that after the SNR drops below 20dB, the audio-visual recognizer's performance is higher than the audio-only recognizer. By the lowest SNR of -10dB, the audio-visual recognizer's accuracy score is 18.32%, 10 percentage points higher than the audio-only score of 8.71%. Above 20dB however, the audio-only recognizer outperforms the audio-visual recognizer. At 40dB SNR (considered "clean", i.e. no AWGN was added to this audio), the audio-only recognizer's accuracy is 53.88%, almost 17 percentage points higher than the audio-visual accuracy of 37.26%. This is most likely due to the large visual component (88 of the 124 coefficients) in the audio-visual vectors. The audio-visual accuracy at 40dB is close to the visual-only accuracy scores found in Section 4.2.

The trends in these results are mostly consistent with other noisy audio experiments in the literature. The most similar experiment for comparison is one by Galatas et al. [23]. Their audio-visual vectors also consisted of DCT coefficients (plus 1st and 2nd derivatives) concatenated with MFCC coefficients (plus 1st and 2nd derivatives), but their HMMs modelled triphones and had 4 mixtures each. Their database was CUAVE [64], hence the task was speaker-dependent isolated-word recognition on a vocabulary of 10 words. Testing at 4 SNRs (10, 5, 0 and -5dB), they found that the word accuracy of the audio-visual speech recognizer was consistently higher than that of the audio-only recognizer. The absolute improvements at each SNR were 0.67% at 10dB, 15.97% at 5dB, 16.33% at 0dB and 5% at -5dB. This is consistent with the trend in Figure 4.11,

Figure 4.10: Audio-visual vs audio-only test set correctness on noisy audio

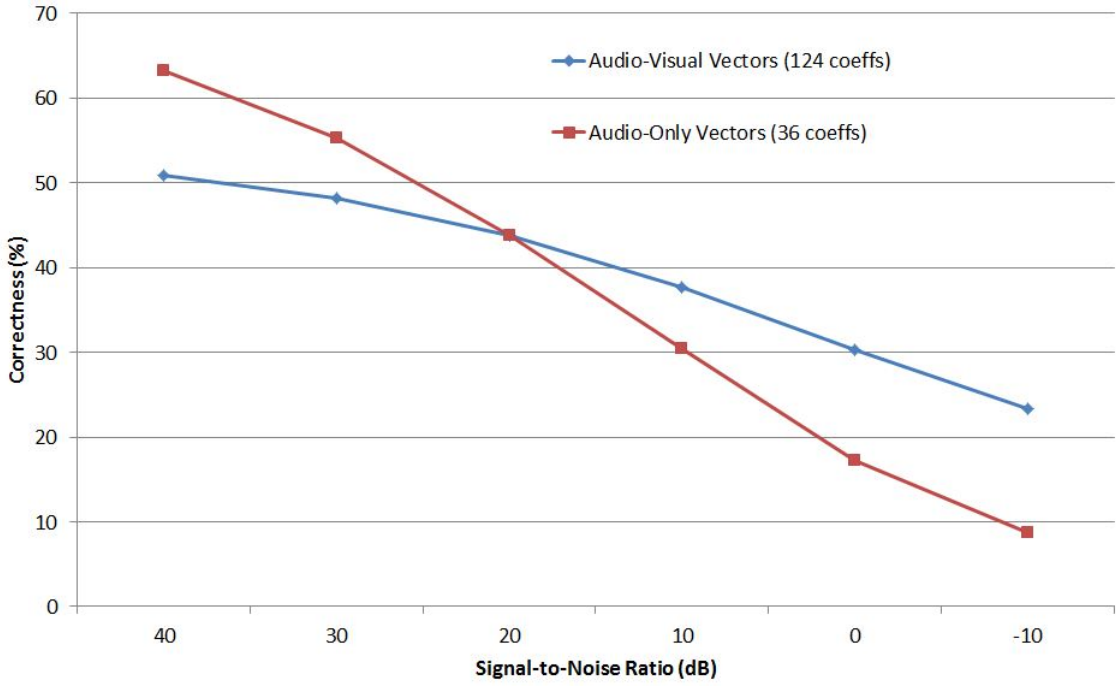
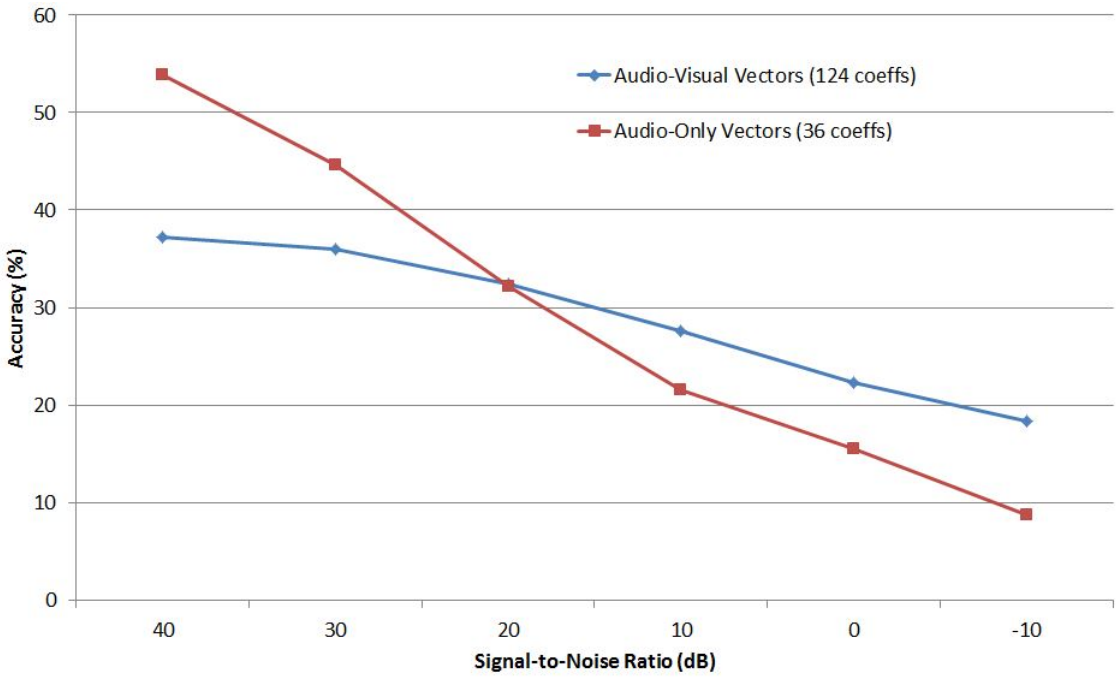


Figure 4.11: Audio-visual vs audio-only test set accuracy on noisy audio



which shows the audio-visual recognizer accuracy above the audio-only accuracy at those SNRs, and shows the accuracy of both recognizers falling as the SNR decreases.

Papandreau et al. [62] also performed noisy audio experiments on CUAVE. Instead of MFCCs, they used 26 FBANK coefficients, and instead of DCTs, they used an AAM (Section 2.2.3). Their audio-visual integration approach was intermediate integration (Section 2.3). Their 10 HMMs modelled each word in CUAVE's vocabulary, and had 8 states each. Their audio vectors are enhanced to provide additional robustness against noise. Despite these differences, a similar trend is observed in their results, which show the audio-visual accuracy higher than the audio-only accuracy by about 15% (absolute) from -5 to 10dB, where the gap narrows until they are within 3% of each other at the "clean" (>20dB) SNR. Also visible is the decrease in the accuracy of both recognizers as the SNR decreases.

Finally, Zhang et al. [91] report correctness results using noisy audio on the AMP/CMU database [8]. For visual features they used shape-based features, while their audio features were 12 MFCC coefficients plus their 1st derivatives. They tested early, intermediate and late integration. The early integration result is the most relevant. Unlike Figure 4.10, their audio-only correctness score remains below the audio-visual score over the entire range of SNRs from 30 to 0dB. However, the gap does narrow from 45 percentage points at 5dB to 10 percentage points at 30dB. The results show that the performance of the audio-only and audio-visual recognizers decreases with the SNR.

4.3.3 Audio-Visual Confusion Matrices

One of the imagined use cases for audio-visual speech recognition is to help distinguish between phonemes which sound similar but are much more visually distinctive (e.g. /m/ and /n/). The performance of the audio-visual recognizer on each phoneme is therefore of interest. The confusion matrices for the audio-only and audio-visual recognizers are given in Figures 4.12 and Figures 4.13.

The confusion matrices show a number of interesting trends. For the /m/ and /n/ case mentioned above, the confusion matrices show that they are confused with one another less in the audio-visual recognizer. This was the expected result. Unfortunately, adding visual features also makes some phonemes less distinguishable. This can be seen for example with the phoneme /b/. In the audio-only recognizer, /b/ is confused with /dh/ more than any other phoneme except for /p/. In the audio-visual recognizer, the number of /b/s recognized incorrectly as /dh/ drops from 91 to 18. The number of /b/s incorrectly recognized as /m/, however, rises from 10 to 59. This is due to the fact that /b/ and /m/ produce the same viseme, while /b/ and /dh/ produce different visemes. The trend of phonemes becoming more confused with others in their viseme group can be seen throughout the audio-visual confusion matrix.

Another trend visible in the confusion matrices is vowels becoming more confused with consonants that typically follow them in words. For example, the phoneme /ah/ is confused

Figure 4.12: Audio-only confusion matrix from 40dB test in Figure 4.11

---------- Confusion Matrix ----------

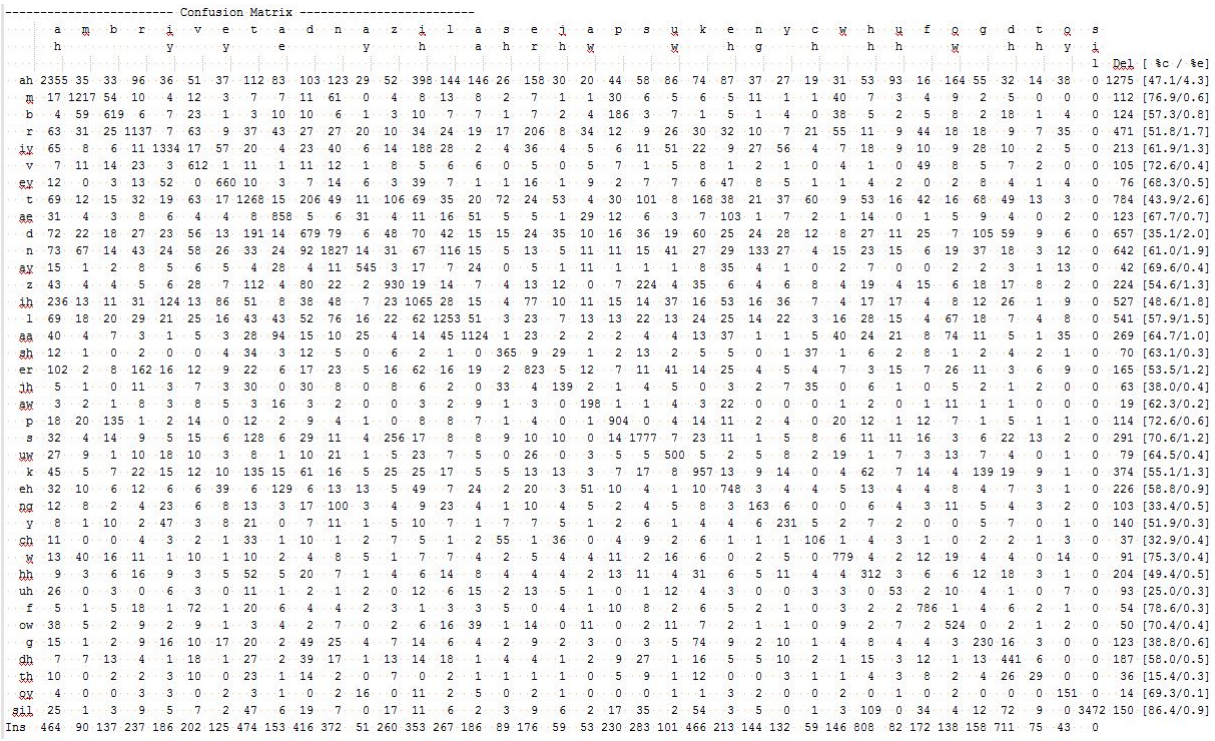
	a h	m	b	r	i y	v	e y	t	a e	d	n	a y	z	i h	l	a a	s	e h	l w	a	p	s	w	k	e h	n g	y	c h	w	h h	w	f	o w	g	d h	t h	o y	s i		
																																							l	Del	[%c / %e]
ah	2478	53	30	46	38	53	30	33	155	58	47	39	22	579	67	209	12	128	17	23	48	15	91	37	180	15	10	13	45	45	203	21	199	33	28	12	51	0	1107	[48.0/4.4]
m	9	1270	17	6	5	14	1	5	4	3	146	2	0	10	13	7	0	0	0	2	11	1	2	2	1	29	3	1	32	3	1	5	3	2	12	0	1	0	71	[78.3/0.6]
b	1	10	721	7	3	43	0	6	2	17	3	0	4	7	6	3	1	4	3	0	134	3	4	4	4	0	2	0	8	3	2	3	4	14	91	4	0	0	83	[64.3/0.7]
r	19	25	16	1571	4	19	12	15	21	11	2	14	3	21	9	6	9	249	16	24	15	0	36	15	18	5	4	17	33	17	30	11	5	9	15	4	26	0	339	[67.5/1.2]
iy	12	4	2	1	1756	6	41	4	2	11	7	3	1	107	3	2	3	28	2	2	6	2	102	4	2	16	84	3	3	17	16	2	2	5	4	0	6	0	98	[77.3/0.8]
v	4	7	50	6	0	594	1	8	1	17	7	1	4	8	4	3	0	4	2	0	11	6	12	1	1	1	3	0	6	3	2	55	8	0	25	8	0	0	85	[68.8/0.4]
ey	2	1	1	4	31	1	802	2	1	1	2	9	3	61	1	1	0	11	0	13	1	0	9	2	18	5	4	0	1	4	2	2	4	1	0	1	6	0	35	[79.6/0.3]
t	27	16	23	12	5	42	8	1647	19	253	21	10	58	37	26	6	107	10	65	6	124	77	10	93	12	9	20	95	6	61	18	50	4	35	68	47	7	0	541	[52.6/2.4]
ae	31	0	3	8	0	5	1	1	1047	1	3	49	2	6	6	58	0	2	1	12	4	0	3	5	70	0	0	1	0	7	4	1	6	4	2	3	0	0	44	[77.8/0.5]
d	28	51	78	7	8	45	2	192	5	1039	25	4	29	35	24	4	10	8	71	5	55	20	19	24	21	10	12	20	3	28	9	8	11	58	97	16	1	0	511	[49.9/1.7]
n	31	305	25	8	14	14	7	11	11	44	2401	4	3	33	43	14	3	7	4	4	21	9	14	7	12	182	6	3	12	15	6	4	10	8	12	3	6	0	319	[72.4/1.5]
ay	3	0	0	11	0	1	15	0	15	1	0	681	1	2	2	13	1	2	1	3	0	2	0	0	16	0	0	0	1	4	0	0	1	3	0	0	23	0	23	[84.9/0.2]
z	13	2	4	1	3	22	1	33	1	11	3	2	1413	12	6	0	3	2	15	0	9	217	5	9	3	1	2	2	1	4	3	17	1	2	7	15	0	0	82	[76.6/0.7]
jh	265	8	7	16	132	11	99	12	1	11	15	5	5	1446	6	1	5	55	4	8	7	4	80	6	62	9	32	10	2	17	50	3	11	6	7	2	10	0	288	[59.5/1.6]
l	43	43	11	17	7	41	7	15	14	56	34	5	3	25	1668	32	8	8	6	9	21	8	26	7	10	14	29	0	30	14	49	10	104	10	12	4	13	0	293	[69.1/1.2]
aa	29	6	3	12	1	1	0	0	193	9	0	35	0	8	16	1227	0	16	1	14	9	0	6	9	49	1	2	0	52	13	21	4	66	4	6	0	32	0	160	[66.5/1.0]
sh	1	0	0	2	0	1	1	5	0	0	0	1	2	0	1	1	542	1	17	0	1	7	3	1	2	0	1	24	0	2	3	7	0	0	0	0	0	0	22	[86.6/0.1]
er	64	5	5	183	15	5	21	4	7	6	3	3	3	67	4	2	2	1058	2	14	5	1	33	13	31	2	1	1	4	7	29	3	5	5	1	3	0	0	85	[65.4/0.9]
jh	0	0	0	1	4	3	0	13	1	9	4	1	2	1	1	0	16	0	276	1	2	4	1	1	1	1	2	45	0	1	14	0	1	0	2	0	0	0	21	[67.6/0.2]
aw	4	2	1	13	0	1	13	1	13	2	0	1	0	5	15	5	2	4	0	170	3	0	2	1	30	0	0	1	0	1	0	2	8	0	1	1	1	0	34	[56.1/0.2]
p	2	3	74	1	0	22	1	18	0	7	4	0	1	2	2	2	0	0	2	0	1010	2	0	26	1	0	1	1	4	25	1	18	4	9	43	5	0	0	69	[78.2/0.5]
s	17	1	1	0	2	3	0	29	1	2	1	3	220	11	1	2	11	6	3	0	3	2234	2	0	4	0	2	7	0	1	4	46	1	3	20	31	0	0	136	[83.6/0.7]
uw	18	3	0	2	36	3	4	5	1	4	4	0	2	27	6	0	1	16	2	2	0	2	606	1	0	4	0	1	18	2	16	0	9	4	0	0	0	0	55	[75.8/0.3]
k	11	0	3	4	1	8	7	36	0	13	4	2	2	11	2	4	1	6	0	2	54	3	2	1606	3	4	12	2	8	26	4	8	2	136	7	4	0	0	114	[80.4/0.6]
eh	40	1	2	1	2	1	42	1	93	2	2	7	2	44	4	10	0	42	0	78	10	1	4	5	914	2	5	0	6	8	15	2	12	4	8	0	4	0	124	[66.5/0.7]
ng	2	19	0	0	6	1	2	3	0	5	62	1	1	0	5	1	0	2	0	0	4	1	1	6	0	426	0	0	1	3	0	0	1	6	3	0	4	0	25	[75.3/0.2]
y	1	2	1	1	52	4	2	7	0	6	1	1	2	8	1	0	7	2	19	0	0	0	5	3	2	2	292	18	0	5	6	0	0	0	4	0	1	0	130	[64.2/0.3]
ch	1	1	1	3	0	0	1	17	1	1	0	1	1	2	0	0	36	0	26	0	0	3	0	6	2	0	0	236	1	0	4	0	1	0	1	0	1	0	12	[68.0/0.2]
w	10	18	21	9	1	5	0	6	0	1	1	2	3	4	11	11	3	3	2	1	11	2	28	1	1	1	5	0	762	10	4	12	11	13	17	0	14	0	122	[75.9/0.4]
hh	2	1	4	3	1	7	6	19	5	3	1	1	3	10	2	1	2	1	3	0	11	6	0	12	4	3	14	1	0	543	1	5	2	7	9	1	0	0	142	[78.2/0.2]
uh	23	0	0	2	4	3	1	0	0	0	1	1	0	11	5	10	1	5	10	0	3	0	8	1	2	0	1	0	6	1	136	1	12	0	0	1	4	0	52	[53.8/0.2]
f	3	0	11	0	1	14	1	10	0	1	0	0	0	0	1	2	2	0	0	0	25	9	1	4	0	0	0	0	0	4	0	894	0	0	17	17	0	0	37	[87.9/0.2]
ow	29	3	2	4	0	6	2	1	11	0	2	2	0	2	48	49	0	12	0	12	5	0	2	1	8	0	0	0	10	3	22	1	514	1	1	1	3	0	37	[67.9/0.4]
g	6	1	22	2	8	9	1	7	4	28	3	3	0	6	6	3	0	1	1	1	14	2	1	68	3	3	10	0	3	3	4	1	5	440	8	0	1	0	38	[64.9/0.4]
dh	3	14	53	2	0	27	1	12	2	32	3	0	7	4	7	1	1	1	0	1	20	14	0	2	1	1	1	0	6	3	0	11	2	9	602	15	1	0	89	[70.1/0.4]
th	3	0	2	0	1	2	0	11	0	4	1	0	11	4	1	0	0	0	0	0	0	11	10	0	3	3	0	1	0	3	5	0	28	1	0	18	83	0	0	18	[40.3/0.2]
ay	1	0	0	2	1	0	3	0	1	0	0	19	0	4	1	2	0	0	0	0	0	0	0	0	1	1	1	0	1	0	2	0	0	1	0	0	181	0	10	[81.5/0.1]
sil	15	10	74	1	3	5	1	20	4	1	4	1	2	7	2	0	6	1	1	0	156	13	1	29	7	2	1	1	4	38	1	52	2	4	86	23	0	3472	119	[85.7/0.9]
Ins	337	137	222	103	117	139	90	267	131	164	144	26	64	291	200	120	72	85	53	36	408	159	152	236	203	86	134	62	125	440	113	224	96	104	244	94	23	0		

more with consonants like /t/, /n/ and /l/ in the audiovisual recognizer. This is most likely due to the fact that early integration does not take into account the asynchrony that often exists between a phoneme and its corresponding viseme (usually due to coarticulation effects) [42].

The effects of the larger audio-visual vector length are most clearly visible on the least-frequently used phonemes /jh/, /ch/, /ng/, /uh/ and /th/. These phonemes are all in the bottom 7 in terms of frequency of use in TIMIT and TCD-TIMIT. The poor performance of /th/ and /uh/ in TIMIT was noted in Section 4.1.1. The large audio-visual vector size worsens the situation, since these phonemes already have relatively few training examples.

The audio-only and audio-visual confusion matrices at -10dB are given in Figures 4.14 and 4.15 respectively. As the SNR is progressively lowered, the number of phonemes deleted by the audio-only recognizer rises exponentially, until at -10dB, deletions account for 88% of the phonemes in the test set. The corresponding number of deletions in the audio-visual recognizer is 38%. As a result, it is difficult to determine which phonemes are most robust to noisy audio. Ideally, phonemes belonging to small, well-performing visemes from Figure 4.7 would be found to maintain good performance in noise. /aw/ is a is a useful reference in this case, as it is the only member of its viseme class (/D/). As the SNR decreases, the number of instances of /aw/ correctly recognized drops from 198 to 143. The corresponding drop in the audio-only case is

Figure 4.13: Audio-visual confusion matrix from 40dB test in Figure 4.11



	a h	m	b	r	i y	v	e y	t	a e	d	n	a y	z	i h	l	a a	s	e h	l r	a h	p	s	w w	k	e h	n g	y	c h	w	h h	w	f	o w	g	d h	t y	o i	s l	Del	[%c / %e]
ah	2355	35	33	96	36	51	37	112	83	103	123	29	52	398	144	146	26	158	30	20	44	58	86	74	87	37	27	19	31	53	93	16	164	55	32	14	38	0	1275	[47.1/4.3]
m	17	1217	54	10	4	12	3	7	7	11	61	0	4	8	13	8	2	7	1	1	30	6	5	6	5	11	1	1	40	7	3	4	9	2	5	0	0	0	112	[76.9/0.6]
b	4	59	619	6	7	23	1	3	10	10	6	1	3	10	7	7	1	7	2	4	186	3	7	1	5	1	4	0	38	5	2	5	8	2	18	1	4	0	124	[57.3/0.8]
r	63	31	25	1137	7	63	9	37	43	27	27	20	10	34	24	19	17	206	8	34	12	9	26	30	32	10	7	21	55	11	9	44	18	18	9	7	35	0	471	[51.8/1.7]
iy	65	8	6	11	1334	17	57	20	4	23	40	6	14	188	28	2	4	36	4	5	6	11	51	22	9	27	56	4	7	18	9	10	9	28	10	2	5	0	213	[61.9/1.3]
v	7	11	14	23	3	612	1	11	1	11	12	1	8	5	6	6	0	5	0	5	7	1	5	8	1	2	1	0	4	1	0	49	8	5	7	2	0	0	105	[72.6/0.4]
ey	12	0	3	13	52	0	660	10	3	7	14	6	3	39	7	1	1	16	1	9	2	7	7	6	47	8	5	1	1	4	2	0	2	8	4	1	4	0	76	[68.3/0.5]
t	69	12	15	32	19	63	17	1268	15	206	49	11	106	69	35	20	72	24	53	4	30	101	8	168	38	21	37	60	9	53	16	42	16	68	49	13	3	0	784	[43.9/2.6]
ae	31	4	3	8	6	4	4	8	858	5	6	31	4	11	16	51	5	5	1	29	12	6	3	7	103	1	7	2	1	14	0	1	5	9	4	0	2	0	123	[67.7/0.7]
d	72	22	18	27	23	56	13	191	14	679	79	6	48	70	42	15	15	24	35	10	16	36	19	60	25	24	28	12	8	27	11	25	7	105	59	9	6	0	657	[35.1/2.0]
n	73	67	14	43	24	58	26	33	24	92	1827	14	31	67	116	15	5	13	5	11	11	15	41	27	29	133	27	4	15	23	15	6	19	37	18	3	12	0	642	[61.0/1.9]
ay	15	1	2	8	5	6	5	4	28	4	11	545	3	17	7	24	0	5	1	11	1	1	1	8	35	4	1	0	2	7	0	0	2	2	3	1	13	0	42	[69.6/0.4]
z	43	4	4	5	6	28	7	112	4	80	22	2	930	19	14	7	4	13	12	0	7	224	4	35	6	4	6	8	4	19	4	15	6	18	17	8	2	0	224	[54.6/1.3]
th	236	13	11	31	124	13	86	51	8	38	48	7	23	1065	28	15	4	77	10	11	15	14	37	16	53	16	36	7	4	17	17	4	8	12	26	1	9	0	527	[48.6/1.8]
l	69	18	20	29	21	25	16	43	43	52	76	16	22	62	1253	51	3	23	7	13	13	22	13	24	25	14	22	3	16	28	15	4	67	18	7	4	8	0	541	[57.9/1.5]
aa	40	4	7	3	1	5	3	28	94	15	10	25	4	14	45	1124	1	23	2	2	2	4	4	13	37	1	1	5	40	24	21	8	74	11	5	1	35	0	269	[64.7/1.0]
sh	12	1	0	2	0	0	4	34	3	12	5	0	6	2	1	0	365	9	29	1	2	13	2	5	5	0	1	37	1	6	2	8	1	2	4	2	1	0	70	[63.1/0.3]
er	102	2	8	162	16	12	9	22	6	17	23	5	16	62	16	19	2	823	5	12	7	11	41	14	25	4	5	4	7	3	15	7	26	11	3	6	9	0	165	[53.5/1.2]
jh	5	1	0	11	3	7	3	30	0	30	8	0	8	6	2	0	33	4	139	2	1	4	5	0	3	2	7	35	0	6	1	0	5	2	1	2	0	0	63	[38.0/0.4]
aw	3	2	1	8	3	8	5	3	16	3	2	0	0	3	2	9	1	3	0	198	1	1	4	3	22	0	0	0	1	2	0	1	11	1	1	0	0	0	19	[62.3/0.2]
p	18	20	135	1	2	14	0	12	2	9	4	1	0	8	8	7	1	4	0	1	904	0	4	14	11	2	4	0	20	12	1	12	7	1	5	1	1	0	114	[72.6/0.6]
s	32	4	14	9	5	15	6	128	6	29	11	4	256	17	8	8	9	10	10	0	14	1777	7	23	11	1	5	8	6	11	11	16	3	6	22	13	2	0	291	[70.6/1.2]
uw	27	9	1	10	18	10	3	8	1	10	21	1	5	23	7	5	0	26	0	3	5	5	500	5	2	5	8	2	19	1	7	3	13	7	4	0	1	0	79	[64.5/0.4]
k	45	5	7	22	15	12	10	135	15	61	16	5	25	25	17	5	5	13	13	3	7	17	8	957	13	9	14	0	4	62	7	14	4	139	19	9	1	0	374	[55.1/1.3]
eh	32	10	6	12	6	6	39	6	129	6	13	13	5	49	7	24	2	20	3	51	10	4	1	10	748	3	4	4	5	13	4	4	8	4	7	3	1	0	226	[58.8/0.9]
ng	12	8	2	4	23	6	8	13	3	17	100	3	4	9	23	4	1	10	4	5	2	4	5	8	3	163	6	0	0	6	4	3	11	5	4	3	2	0	103	[33.4/0.5]
y	8	1	10	2	47	3	8	21	0	7	11	1	5	10	7	1	7	7	5	1	2	6	1	4	4	6	231	5	2	7	2	0	0	5	7	0	1	0	140	[51.9/0.3]
ch	11	0	0	4	3	2	1	33	1	10	1	2	7	5	1	2	55	1	36	0	4	9	2	6	1	1	1	106	1	4	3	1	0	2	2	1	3	0	37	[32.9/0.4]
w	13	40	16	11	1	10	1	10	2	4	8	5	1	7	7	4	2	5	4	4	11	2	16	6	0	2	5	0	779	4	2	12	19	4	4	0	14	0	91	[75.3/0.4]
hh	9	3	6	16	9	3	5	52	5	20	7	1	4	6	14	8	4	4	4	2	13	11	4	31	6	5	11	4	4	312	3	6	6	12	18	3	1	0	204	[49.4/0.5]
uh	26	0	3	0	6	3	0	11	1	2	1	2	0	12	6	15	2	13	5	1	0	1	12	4	3	0	0	3	3	0	53	2	10	4	1	0	7	0	93	[25.0/0.3]
f	5	1	5	18	1	72	1	20	6	4	4	2	3	1	3	3	5	0	4	1	10	8	2	6	5	2	1	0	3	2	2	786	1	4	6	2	1	0	54	[78.6/0.3]
ow	38	5	2	9	2	9	1	3	4	2	7	0	2	6	16	39	1	14	0	11	0	2	11	7	2	1	1	0	9	2	7	2	524	0	2	1	2	0	50	[70.4/0.4]
g	15	1	2	9	16	10	17	20	2	49	25	4	7	14	6	4	2	9	2	3	0	3	5	74	9	2	10	1	4	8	4	4	3	230	16	3	0	0	123	[38.8/0.6]
dh	7	7	13	4	1	18	1	27	2	39	17	1	13	14	18	1	4	4	1	2	9	27	1	16	5	5	10	2	1	15	3	12	1	13	441	6	0	0	187	[58.0/0.5]
th	10	0	2	2	3	10	0	23	1	14	2	0	7	0	2	1	1	1	1	0	5	9	1	12	0	0	3	1	1	4	3	8	2	4	26	29	0	0	36	[15.4/0.3]
oy	4	0	0	3	3	0	2	3	1	0	2	16	0	11	2	5	0	2	1	0	0	0	1	1	3	2	0	0	2	0	1	0	2	0	0	0	151	0	14	[69.3/0.1]
sil	25	1	3	9	5	7	2	47	6	19	7	0	17	11	6	2	3	9	6	2	17	35	2	54	3	5	0	1	3	109	0	34	4	12	72	9	0	3472	150	[86.4/0.9]
Ins	464	90	137	237	186	202	125	474	153	416	372	51	260	353	267	186	89	176	59	53	230	283	101	466	213	144	132	59	146	808	82	172	138	158	711	75	43	0		

from 170 to 15. Similar results can be seen for the two phonemes of viseme /A/ (/f/ and /v/) and /b/ and /p/ in viseme /C/.

4.3.4 Individual Speaker Performances

One of the trends looked for in the individual audio-visual speaker performances was whether the best speakers from the visual-only experiments (Table 4.8) had higher-than-average audio-visual scores as the noise increased, and vice-versa with the worst speakers. To evaluate this, the average accuracy of the 10 best and worst speakers from Table 4.8 is compared to the overall accuracy at each SNR in Table 4.13. Their audio-only performances are also given for comparison.

Table 4.13 supports the theory that the best visual-only speakers have above-average performance in noisy conditions, and that the converse is also true for the worst visual-only speakers. The table shows that the average audio-visual accuracy of the 10 best speakers from Table 4.8 is higher than the overall average at every SNR, while the opposite is true for the worst 10. Looking at their audio-only scores, these differences are not visible. The audio-only accuracy of the worst 10 is actually higher than that of the best 10 from 30 to 0dB. Hence the better/-worse audio-visual performances of the groups are due to their visual performances. Finally, the

Figure 4.14: Audio-only confusion matrix from -10dB test in Figure 4.11

---------- Confusion Matrix ----------

	a h	m	b	r	i y	v	e y	t	a e	d	n	a y	i h	l	a r	e	a w	p	u w	k	e h	n g	y	w	h	f	o w	d h	t h	o y	s i		
																															l	Del	[%c / %e]
ah	69	12	4	0	0	2	3	0	9	0	5	5	12	2	0	0	4	32	2	0	5	2	2	0	21	0	1	22	6	2	0	6048	[31.1/0.2]
m	0	141	2	0	0	0	1	0	3	2	4	1	0	2	1	0	3	31	0	0	2	0	0	0	3	2	1	5	2	0	0	1488	[68.4/0.1]
b	0	0	42	0	0	0	0	0	2	0	0	1	0	0	0	0	0	13	0	0	2	0	0	0	6	0	0	8	2	0	0	1128	[55.3/0.1]
r	0	5	0	1	0	2	0	0	1	0	4	2	0	2	0	0	1	33	0	0	3	0	0	0	8	0	0	10	1	0	0	2592	[1.4/0.1]
iy	0	10	0	0	2	4	1	0	4	1	3	3	1	0	0	0	2	49	0	0	3	3	0	0	10	2	0	12	2	0	0	2257	[1.8/0.2]
v	0	1	0	0	0	15	1	0	0	0	0	1	1	0	0	0	0	7	0	0	0	0	0	0	0	0	0	1	0	0	0	921	[55.6/0.0]
ey	0	2	0	0	0	0	33	0	1	0	0	1	2	0	1	0	0	13	0	0	1	0	0	0	2	0	0	3	0	0	0	983	[55.9/0.0]
t	0	10	2	0	0	3	6	15	8	0	10	5	5	1	3	0	2	67	0	0	4	1	0	0	10	0	0	10	2	0	0	3511	[9.1/0.2]
ae	0	3	0	0	0	0	1	0	130	0	0	0	2	0	0	0	0	6	0	0	2	0	0	0	3	0	0	3	2	0	0	1238	[85.5/0.0]
d	0	11	3	0	0	4	2	0	3	1	3	3	5	1	0	0	2	51	0	0	1	2	0	0	10	0	0	11	4	0	0	2476	[0.9/0.2]
n	0	8	3	0	0	2	1	0	5	0	222	2	6	5	0	0	5	41	0	0	4	3	0	0	7	1	1	13	2	0	0	3304	[67.1/0.2]
ay	0	0	0	0	0	1	1	0	0	0	0	82	2	0	1	0	1	11	0	0	0	1	0	0	0	0	0	3	1	0	0	721	[78.8/0.0]
z	0	13	3	0	0	5	1	0	9	1	15	4	6	2	1	0	4	67	0	0	5	8	0	0	7	0	0	13	5	0	0	1758	[0.0/0.3]
ih	1	5	0	0	0	1	1	0	2	0	0	0	137	2	0	0	3	12	0	0	3	0	0	0	7	0	1	3	0	0	0	2540	[77.0/0.1]
l	0	7	1	0	0	2	1	1	3	0	5	2	1	85	0	0	2	45	1	0	1	0	0	0	4	1	0	5	10	0	0	2529	[48.0/0.1]
aa	0	0	2	0	0	0	0	0	0	0	0	0	0	1	71	0	1	13	0	0	0	0	0	0	6	0	2	7	1	1	0	1900	[67.6/0.1]
sh	0	1	0	0	0	0	1	0	0	0	2	0	2	0	1	0	0	9	0	0	0	0	0	0	1	0	0	1	0	0	0	630	[0.0/0.0]
er	0	9	2	0	0	3	2	0	4	1	2	1	6	2	0	1	1	39	0	0	3	3	0	0	6	1	2	8	0	0	0	1606	[1.0/0.2]
jh	0	3	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	7	0	0	0	0	0	0	2	0	0	5	0	0	0	410	[0.0/0.0]
aw	0	1	1	0	0	0	0	0	2	0	0	0	0	0	0	0	15	5	0	0	0	0	0	0	0	0	0	1	0	0	0	312	[60.0/0.0]
p	0	1	1	0	0	1	1	0	0	0	3	0	0	1	0	0	2	508	0	0	1	0	0	0	2	0	0	3	2	0	0	834	[96.6/0.0]
s	0	18	2	0	0	2	1	0	4	2	1	7	5	6	4	0	5	59	0	1	3	2	0	0	7	0	0	9	3	0	0	2667	[0.0/0.2]
uw	0	3	0	0	0	0	1	0	0	0	2	1	0	0	0	0	2	10	0	0	0	0	0	0	4	0	0	3	0	0	0	828	[0.0/0.0]
k	0	1	3	0	0	0	0	0	0	1	3	1	1	1	1	0	0	20	0	1	0	2	0	0	2	0	0	10	1	0	0	2064	[2.1/0.1]
eh	0	2	0	0	0	0	1	0	2	0	1	0	1	0	0	0	0	7	0	0	98	0	0	1	3	1	2	1	1	0	0	1377	[81.0/0.0]
ng	0	3	0	0	0	1	2	0	1	0	2	0	0	5	0	0	0	22	0	0	1	6	0	0	1	1	0	1	0	0	0	545	[13.0/0.1]
y	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	5	0	0	0	0	0	0	0	0	0	4	0	0	0	574	[0.0/0.0]
ch	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	1	0	0	1	2	0	0	342	[0.0/0.0]
w	0	1	2	0	0	0	0	0	0	0	1	0	2	0	0	0	0	6	0	0	0	0	0	4	1	0	0	3	1	0	0	1105	[19.0/0.0]
hh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	87	0	0	0	0	0	0	742	[92.6/0.0]
uh	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	0	0	0	300	[0.0/0.0]
f	0	1	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	11	0	0	1	0	0	0	3	3	0	4	0	0	0	1027	[11.1/0.0]
ow	0	0	0	0	0	1	1	0	4	0	2	0	1	0	0	0	1	12	0	0	3	2	0	0	4	0	6	3	0	0	0	754	[15.0/0.1]
g	0	3	2	0	0	0	0	0	0	0	2	0	0	1	0	0	0	7	0	0	0	1	0	0	2	0	0	2	0	0	0	696	[0.0/0.0]
dh	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	3	0	1	128	1	0	0	811	[93.4/0.0]
th	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	2	0	0	216	[25.0/0.0]
oy	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	227	[0.0/0.0]
sil	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	3472	690	[99.8/0.0]
Ins	0	6	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	12	0	0	1	0	0	0	4	0	0	4	1	0	0		

Table 4.13: Differences between audio-visual/audio-only accuracy averages of visual-only best/worst 10 (found in Table 4.8) and overall average at each SNR. Positive differences are above-average.

SNR	40dB		30dB		20dB		10dB		0dB		-10dB	
	AV	Audio	AV	Audio	AV	Audio	AV	Audio	AV	Audio	AV	Audio
Overall average %	37.24	53.88	35.95	44.63	32.41	32.2	27.55	21.6	22.3	15.49	18.33	8.73
Diff. from overall % (visual best 10)	1.9	0.52	1.94	0.02	2.29	0.23	1.96	0.4	1.29	-0.09	1.5	0.1
Diff. from overall % (visual worst 10)	-2.32	-0.89	-1.4	2.32	-1.21	2.41	-1.62	0.89	-1.43	0.71	-1.82	0.06

Figure 4.15: Audio-visual confusion matrix from -10dB test in Figure 4.11

---------- Confusion Matrix ----------

	a h	m	b	r	i y	v	e y	t	a e	d	n	a y	z	i h	l	a a	s h	e r	j h	a w	p	s	u w	k	e h	n g	y	c h	w	h h	u	f	o w	g	d h	t h	o y	s i		
																																							l	Del	[%c / %e]
ah	1263	27	71	11	28	59	48	45	78	33	45	7	19	92	30	64	36	23	16	50	165	22	39	69	121	99	29	10	36	345	10	130	36	12	410	27	1	0	2664	[35.0/3.8]
m	22	229	82	6	5	25	9	8	20	3	10	3	8	18	8	15	3	3	2	12	189	6	8	8	29	21	7	3	15	81	0	35	6	1	117	6	0	0	671	[22.4/1.3]
b	9	9	353	2	3	16	11	6	7	1	5	3	2	18	4	9	3	4	4	6	150	6	2	6	20	7	5	0	10	40	1	13	3	2	65	2	0	0	397	[43.7/0.7]
r	42	11	40	285	3	71	16	13	37	8	22	5	10	19	11	18	5	17	2	24	95	3	23	26	56	42	11	4	44	178	1	94	20	7	187	15	2	0	1198	[19.4/1.9]
iy	35	13	27	3	195	38	37	16	25	20	31	2	7	53	20	23	9	7	3	39	85	2	9	25	66	43	26	6	7	197	2	62	12	8	223	16	0	0	977	[14.0/2.0]
v	5	4	12	5	1	364	8	6	9	1	1	0	4	3	1	5	2	3	2	12	28	3	1	3	9	9	0	2	3	50	0	60	7	1	52	5	0	0	267	[53.5/0.5]
ey	9	4	10	4	5	5	228	9	16	2	7	4	3	28	5	7	4	1	4	13	23	3	6	14	51	26	5	4	5	63	0	9	2	4	64	5	0	0	390	[35.0/0.7]
t	33	15	43	14	9	61	28	419	41	22	44	5	16	28	20	17	28	10	12	17	101	8	17	35	69	66	21	18	21	294	4	109	15	11	354	25	0	0	1625	[20.4/2.7]
ae	10	3	10	4	3	14	9	5	524	2	9	1	2	4	1	23	5	1	2	23	20	2	5	8	51	14	5	4	4	71	0	17	1	3	60	6	0	0	464	[56.6/0.7]
d	22	13	27	8	8	61	11	47	24	153	31	1	17	40	9	26	15	6	7	21	102	11	16	35	60	54	13	10	12	159	2	83	12	7	243	24	0	0	1203	[11.0/2.0]
n	29	15	43	14	17	63	25	25	44	14	483	4	12	40	15	20	11	6	8	26	102	11	14	40	84	67	15	15	19	236	4	95	11	21	336	18	1	0	1632	[24.1/2.5]
ay	10	0	10	5	5	12	11	8	24	5	12	101	1	10	6	16	2	3	3	14	34	1	3	11	25	15	1	4	2	58	0	8	4	1	51	5	0	0	344	[21.0/0.6]
z	25	7	22	7	6	26	19	30	20	17	21	2	100	23	21	24	15	6	5	14	53	18	15	24	39	27	16	11	10	131	3	74	7	8	203	19	1	0	858	[9.4/1.6]
ih	48	8	24	7	15	31	35	25	32	10	20	2	6	431	9	23	8	18	8	31	71	4	16	21	66	38	18	5	9	184	2	53	12	4	210	15	0	0	1199	[28.4/1.8]
l	32	14	44	13	7	36	30	22	33	16	42	5	4	27	236	27	12	3	3	21	95	12	13	26	58	48	19	9	6	261	2	34	10	10	247	19	0	0	1210	[15.8/2.1]
aa	26	10	17	3	3	22	9	8	41	3	21	1	2	12	7	603	5	7	2	12	49	3	5	13	50	23	6	3	13	94	2	21	14	2	115	11	2	0	765	[48.6/1.0]
sh	7	0	5	2	2	5	5	10	4	7	3	0	6	6	3	1	57	1	5	3	21	1	2	13	11	13	4	8	2	52	0	21	2	2	62	4	0	0	298	[16.3/0.5]
er	41	15	17	15	7	40	11	14	24	7	26	1	10	17	7	21	8	99	2	17	73	7	23	19	38	27	10	3	7	115	2	60	11	4	135	14	1	0	754	[10.4/1.4]
jh	3	1	7	2	0	7	3	7	10	3	4	1	2	3	1	3	6	0	17	2	14	0	4	3	12	7	2	10	1	23	0	17	5	0	47	1	0	0	201	[7.5/0.3]
aw	3	3	3	4	0	3	2	1	7	2	1	0	1	4	0	6	0	0	0	143	6	0	1	1	11	1	0	1	2	20	0	2	2	0	23	1	0	0	83	[56.3/0.2]
p	7	5	57	1	0	8	3	5	8	3	4	0	2	9	2	8	3	1	0	1	871	4	1	2	8	2	0	0	6	23	0	10	0	1	42	2	0	0	261	[79.3/0.4]
s	31	18	31	14	8	35	21	35	35	27	40	3	26	29	9	16	22	9	9	13	70	162	17	30	62	57	19	16	15	193	3	64	11	6	314	24	1	0	1313	[10.8/2.2]
uw	17	4	7	3	2	17	4	5	6	6	11	1	4	11	1	3	4	4	0	7	32	2	152	8	4	11	4	2	17	48	0	18	10	4	61	3	0	0	361	[30.8/0.6]
k	16	3	19	5	5	33	12	27	20	13	14	0	8	17	6	12	6	4	0	9	53	2	6	274	38	28	16	5	16	214	2	51	8	14	229	13	0	0	914	[22.9/1.5]
eh	11	5	10	2	2	6	14	9	47	4	10	1	1	17	3	19	3	3	5	20	30	1	4	10	575	10	4	7	2	71	1	15	5	1	72	4	0	0	494	[57.3/0.7]
ng	9	3	4	2	6	15	9	5	6	2	6	0	2	3	6	4	1	1	0	6	18	3	4	5	15	100	3	1	2	48	2	10	2	2	61	5	0	0	220	[27.0/0.4]
y	7	2	7	1	5	2	4	5	2	1	8	0	0	5	1	3	3	2	0	2	8	3	0	5	3	8	66	2	7	37	0	11	2	4	45	4	1	0	319	[24.8/0.3]
ch	11	2	2	0	1	2	1	1	0	2	2	0	3	4	0	2	10	0	1	0	15	3	1	6	7	5	4	26	0	21	0	7	1	1	37	1	0	0	180	[14.5/0.2]
w	9	7	26	5	2	13	5	4	4	1	5	0	2	6	5	6	5	1	0	7	84	3	8	6	10	11	4	1	434	36	4	25	4	0	41	3	0	0	339	[55.1/0.6]
hh	8	1	2	1	1	11	4	2	2	1	6	0	2	3	4	3	1	0	1	2	17	0	3	3	9	1	1	2	2	479	0	7	3	2	47	0	0	0	205	[75.9/0.2]
uh	8	1	2	1	0	8	2	3	1	1	8	0	0	5	4	4	3	4	1	1	13	3	4	0	1	6	2	1	1	16	3	9	2	1	25	1	0	0	160	[2.1/0.2]
f	1	1	6	1	2	40	3	2	6	2	4	0	0	2	1	6	1	1	1	3	19	1	4	5	11	13	1	4	2	32	0	566	2	2	46	3	1	0	259	[71.2/0.4]
ow	8	3	8	8	2	15	4	6	10	1	3	1	0	10	4	13	0	5	1	9	21	0	17	11	15	13	2	2	11	38	1	21	216	2	47	3	0	0	263	[40.7/0.5]
g	6	1	2	3	8	11	4	6	4	4	7	3	2	5	3	8	3	0	1	4	28	4	0	12	17	16	9	3	4	85	1	16	2	31	72	9	0	0	322	[7.9/0.6]
dh	2	2	5	1	2	3	1	5	2	4	4	0	1	1	0	1	3	0	2	3	13	0	0	2	7	6	3	0	2	21	1	11	2	2	621	3	0	0	212	[84.4/0.2]
th	3	0	3	2	0	4	2	0	0	2	2	0	2	1	0	3	2	0	0	1	7	1	4	0	7	2	2	0	1	14	1	7	2	1	26	18	0	0	104	[15.0/0.2]
oy	9	0	3	2	1	2	4	1	2	0	1	0	0	4	1	3	3	1	0	1	9	0	8	3	6	2	1	0	3	28	0	8	4	1	12	0	3	0	106	[2.4/0.2]
sil	12	1	9	5	1	8	1	4	5	8	9	1	3	7	4	10	1	3	1	3	30	5	2	13	10	16	5	1	4	77	0	32	3	1	76	5	0	3472	321	[90.2/0.6]
Ins	73	33	100	32	19	137	44	63	75	40	68	8	31	67	29	63	26	15	9	46	229	18	26	89	121	97	35	19	45	613	6	170	30	17	570	46	1	0		

audio-only and audio-visual performance of each speaker at each SNR is visible on the scatter plots of Figures 4.16 and 4.17. A full list of individual speaker scores is given in Appendix D.

4.4 Summary

This chapter is concerned with the audio, visual and joint audio-visual baseline results obtained on TCD-TIMIT using basic state-of-the-art methods. Section 4.1 compares audio-only baseline results on TIMIT and TCD-TIMIT. Speaker-dependent and independent audio-only baselines are obtained on TCD-TIMIT. The TIMIT baseline is compared to TIMIT baselines in the literature to ascertain whether the training process was reliable. The baseline is found to be relatively low, but comparable to others. Hence, the recognizer's training process is deemed to be reliable.

Section 4.2 discusses the visual-only recognition experiments undertaken. First, visual-only results were obtained for several DCT vector and HMM state lengths to determine the best combination. This combination was then used to obtain speaker-dependent and independent visual-only baselines for TCD-TIMIT. The performances of the recognizers and individual speakers are discussed.

Audio-visual baselines are discussed in Section 4.3. The performance of an audio-visual

Figure 4.16: Audio-visual performance of each speaker at each SNR

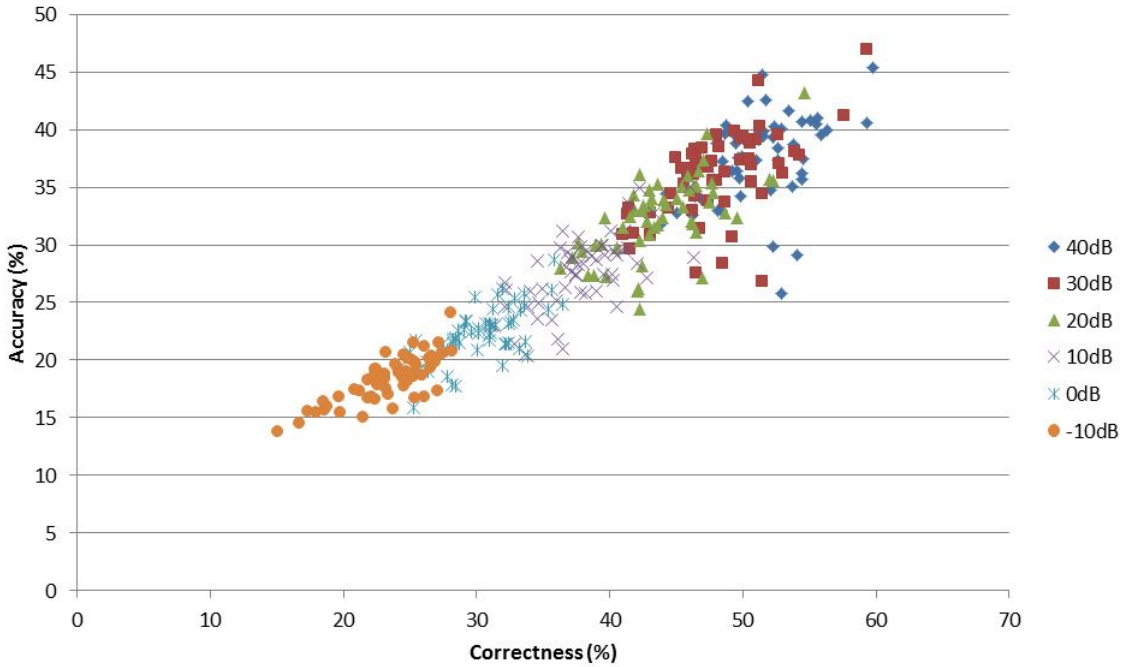
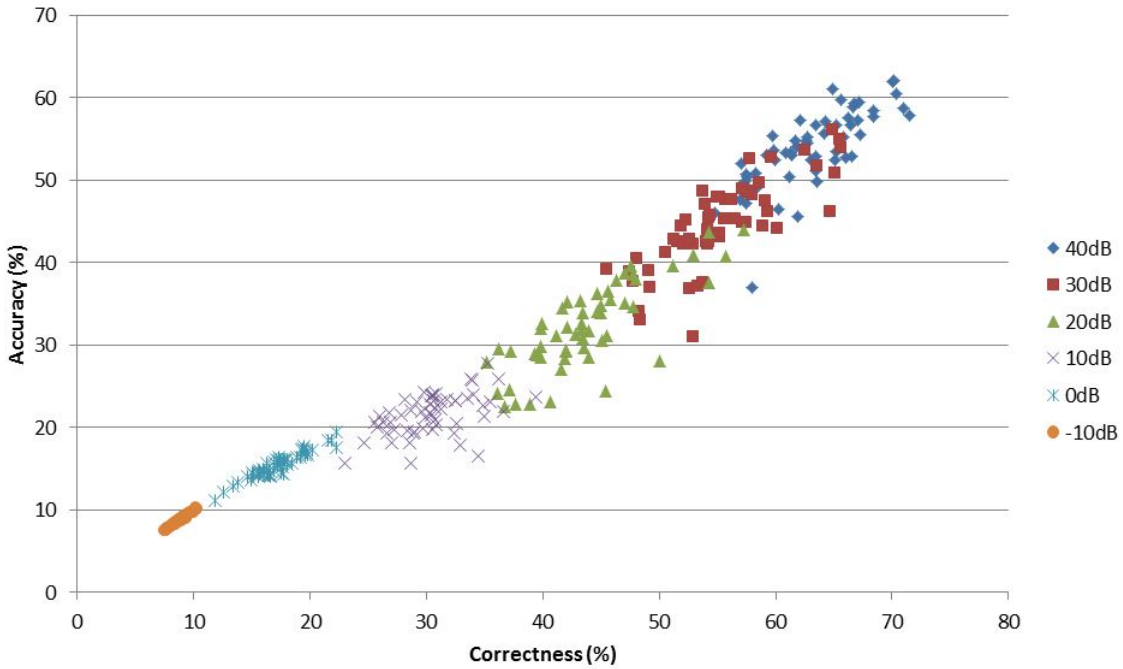


Figure 4.17: Audio-only performance of each speaker at each SNR



recognizer on increasingly noisy audio is compared to that of an audio-only recognizer. The results are compared to others in the literature for this type of experiment. Individual speaker performances are also discussed.

All of the baselines in this chapter were obtained on 56 of the 59 volunteers in the main (volunteer) part of the TCD-TIMIT database. Chapter 5 introduces the second (lipspeaker) part of the database and compares the performance of lipspeakers to volunteers.

5

Lipspeakers of TCD-TIMIT

5.1 Inclusion of Lipspeakers

A brief introduction to the concept of lipspeaking was given in Section 2.4. As explained in that section, lipspeakers undergo training to make themselves easier for human lipreaders to lipread. They often teach lipreading classes, as well as serving as translators for lipreaders. Since human lipreaders find them easier to lipread than regular speakers, it is hypothesized that they may provide insight as to the best features to use for visual speech recognition.

To investigate this possibility, 3 lipspeakers were recorded as an additional part of the TCD-TIMIT database. A comparison of lipspeaker versus regular volunteer visemes can be seen in Figure 5.1. Some information about the lipspeakers was given in Section 3.2.3. The recording setup used was the same as the one described in Section 3.1.3, but the lipspeakers recorded almost 4 times as much material as the 59 volunteers (377 sentences vs 98). Parts of the volunteers' scripts were re-used to create the lipspeaker scripts. In creating the lipspeaker scripts, the only sentences duplicated were SA1 and SA2. All of the other sentences are unique to each lipspeaker. The post-processing, creation of label files and feature extraction of the lipspeaker data was done following the same methods detailed in Chapter 3. All experiments in this chapter use 4-state HMMs and 44-coefficient DCT vectors (plus 1st and 2nd derivatives) unless otherwise specified.

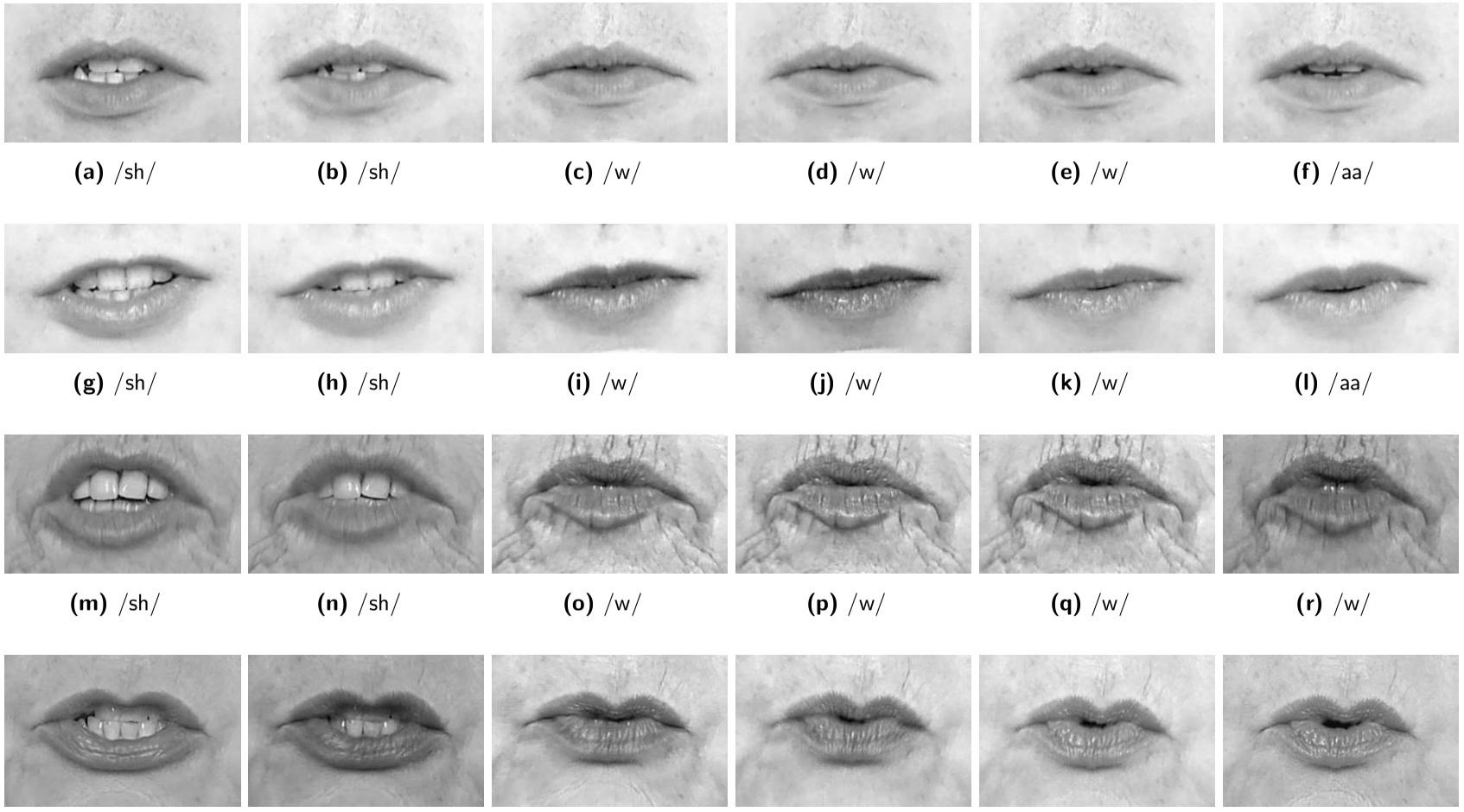
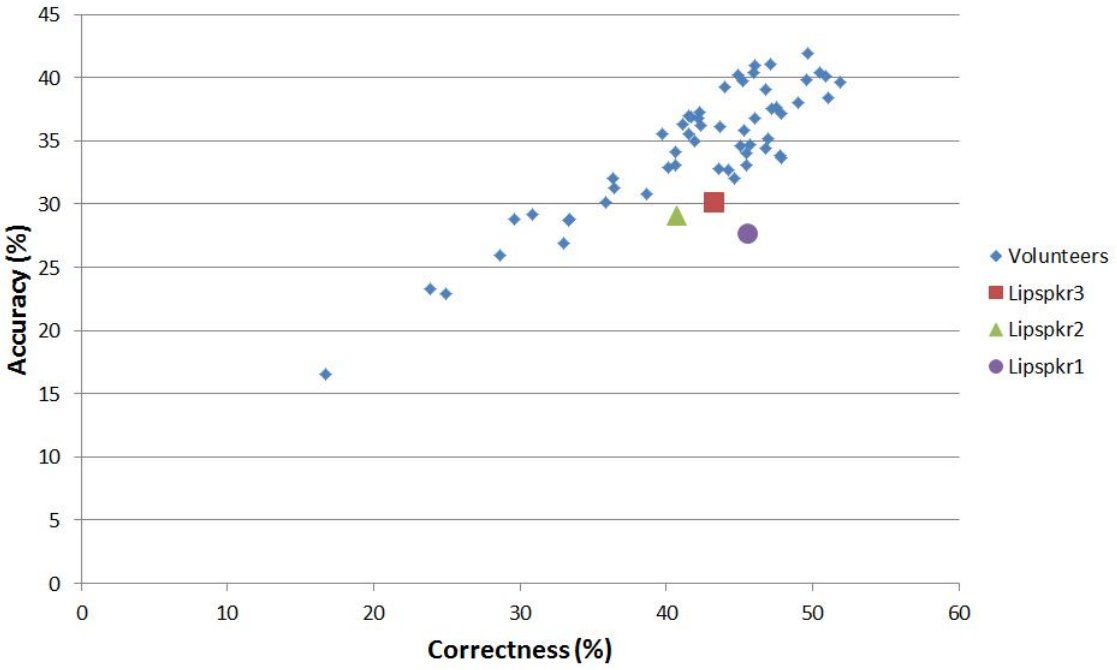


Figure 5.1: ROIs extracted from two volunteers (06M and 37F) and Lipspeakers 2 and 3 for a particular phoneme sequence: /sh/ /w/ /aa/. This sequence was found in the middle of the words "wash water" in TIMIT's SA1 sentence, which was said by all speakers in TCD-TIMIT. The first and second rows are volunteers (06M and 37F), the 3rd row is Lipspeaker 2 and the 4th row is Lipspeaker 3. Note that Lipspeaker 2 is still on the /w/ phoneme by the 6th frame. The images show how the lipspeakers articulate visemes more than the volunteers.

Figure 5.2: Performance of lipspeakers and volunteers on volunteer-trained HMMs of Section 4.2.1



5.2 Visual-Only Experiments

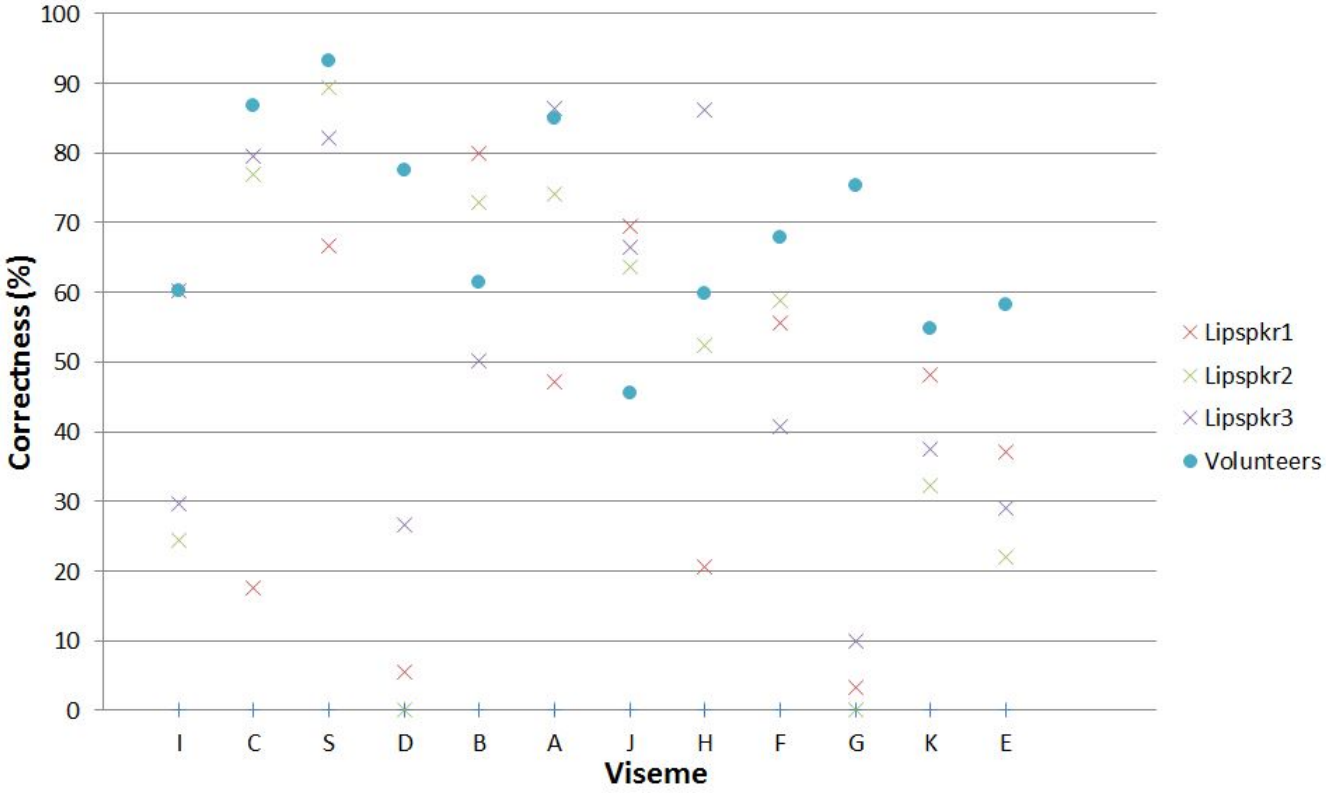
5.2.1 Lipspeakers Tested on Volunteer-trained HMMs

The first experiment run on the lipspeakers' visual data was to test it all on the volunteer-trained HMMs of Section 4.2.1. The results were then compared to the individual volunteer results. This was not exactly a fair comparison, since the HMMs were speaker-dependent (i.e. they had seen some data from every volunteer during training), whereas they had not seen the lipspeakers before. Despite this, it was considered a useful baseline. The results are given in Figure 5.2.

Figure 5.2 shows that the lipspeakers did not perform particularly well compared to the volunteers. The average accuracy of the whole set was 34.23%, so the lipspeaker accuracies of 30.16%, 29.12% and 27.63% put them all below average. The lipspeakers' correctness scores are closer to the average, but this is misleading since correctness does not include insertion penalties (see Section 2.1.5). From the HTK output, lipspeakers had higher levels of insertions (18%, 12% and 13%) than volunteers (7%), most likely because they were not seen during training. To see whether the lipspeakers performed better or worse than average on any particular viseme, their confusion matrices were compared to the overall volunteer confusion matrix of Figure 4.7. The recognition scores from the confusion matrices are compared in Figure 5.3.

As Figure 5.3 illustrates, the correctness scores vary significantly between each lipspeaker

Figure 5.3: Correctness scores from the volunteer confusion matrix of Figure 4.7 compared to each lipspeaker's confusion matrix.

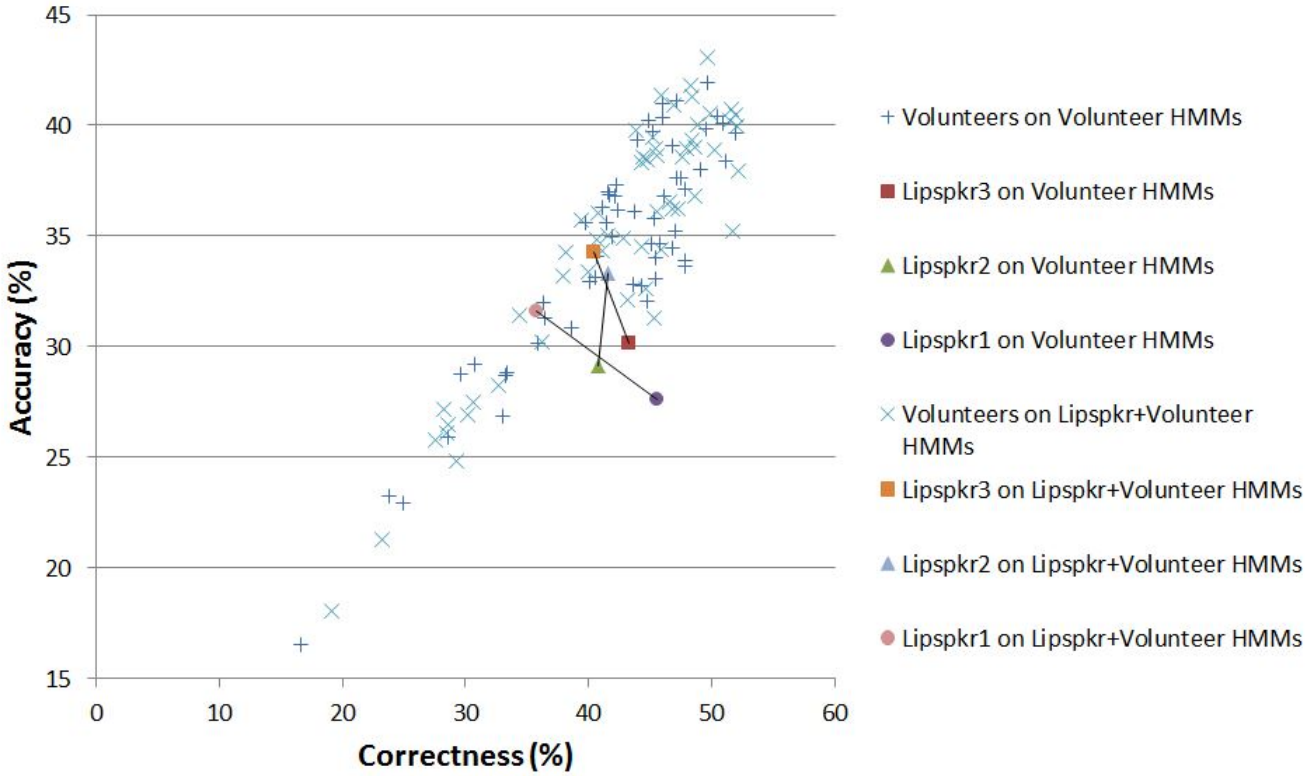


and the volunteer averages. The high rates of insertions for lipspeakers are a large factor in these differences and make it difficult to compare lipspeakers and volunteers. The differences between lipspeakers, however, are more surprising than the differences between lipspeakers and volunteers. It was expected (according to Section 2.4) that the visemes produced by each lipspeaker would be similar to one another and thus produce similar recognition scores. The lipspeakers perform particularly poorly on the more infrequent visemes /D/, /G/, /K/ and /E/, which is understandable, since a single incorrect insertion of one of these visemes has a larger effect on its recognition score.

5.2.2 HMMs Trained on Lipspeakers and Volunteers

The fact that the experiment above was performed on volunteer-trained speaker-dependent HMMs biases the results in favour of the volunteers. Therefore it is difficult from that experiment alone to determine whether lipspeakers have any advantage over volunteers. The next logical experiment was to train new speaker-dependent HMMs using the same amount of training and test data from the volunteers and lipspeakers. The expected outcome was that the lipspeaker performances would improve compared to those in Figure 5.2. The results of this experiment

Figure 5.4: Performance of lipspeakers and volunteers on speaker-dependent HMMs trained on data from both, versus Figure 5.2

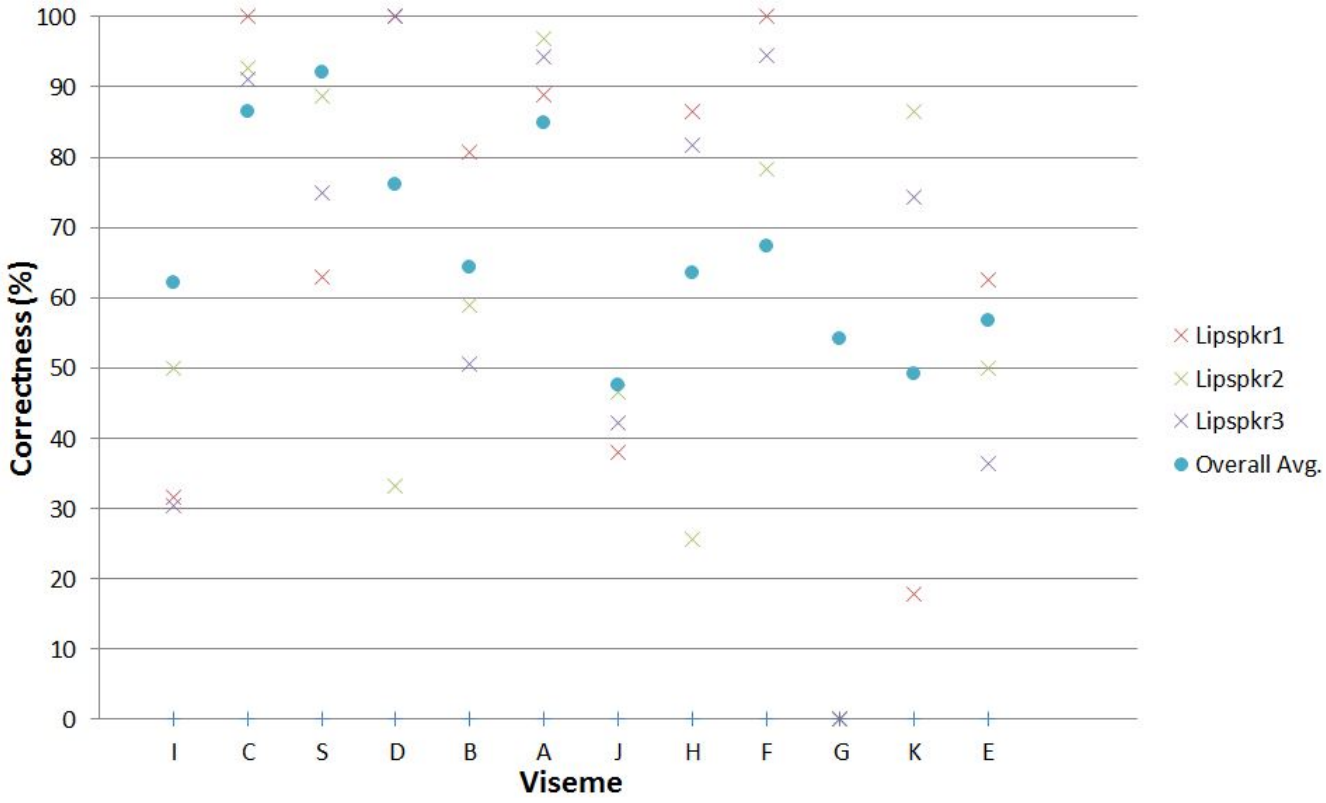


are compared to the previous results in Figure 5.4.

As Figure 5.4 shows, the lipspeakers' results did improve when they contributed to the recognizer's training data. Each lipspeaker's accuracy improved by roughly 4%. This brings them all closer to the new overall average of 34.98% (which only changed by .75%), but they are still below-average. Lipspeakers 1 and 3 also saw their correctness scores drop by 10% and 3% respectively, but again the cause for this was found to be their levels of insertions, which had dropped to a level similar to that of the volunteers. For example, the insertion rate for Lipspeaker 1 dropped from 18% of the total number of visemes, to 4%, which explains the 10% reduction in correctness. The reduction in insertions brought with it an increase in the number of deletions, from 22% to 42%. Meanwhile, the average level of insertions amongst the volunteers remained virtually unchanged in both experiments at approximately 7%. The lipspeaker confusion matrices were again compared to the overall confusion matrix in Figure 5.5.

The overall scores in Figure 5.5 have not changed significantly from those in Figure 5.3. This highlights the fact that despite including the lipspeakers in the training data, their influence on the recognizer was low. This is due to each volunteer and lipspeaker contributing the same amount of training and test sentences (67 and 31 respectively). As for the lipspeakers' individual

Figure 5.5: Comparison between correctness scores in lipspeakers' confusion matrices versus overall (lipspeakers+volunteers) confusion matrix.



scores, the higher deletion rates caused scores on some visemes to increase by a large margin compared to Figure 5.3. For example, Lipspeaker 1's correctness score on /C/ increased from 17.6% to 100%. Some part of this increase is due to Lipspeaker 1's deletion rate rising from 22% to 42%. Hence it is unclear from the confusion matrices whether the lipspeakers' viseme performances became more consistent with one another compared to Figure 5.3.

5.2.3 Lipspeaker-only VS Volunteer-only Recognizers

The experiments above showed the lipspeakers' performances improving when they had some influence on the recognizer's training. The next experiment was to determine if performances would further improve when they had a much higher (i.e. total) influence on a recognizer's training. A new speaker-dependent recognizer was trained and tested on lipspeaker data only. A 67-33 train-test split was used, meaning that each of the 3 lipspeakers contributed 251 sentences to the training set and 126 to the test set. The recognizer was trained and tested using the exact same settings as Section 4.2. The results are compared to the speaker-dependent volunteer results of Table 4.7 in Table 5.1.

Table 5.1: Lipspeaker results on lipspeaker-only trained HMMs vs volunteer results on volunteer-only trained HMMs

	Lipspeakers		Volunteers	
	Train set	Test set	Train set	Test set
%correct	60.38	57.85	42.69	41.98
%accuracy	56.45	52.74	36.05	34.54

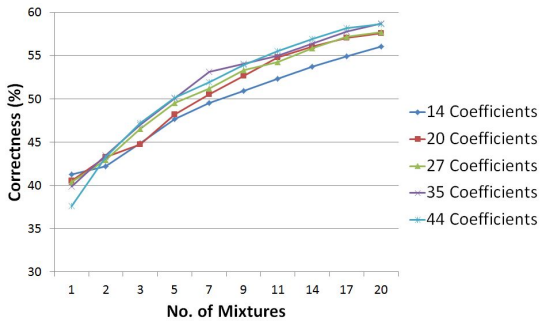
The results of Table 5.1 show the lipspeaker recognizer outperforming the volunteer recognizer by a large margin. Test-set accuracy is 18 percentage points higher, while training-set accuracy is 20 points higher. There are some obvious caveats to these results: the lipspeaker recognizer was trained and tested on 3 people versus the volunteer recognizer's 56, and saw far more data per speaker (251 vs 67 sentences each). Ideally, the influence of these differences could be estimated by training a recognizer using the same amount of data from 3 volunteers, but unfortunately this data does not exist. Nevertheless, the size of the margin between the two sets of results supports the theory that lipspeakers are easier for automatic systems to lipread than volunteers. To determine whether the lipspeakers held this advantage regardless of the number of HMM states or DCT coefficients, a recognizer was trained on lipspeaker data for the same combinations of state lengths and coefficients used in Section 4.2.1. Graphs of the results are given in Figure 5.6.

As was the case with Figure 4.3, Figure 5.6 shows that there is very little difference between 3, 4 and 5-state HMMs and between DCT coefficient lengths above 20. The lipspeaker scores are higher than the volunteer scores in every case. This indicates that their advantage is persistent over these parameter choices.

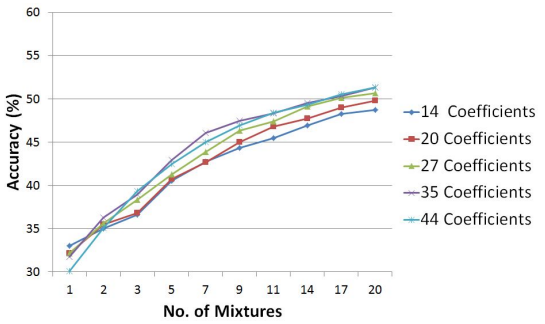
Table 5.2: Individual Lipspeaker accuracy results from recognizers trained on volunteers (Section 5.2.1), lipspeakers+volunteers (Section 5.2.2) and lipspeakers (this section).

	Volunteer HMMs	Lipspeaker+Volunteer HMMs	Lipspeaker HMMs
Lipspeaker 1	27.63	31.6	47.58
Lipspeaker 2	29.12	33.3	57.79
Lipspeaker 3	30.16	34.25	53.65

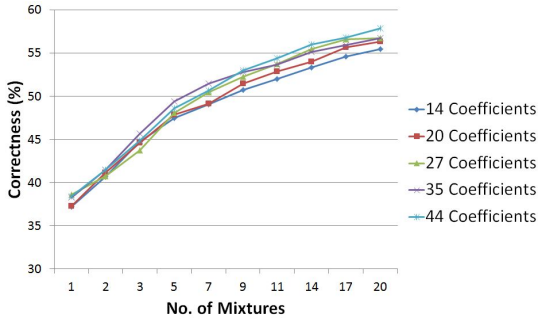
Table 5.2 compares the test-set accuracy for each lipspeaker between the experiments of Sections 5.2.1, 5.2.2 and this section. The table highlights the jump in each lipspeaker's accuracy over the three experiments. Comparing the performances on the lipspeaker-trained HMMs to the best volunteer performances on the volunteer-trained HMMs (Table 4.8), even the worst lipspeaker score is higher than the best volunteer score. Nevertheless, there is a considerable



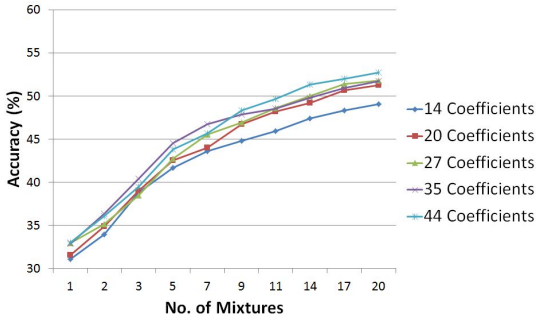
(a) Correctness vs No. of Mixtures (3-state HMMs)



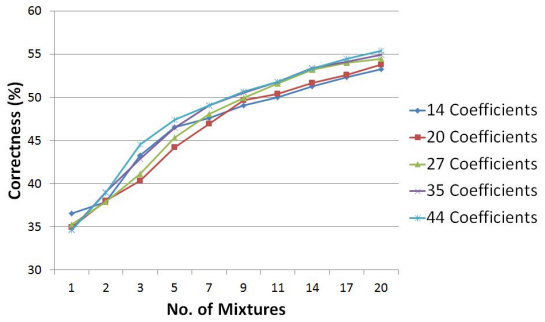
(b) Accuracy vs No. of Mixtures (3-state HMMs)



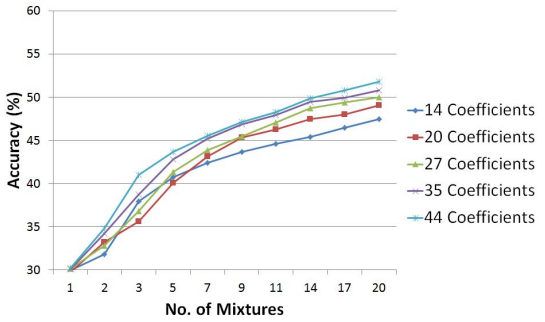
(c) Correctness vs No. of Mixtures (4-state HMMs)



(d) Accuracy vs No. of Mixtures (4-state HMMs)



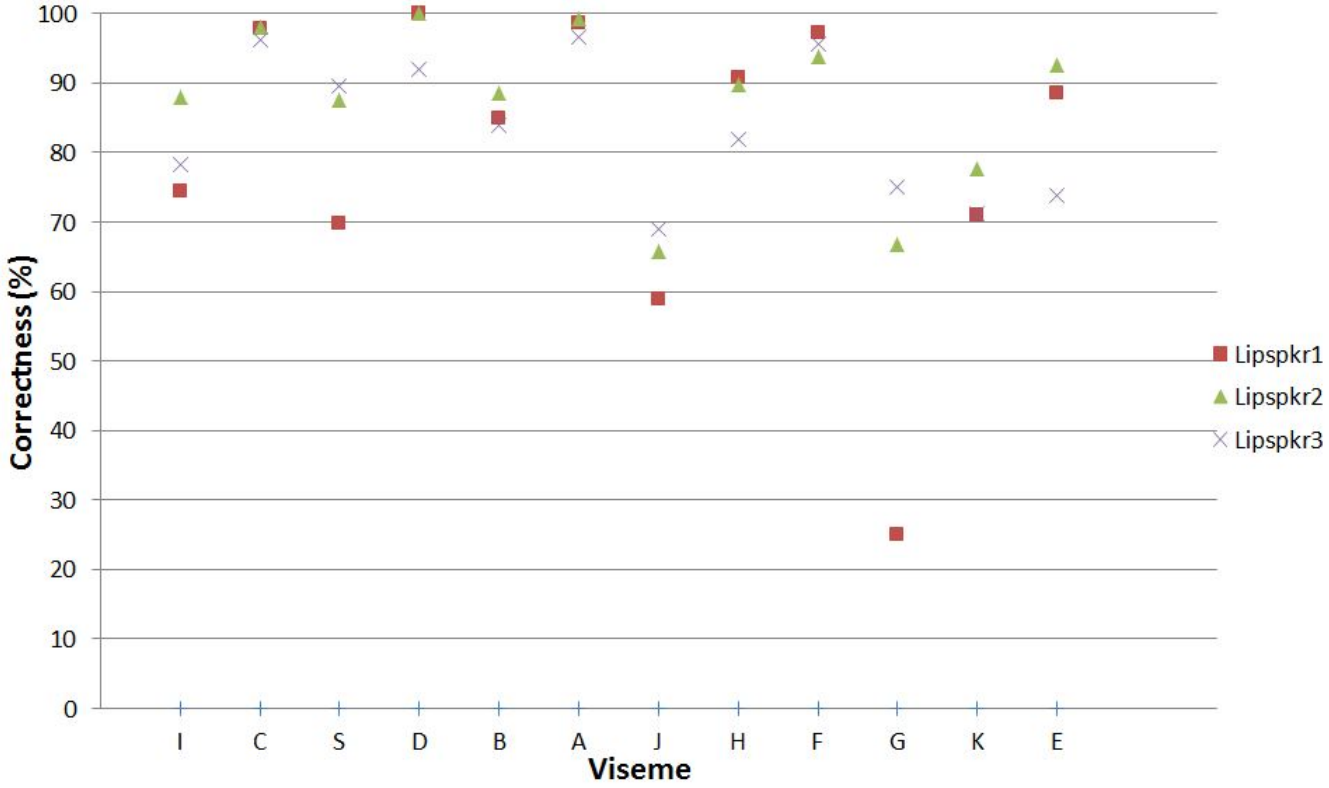
(e) Correctness vs No. of Mixtures (5-state HMMs)



(f) Accuracy vs No. of Mixtures (5-state HMMs)

Figure 5.6: Lipspeaker recognition results using different combinations of DCT vector and HMM state lengths. The graphs are very similar between the HMM state lengths, but the accuracy is highest at 4 states, while the correctness is highest at 3 states. 44 DCT coefficients consistently provides the highest scores, but not by large margins. The graphs are very similar to those of Figure 4.3, as expected.

Figure 5.7: Comparison between the lipspeakers' confusion matrix scores.



gap between the worst and best lipspeaker scores. Lipspeaker 1 has the lowest accuracy in all three tests. One noticeable difference between Lipspeaker 1 and the others is that she spoke more slowly. Her average clip length is 6.23s, compared to 5.07s for Lipspeaker 1 and 5.13s for Lipspeaker 3. Also, the other two lipspeakers agreed that Lipspeaker 1 places the most emphasis on each viseme. To see the difference in each lipspeaker's performance per viseme, their confusion matrix scores are compared in Figure 5.7.

As Figure 5.7 shows, the lipspeaker scores on each viseme are much more similar compared to Figures 5.3 and 5.5. The fact that the recognizer was trained only on the three lipspeakers is at least partially responsible for this, but the similar scores on most visemes also suggest that the lipspeakers produce similar visemes. The scores are high overall, particularly on the visemes ranked highest in terms of visibility by Jeffers and Barley (see Table 2.4). The overall deletion and insertion rates were low compared to the previous experiments, at 29% and 5% respectively. Lipspeaker 1's deletion rate was 4% higher than the others, which contributed to her comparatively low score. Also, she had the lowest scores on the most common visemes /I/, /J/ and /S/, which further affected her overall score.

5.2.4 Volunteers Tested on Lipspeaker-trained HMMs

It was speculated that since the lipspeakers supposedly produce more well-defined visemes than regular speakers, perhaps the recognizer trained only on lipspeaker data would produce higher recognition scores on the volunteers. To test this, the 17-volunteer test set used in Section 4.2.2 was used as test data on the lipspeaker-trained recognizer. The results are compared to the equivalent results on the volunteer-trained HMMs from Section 4.2.2 in Table 5.3.

Table 5.3: Volunteer results on volunteer-trained HMMs (Section 4.2.2) and lipspeaker-trained HMMs (this section)

	Correctness (%)	Accuracy (%)
Volunteer-trained HMMs	46.78	34.77
Lipspeaker-trained HMMs	29.87	28.54

The comparison in Table 5.3 is not entirely fair, since the speaker-independent recognizer in Section 4.2.2 saw a lot more data from different speakers (3822 sentences from 39 volunteers) than the lipspeaker-trained recognizer (753 sentences from 3 lipspeakers). Nevertheless, the results are much lower on the lipspeaker-trained recognizer. The fact that it was less speaker-independent inevitably affected the results negatively, but the results reject the speculation that the lipspeaker-trained HMMs are also "better" for volunteers.

To further analyse the results, the volunteers' confusion matrix from the lipspeaker-trained recognizer is given in Figure 5.8 (the equivalent confusion matrix for the volunteer-trained recognizer was given in Figure 4.9). The most evident issue is the number of deletions, which account for 54% of total visemes. The equivalent number of deletions in Figure 4.9 was 28%. As a result, it is difficult to evaluate individual viseme performances from the confusion matrix. The deletion rate is especially high for infrequent visemes such as /A/, /D/, /E/, /F/, /G/ and /H/. The most frequent visemes (/I/, /J/, /B/ and /S/), as well as /K/ strangely, are the only visemes with a correctness score above 10%. For the frequent visemes to have the highest recognition rates is understandable, since there was more data to train their models, but /K/ is quite infrequent. Another unusual feature of the confusion matrix is the number of insertions and substitutions for /S/, the silence viseme. Since /S/ is mapped from the silence phoneme in label files, and since the mouth may not necessarily be in a "silence" position when there is audio silence, it is likely that a wide variety of mouth positions are mapped to /S/, which might explain the insertions and substitutions. However, this was not an issue in the volunteer-trained recognizer's confusion matrix (Figure 4.9) or any other confusion matrix for that matter.

As in Table 4.11, the 5 best and worst individual speakers are listed in Table 5.4 for comparison. Of interest was whether the same speakers were present in the top and bottom 5 in each experiment. Speakers 55F and 49F are still in the top 5, and 34M, 45F and 56M are still in the bottom 5 (as in 4.11). However, 18M and 15F have moved from the top to the bottom

Figure 5.8: Confusion matrix for volunteer results on lipspeaker-trained HMMs in Table 5.3

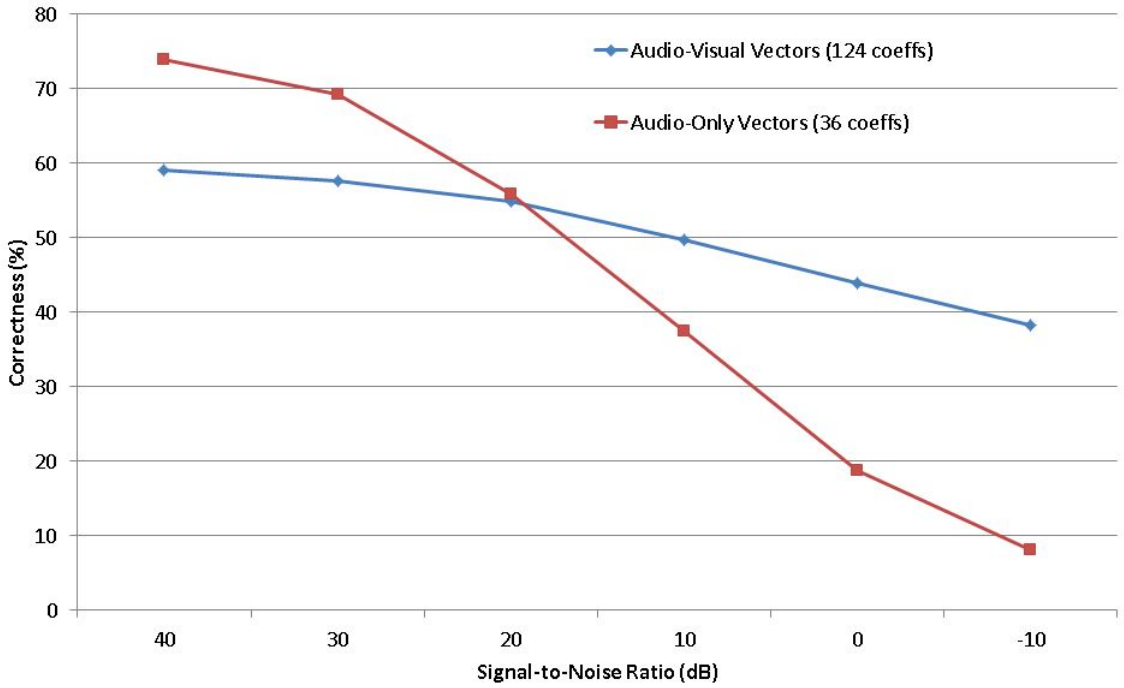
```
====================== HTK Results Analysis =======================
  Date: Fri Nov 01 15:54:42 2013
  Ref : TCDTIMITjeffersVisemesOffset.mlf
  Rec : >L\4states\44coeffs\Results\split1Results\TCDTestRecognition\L1_60.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=1666, N=1666]
WORD: %Corr=29.87, Acc=28.54 [H=16049, D=29145, S=8531, I=715, N=53725]
------------------------ Confusion Matrix -------------------------
       I   C   S   D   B   A   J   H   K   E  Del [ %c / %e]
   I 8620   5 1241   0 108   0  42   0 389  16 6269 [82.7/3.4]
   C   43  48  504   0  36   0  30   0 197  10 2942 [ 5.5/1.5]
   S   38   0 3510   0   0   0   1   0  11   0  396 [98.6/0.1]
   D   15   1   41   0   2   0   0   0  13   1  276 [ 0.0/0.1]
   B   65   2  751   0 1090  1  33   1 280  14 4229 [48.7/2.1]
   A   19   2  276   0  16   0  18   0 104   7 1444 [ 0.0/0.8]
   J  101   6 1453   0  95   1 1466  5 486  22 6652 [40.3/4.0]
   H   72   1  685   0  46   0  30  30 195  10 3394 [ 2.8/1.9]
   F   10   1  234   1  26   1  11   1  69   9 1007 [ 0.0/0.7]
   G    5   0   22   0   3   0   1   0   8   1  173 [ 0.0/0.1]
   K   24   1  296   0  18   0  12   1 1263  3 1488 [78.1/0.7]
   E    4   1  164   0  10   0  10   0  43  22  875 [ 8.7/0.4]
Ins    41   0 492   0  30   0  21   5 118   8
===================================================================
```

5, while 25M has moved from the bottom to the top. Hence, the implication is that a good performance on the volunteer-trained recognizer does not guarantee a good performance on the lipspeaker-trained recognizer. This highlights the difference between both recognizers.

Table 5.4: The 5 best and worst speakers from the volunteer results on the lipspeaker-trained recognizer

Top 5 Speakers			Bottom 5 Speakers		
Speaker	Correctness	Accuracy	Speaker	Correctness	Accuracy
25M:	30.52	30.20	45F:	19.90	19.58
55F:	32.98	30.90	34M:	23.33	22.75
49F:	35.37	32.04	15F:	26.78	24.69
08F:	38.25	35.60	56M:	25.63	25.63
58F:	37.55	36.01	18M:	27.22	26.24

Figure 5.9: Audio-visual vs audio-only test set correctness on noisy audio



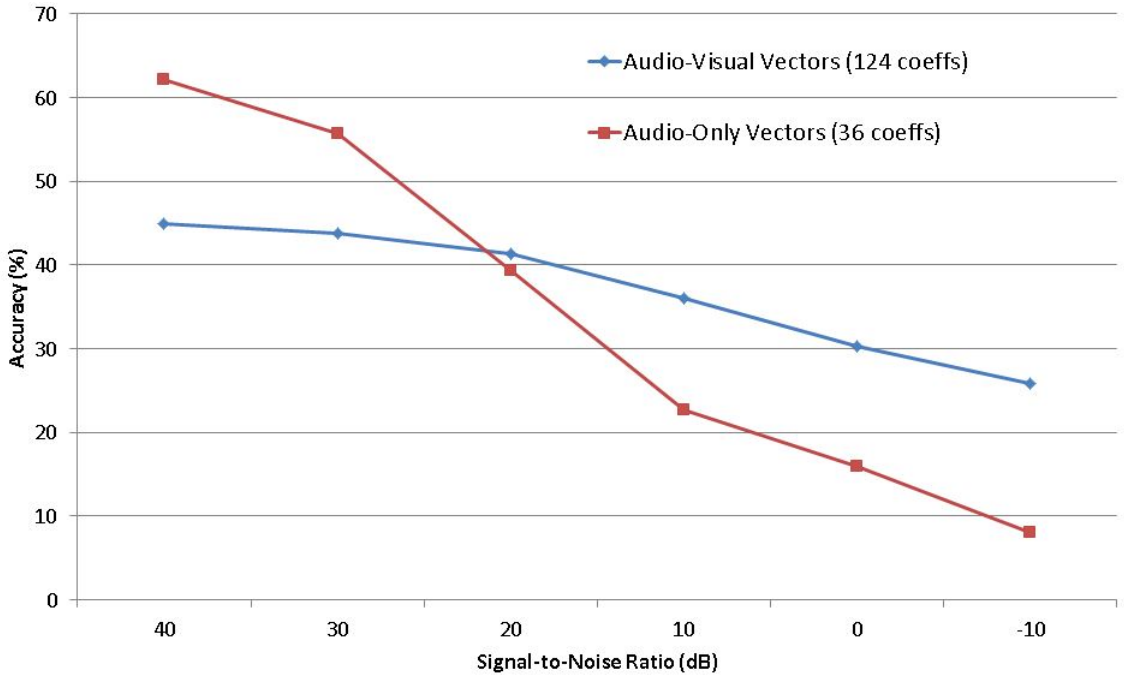
5.3 Audio-Visual Experiments

The final experiments run on the lipspeaker data were to compare audio-visual versus audio-only performance in noise. This was done on volunteer data in Section 4.3.2. The lipspeaker results are also compared to results from that section. The experiment settings from Section 4.3.2 were copied. The training set consisted of 251 sentences from each lipspeaker, leaving 126 sentences from each as the test set. The results are graphed in Figures 5.9 and 5.10.

As Figures 5.9 and 5.10 show, the audio-visual recognizer begins to outperform its audio-only counterpart at an SNR of roughly 22dB. The performance gap then widens as the SNR is lowered further. At the lowest SNR of -10dB, the accuracy of the audio-visual recognizer is 25.86% versus the audio-only accuracy of 8.01%. At the other end of the graph, the audio-visual recognizer's accuracy with clean audio is 44.94% versus the audio-only accuracy of 62.16%.

The trends in these results are the same as those found on the volunteer data in Figures 4.10 and 4.11. Compared to the volunteer results, the lipspeakers' audio-visual accuracy surpasses the audio-only accuracy at a slightly higher SNR (22dB vs 20dB). At every SNR, the audio-visual correctness and accuracy are roughly 8% higher for the lipspeakers than for volunteers. This is understandable, considering the results of Section 5.2.3. The audio-only scores, on the other hand, are roughly 8dB higher for lipspeakers than for volunteers at 40, 30 and 20dB. At 10dB, the audio-only accuracy of the lipspeakers drops sharply, leaving it at a similar level (22.69%) to the corresponding volunteer score (21.58%). This occurs at 0dB for the correctness score. As

Figure 5.10: Audio-visual vs audio-only test set accuracy on noisy audio



a result, at the final SNR of -10dB, the audio-only scores for lipspeakers and volunteers are very similar at about 8% in both cases. This indicates that, in terms of audio, the benefit of fewer speakers and more training data from each speaker has been wiped out by noise at -10dB. This further emphasizes the importance of the corresponding audio-visual results for the lipspeakers.

5.3.1 Audio-Visual Confusion Matrices

In Section 4.3.3, the confusion matrices from the volunteer experiments in noise were examined. The corresponding lispeaker confusion matrices are examined here. Of interest is whether the same trends are present in both sets. The audio-only and audio-visual confusion matrices for the lipspeakers at 40dB are given in Figures 5.11 and 5.12.

As expected, the same trends noted in the volunteer confusion matrices are visible in the lipspeaker confusion matrices. Similar-sounding phonemes with different visemes (e.g. /m/ and /n/) are confused less when visual information is introduced, while the opposite effect is observed on different-sounding phonemes with the same viseme (e.g. /g/ and /k/). Also visible is the increased confusion between vowels and consonants which typically follow them (e.g. /ah/ and /t/, /n/ and /l/), most likely due to coarticulation effects. Comparing the volunteer and lipspeaker audio-visual confusion matrices, most phonemes have higher correctness scores in the lipspeaker matrix. The only phonemes to have higher scores in the volunteer matrix are the least-frequent phonemes /jh/, /ch/, /ng/, /uh/, /oy/ and /th/. These phonemes were the

Figure 5.11: Audio-only confusion matrix from 40dB test in Figure 5.10

```
====================== HTK Results Analysis =======================
  Date: Tue Oct 22 13:24:14 2013
  Ref : TCDTIMITphonemesOffset.mlf
  Rec : Results\splitlResults\TCDTestRecognition\TCDTestRec61.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=378, N=378]
WORD: %Corr=73.87, Acc=62.16 [H=10628, D=721, S=3038, I=1685, N=14387]
------------------------ Confusion Matrix -------------------------
       a   m   b   r   i   v   e   t   a   d   n   a   z   i   l   a   s   e   j   a   p   s   u   k   e   n   y   c   w   h   u   f   o   g   d   t   o   s
       h                   y       y       e               y       h       a   h   r   h   w               w       h   g       h       h   h       w       h   h   y   i
                                                                                                                                                                   l  Del [ %c / %e]
  ah  767   6   0  10   9   7   1  10  52  12   9   6   2  96   8  52   1  30   0   2   4   0  13   8  29   3   1   0   2   7  15   1  14   1   1   0   0   0   99 [65.1/2.9]
   m    0 336   1   1   1   1   0   1   0   1  26   0   0   0   0   0   0   2   0   0   1   0   0   0   0   2   0   0   5   1   0   0   0   0   0   0   0    7 [88.7/0.3]
   b    5   1 207   0   1   7   0   4   0   2   1   0   1   1   0   0   0   0   0   0  23   0   0   3   0   0   0   0   2   0   0   3   0   4  15   0   0   0   13 [73.9/0.5]
   r    7   5   6 432   1   1   3   8   3   2   0   2   0   5   6   1   0  27   0   0   0   5   0   4   1   0   0   1   0  14   2   1   4   5   0   3   0   0   0   91 [78.7/0.8]
  iy    2   0   1   0 468   0   5   0   0   1   0   0   0  32   0   0   0   3   0   0   0   0   6   2   0   0   2   0   0   0   0   0   1   0   0   0   0   0    9 [89.5/0.4]
   v    2   1  10   1   0 177   0   0   0   5   1   0   1   0   0   0   0   1   0   0   2   1   1   0   0   0   0   0   0   0   0  12   0   1   4   0   0   0   11 [80.5/0.3]
  ey    3   0   0   0   5   0 187   1   1   0   0   0   0   2   0   0   0   0   0   0   1   0   0   0   5   0   0   0   0   0   0   0   0   0   0   0   0   0    5 [91.2/0.1]
   t    3   0   1   1   1   2   1 641   0  59   0   0   5   1   1   0   6   2   1   0  13   3   1  22   0   2   1   3   0   3   0   7   1   1  12   0   0   0   58 [80.7/1.1]
  ae   10   0   0   0   0   1   0   0 298   0   0   5   0   0   0  11   0   0   0   0   8   0   0   0   1   2   0   0   0   0   1   0   0   0   0   0   0   0    4 [88.4/0.3]
   d    8   5   5   2   2   6   0  73   0 377   1   1   1   3   1   0   2   0   3   0   4   2   2   3   1   1   1   2   0   0   0   2   0   4  21   1   0   0   65 [70.6/1.1]
   n    3  26   3   1   1   0   1   1   2   4 695   0   0   0   5   1   0   0   0   0   0   0   1   1   0  15   0   0   1   1   0   0   0   0   1   0   0   0   27 [91.1/0.5]
  ay    1   0   0   0   0   0   0   0   0   2   0   0 203   0   0   0   1   0   1   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0    1 [96.7/0.0]
   z    2   0   0   0   0   0   1   0   8   0   1   0   0 340   1   0   0   1   0   0   0   0   4  51   0   2   0   0   0   0   0   2   0   2   0   0   0   0    8 [81.9/0.5]
  ih   51   0   0   4  45   0   0  14   2   0   5   2   0   3 486   0   0   0   4   0   0   2   0   2   0  12   0   2   0   0   4   1   1   0   0   0   0   1   0   28 [75.8/1.1]
   l    7   5   2   2   7   4   0   4   0   9   4   1   1   3 463   3   1   2   0   0   3   0   1   2   2   0   1   0   0   3   2   2   5   1   0   0   0   0   28 [85.7/0.5]
  aa   18   1   1   2   0   3   0   2  21   0   0  15   0   0   0 354   0   9   0   1   1   0   0   2   2   1   0   0   4   0   3   0   5   0   1   0   1   0   13 [79.2/0.6]
  sh    0   0   0   2   0   0   1   0   0   1   0   0   0   0   0   0 133   0   0   0   0   1   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0    0 [95.0/0.0]
  er   30   0   1  36   2   0   0   1   1   1   1   0   0   4   1   6   0 272   0   0   2   0   1   1   5   0   0   0   1   0   0   0   7   1   0   0   0   0    9 [72.7/0.7]
  jh    3   0   0   1   0   0   0   1   0  10   0   0   0   0   1   0  10   0  63   0   0   0   1   0   0   1   5   9   0   0   2   1   0   0   0   0   0   0    4 [58.3/0.3]
  aw    0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   3   0   0   0   0   0  52   3   0   0   0   2   0   0   0   0   0   0   1   0   0   0   0    0 [82.5/0.1]
   p    0   0  14   1   0   3   0   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0 286   0   0  13   0   0   0   0   0   4   0   2   0   1   4   0   0   0    6 [86.1/0.3]
   s    2   1   1   1   0   0   0   8   0   1   0   0  61   0   0   0   0   0   0   0   0   0   1 546   0   0   0   0   0   0   0   0   0   4   0   0   0   0   0   19 [87.2/0.6]
  uw    6   0   1   1   0   0   0   0   0   1   1   0   0   2   0   1   0   0   0   0   0   0   0   0 190   0   0   0   0   0   2   0   0   0   0   0   0   0    9 [92.7/0.1]
   k    2   0   1   0   0   0   0  12   0   3   1   0   0   1   0   0   2   0   0   0   0  10   0   0   0 421   0   0   0   0   0   4   0   3   0  11   2   0   0   0    9 [89.0/0.4]
  eh    7   0   0   1   0   0   6   0  17   1   0   0   0   6   0   0   0   3   0   4   0   0   0   0   0   0 269   0   0   0   0   0   0   1   0   0   0   0   0   0   11 [85.4/0.3]
  ng    2   2   1   0   1   0   0   1   0   1  13   0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0 125   0   0   0   0   0   0   1   0   0   0   0   0    3 [83.9/0.2]
   y    0   0   0   0   0   5   1   0   2   0   0   0   0   0   2   0   0   0   0   3   0   0   0   0   0   0   0   0 103   1   1   0   0   0   0   0   0   0   0   0    8 [87.3/0.1]
  ch    2   0   0   0   0   0   0   0  11   0   1   0   0   0   0   0   0   0  26   0   7   0   0   0   0   1   2   0   0   1  53   0   1   0   1   0   1   0   0   0    1 [49.5/0.4]
   w    2   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0 223   1   0   1   0   1   0   0   0   0    3 [96.1/0.1]
  hh    2   0   0   1   0   0   0   1   1   0   0   0   0   0   0   1   0   1   0   0   0   0   0   3   0   0   1   0   0   1   0   0 120   0   4   0   1   2   0   0   0    9 [86.3/0.1]
  uh    8   0   0   0   0   1   1   0   0   0   0   0   0   0   2   2   4   1   0   0   0   0   0   0   4   0   0   0   0   0   0   1   0  21   0   2   0   0   0   0   10 [44.7/0.2]
   f    0   0   2   0   0   1   0   3   1   2   0   0   0   0   0   0   0   0   0   0   0   0   0  11   3   0   3   0   0   0   0   0   0   1   0 203   0   0   4   1   0   0    7 [86.4/0.2]
  ow    7   0   0   2   0   0   0   0   0   1   0   0   0   0   1   8   0   3   0   0   0   0   0   1   0   0   0   0   0   0   0   1   0   2   0 133   0   0   0   1   0    2 [83.1/0.2]
   g    4   1   4   1   2   0   0   2   0   4   2   0   1   2   0   0   0   0   0   0   1   3   0   0  32   0   0   1   0   0   1   0   3   0 107   4   0   0   0    5 [61.1/0.5]
  dh    3   2   9   2   0   7   0   4   0  15   1   0   1   1   0   0   0   1   0   0  13   1   0   2   1   0   0   0   0   0   0   0   0   6   0   1 108   1   0   0   13 [60.3/0.5]
  th    1   0   0   0   0   0   1   0   5   0   1   0   0   2   0   0   0   0   0   0   0   4   1   0   3   0   0   0   0   0   0   2   0   8   0   0   6   0   0   0    2 [ 0.0/0.2]
  oy    0   0   0   0   0   2   0   1   2   0   0   0   4   0   4   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   2   0   1   0   0   0   0   0  14   0    0 [45.2/0.1]
 sil   17   5  28   0   3  17   1  64   4  12   7   0   6   5   1   6   1   3   1   0 175   9   0  60   1   1   0   3   6  27   0  95   0   2  93   0   0 755 124 [53.6/4.5]
Ins   121  37  72  45  51  48  15 153  30  88  38  10  16  74  36  24  19  23   3   1 129  38  15 106  41  11  25  15  20 114   9 144  20  13  80   0   0   1
===================================================================
```

lowest-scoring in the volunteer confusion matrix, and the cause for this was thought to be their relative lack of training data. This theory is further supported by their even worse performance in the lipspeaker confusion matrices, since the amount of lipspeaker training data was lower still (753 sentences vs 3752).

Looking at the changes in the confusion matrices as the SNR is lowered, again the same trends are seen in the lipspeaker and volunteer audio-visual matrices. The audio-only and audio-visual lipspeaker confusion matrices at -10dB are given in Figures 5.13 and 5.14. The most robust phonemes in both cases are the ones in the smallest viseme groups. As such, the highlighted examples in Section 4.3.3 (/aw/, /f/, /v/, /b/, /p/) are also among the highest-performing phonemes in Figure 5.14. Deletions are once again the main factor in the audio-only results (Figure 5.13), accounting for 90% of all phonemes in the test set. Only 21% of phonemes were deleted by the audio-visual recognizer. This is even better than the volunteers' figure of 38%, and is a large factor in the improved overall score of 25.86% versus 18.32% accuracy. The high scores of some phonemes at -10dB are inflated by large levels of insertions, particularly /hh/

Figure 5.12: Audio-visual confusion matrix from 40dB test in Figure 5.10

```
====================== HTK Results Analysis =======================
  Date: Tue Oct 22 12:21:22 2013
  Ref : TCDTIMITphonemesOffset.mlf
  Rec : Results\splitlResults\TCDTestRecognition\TCDTestRec61.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=378, N=378]
WORD: %Corr=59.00, Acc=44.94 [H=8489, D=1785, S=4113, I=2024, N=14387]
------------------------ Confusion Matrix -------------------------
       a   m   b   r   i   v   e   t   a   d   n   a   z   i   l   a   s   e   j   a   p   s   u   k   e   n   y   c   w   h   u   f   o   g   d   t   o   s
       h                   y       y       e               y       h       a   h   r   h   w               w       h   g       h       h   h       w       h   h   y   i
                                                                                                                                                                   l   Del [ %c / %e]
  ah 680   4   8  10  12   1   1  15  30  32  30   1   7  72  17  23   6  19   2   2   2   2  10  16  15   1   6   1   2   8   9   1  10   3   1   0   1   0 218 [64.2/2.6]
   m   6 306  17   0   0   1   0   2   2   0  13   0   1   1   2   2   0   0   1   1   8   0   1   1   0   2   0   0   2   0   0   0   1   0   0   0   0   0  16 [82.7/0.4]
   b  11  20 156   1   0   0   0   3   0   2   0   0   0   0   3   3   0   0   0   0  51   1   0   2   1   0   0   0   0   0   0   0   0   0   1   0   0   0  38 [61.2/0.7]
   r  26   4   4 343   0   7   2  12   7   2   9   4   1   2   7   6   0  27   1   1   3   2   1   8   4   0   2   0   5   2   2   7   4   1   1   0   0   0 133 [67.7/1.1]
  iy  13   3   2   2 375   0   8   5   0   4   4   0   1  43   6   0   2   3   1   0   0   1   0   2   0   1   2   0   0   6   0   0   0   2   1   0   0   0  45 [77.0/0.8]
   v   0   1   1   4   0 153   0   0   1   3   3   0   0   2   1   1   0   0   0   0   0   0   0   0   3   0   0   0   0   0   1   0  33   1   0   1   0   0   0  22 [73.2/0.4]
  ey   5   0   1   0   7   0 164   0   0   2   3   0   1   6   2   1   1   0   0   0   0   0   0   0   2   6   0   0   0   0   0   1   0   0   0   1   0   0   0   7 [80.8/0.3]
   t  23   2   0   3   4   3   0 475   3  83   1   1  16  10   6   3   6   5   1   0   0  18   4  22   4   0  13   2   1   8   0   0   1   0  12   0   0   0 122 [65.1/1.8]
  ae  11   1   0   0   0   2   0   1 263   3   0   2   1   1   2   9   1   3   0   3   0   1   1   0  16   0   0   0   0   2   0   1   0   0   1   0   0   0  16 [80.9/0.4]
   d  28   2   1   4   1   3   1  78   2 238  16   0   7  15   7   2   3   2   1   0   1   9   2   9   1   1   5   0   1   4   1   2   0   2  19   0   0   0 131 [50.9/1.6]
   n  17   9   3   2   5   2   0  14   3  13 490   2   2  15  25   1   3   0   0   1   1   4   0   4   4   7   5   0   2   8   3   0   2   3   3   0   0   0 137 [75.0/1.1]
  ay   5   1   0   1   1   0   0   0  12   0   1 174   0   0   0   0   0   0   0   2   0   1   0   0   4   0   0   0   0   2   0   0   0   0   0   0   0   0   7 [85.3/0.2]
   z   6   1   0   2   0   1   0  28   0  14   3   0 230   8   2   3   0   0   1   0   0  63   1   7   0   0   1   1   2   6   0   0   0   0   0   0   0   0  43 [60.5/1.0]
  jh  61   3   2   3  30   0   5  11   1  10   7   0   4 390   4   1   1   2   1   0   2   2   0   3   7   0   4   0   4   7   0   0   0   1   7   0   0   0  96 [68.1/1.3]
   l  23   2   3   8   3   1   0   6   7  11  17   1   0  23 305   8   1   3   0   0   2   5   2  12   2   1   6   0   1   7   0   0   0   1   2   0   0   0 105 [65.9/1.1]
  aa  18   1   2   2   0   0   0   2   6   2   1   2   0   0   3 381   1   0   0   1   1   1   0   3   0   0   0   0   0   0   3   0   0   5   0   0   0   0   0  25 [87.6/0.4]
  sh   2   0   0   0   0   0   0   0   7   0   0   0   0   0   0   1   0   0 116   0   2   0   0   1   0   0   0   0   2   3   0   0   1   1   0   0   0   0   0   4 [85.3/0.1]
  er  47   1   0  45   1   1   0   3   1   4  11   0   1   9   7   4   0 185   0   1   1   0   2   0  11   0   0   0   2   1   2   0   7   0   1   0   0   0  35 [53.2/1.1]
  jh   4   0   0   3   0   0   0   8   1   6   2   0   3   1   0   1  18   3  31   0   0   0   2   0   1   1   6   1   0   1   0   0   0   0   1   0   0   0  18 [33.0/0.4]
  aw   0   0   0   1   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0  57   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   1 [91.9/0.0]
   p   5   8  42   2   1   1   0   5   1   6   1   0   0   2   0   0   1   0   0   0 226   1   0   2   0   0   0   0   2   8   0   1   0   0   1   0   0   0  22 [71.5/0.6]
   s   5   0   1   2   0   2   0  25   1  13   0   0  62   5   0   1   1   3   0   0   1 469   1   1   0   0   2   0   1   1   0   0   0   0   0   0   0   0  48 [78.6/0.9]
  uw  11   1   2   0   2   0   0   1   0   2   1   1   0   2   2   1   1   2   0   0   0   0 158   2   0   0   1   0   8   0   0   0   7   0   0   0   0   0   9 [77.1/0.3]
   k  12   1   0   3   4   0   3  27   1  14   3   1   0  12   5   3   0   1   0   0   1   2   0 260   4   0   8   0   4   9   0   1   3  13   2   0   0   0  85 [65.5/1.0]
  eh  12   3   0   0   1   0   6   3  20   0   3   0   1  13   4   0   2   2   0   2   1   2   0   1 218   0   0   0   0   3   0   1   0   0   1   0   0   0  27 [72.9/0.6]
  ng   4   1   2   1  24   0   0   2   0   2  31   1   0   3   6   0   0   0   0   0   0   1   0   0   4   0  24   1   1   0   2   0   0   0   0   2   0   0   0  40 [21.4/0.6]
   y   7   0   0   0   0   4   0   0   5   0   5   1   0   0   7   4   0   2   1   2   0   0   0   0   2   1   1   0  62   1   0   1   1   1   0   0   0   0   0  18 [57.4/0.3]
  ch   4   1   2   0   1   0   0  13   0   4   1   0   0   4   1   0  33   1  11   0   0   0   0   1   2   0   0   2  15   0   3   0   0   0   0   0   0   0   0   9 [15.2/0.6]
   w   3   1   0   1   0   0   0   0   0   0   0   4   1   1   1   0   1   0   0   0   0   0   0   1   1   0   0   0   0 215   0   0   0   2   0   0   0   0   0   3 [92.7/0.1]
  hh   4   0   0   2   0   1   0   9   1   5   2   0   0   4   2   2   0   0   0   0   0   1   0   0   3   0   1   1   0   1  79   1   0   0   0   0   0   0   0  29 [66.4/0.3]
  uh   7   0   0   0   0   0   0   0   3   0   2   3   0   0   0   1   2   0   4   1   0   0   0   7   0   0   0   2   0   1   0  11   0   1   0   0   0   0   0  12 [24.4/0.2]
   f   1   0   0   0   0   0  18   0   2   0   0   0   0   0   1   0   0   0   0   0   0   0   0   1   0   1   1   0   0   0   0   0   1 208   0   0   2   0   0   0   6 [88.1/0.2]
  ow   5   0   0   0   0   0   2   0   1   0   0   2   0   1   0   0   8   1   0   0   0   0   0   0   2   0   1   0   0   0   0   0   0   0 132   0   0   0   0   7 [85.2/0.2]
   g   8   0   0   1   6   0   0   8   0  14   9   1   0  11   5   0   1   0   0   0   0   2   4   1  31   1   1   2   1   0   6   1   0   0  28   5   0   0   0  33 [19.0/0.8]
  dh   1   1   1   2   3   6   0   7   0   7   5   1   0   2   1   0   0   2   0   0   0   0   4   0   2   0   0   1   0   0   0   0   1   0   0 108   0   0   0  37 [69.7/0.3]
  th   1   0   0   0   0   0   2   0   4   0   2   1   0   1   2   1   0   0   1   0   0   0   3   0   4   0   0   0   0   0   0   0   0   2   0   0   6   2   0   0   4 [ 6.3/0.2]
  oy   2   0   2   0   2   0   1   1   0   0   0   4   0   1   1   1   0   1   0   0   1   0   1   0   0   0   0   0   0   2   0   0   0   1   0   0   0   7   0   3 [25.0/0.1]
 sil  51  13  11   5  15   3   1  55   9  42  13   1  16  16  11  15  11   7   0   0   5  37   3  75  12   1   8   1   2 115   1   9   5  10  24   0   0 755 174 [55.6/4.2]
Ins 185  22  33  48  73  19  26 193  47 156  96   9  87 115  60  54  32  34   5   1  28  67  13 167  31   9  25   6  24 252   6  18  18  18  46   0   0   1
===================================================================
```

and /dh/. This is understandable, since the audio profile of these phonemes is the closest to noise. However, some phonemes, for example /w/, /ae/ and /aa/, have relatively high levels of correctness without a large number of insertions, an audio profile close to noise or a small viseme group. It is possible that the lipspeakers have made these phonemes more visually distinctive than the volunteers.

5.3.2 Individual Lipspeaker Performances

The differences between the individual lipspeaker scores in the visual-only experiments were noted in Section 5.2.3. To check whether these differences persisted into the audio-visual results, each lipspeaker's performance was compared to the overall average at each SNR. The results are given in Table 5.5. Since the audio component of each audio-visual score must be taken into account, audio-only performance comparisons are also given. The audio-only and audio-visual performances of each lipspeaker at each SNR are graphed in the scatter plots of Figures 5.15 and 5.16 respectively.

Figure 5.13: Audio-only confusion matrix from -10dB test in Figure 5.10

```
====================== HTK Results Analysis =======================
  Date: Tue Oct 22 13:38:33 2013
  Ref : TCDTIMITphonemesOffset.mlf
  Rec : Results\splitlResults\TCDTestRecognition\TCDTestRec61.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=378, N=378]
WORD: %Corr=8.02, Acc=8.01 [H=1154, D=13007, S=226, I=2, N=14387]
------------------------ Confusion Matrix -------------------------
       a   m   b   r   v   e   t   a   d   n   a   i   l   a   e   p   k   e   w   h   f   o   g   d   o   s
       h                       y       e           y   h       a   r           h       h       w       h   y   i
                                                                                                           l  Del [ %c / %e]
  ah  77   0   1   0   0   0   0   1   0   0   0   0   2   0   0   3   1   0   0   0   1   0   0   1   0   0 1191 [88.5/0.1]
   m   0  37   1   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   1   0   0   0   0   0   0  345 [90.2/0.0]
   b   0   0   8   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  284 [88.9/0.0]
   r   0   0   0   1   0   0   0   0   0   1   0   0   1   0   0   5   0   1   0   0   0   0   0   1   0   0  630 [10.0/0.1]
  iy   0   1   0   0   4   0   0   0   0   0   0   0   0   1   1   7   1   0   0   0   0   0   0   0   0   0  517 [ 0.0/0.1]
   v   0   0   0   0  19   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0  211 [95.0/0.0]
  ey   0   0   0   0   1   1   0   0   0   1   0   0   1   0   0   1   1   1   0   0   0   1   0   0   0   0  202 [12.5/0.0]
   t   0   2   2   0   3   0   0   0   0   0   0   0   2   1   0   2   1   0   0   2   1   0   0   0   0   0  836 [ 0.0/0.1]
  ae   2   1   0   0   0   0   0   5   0   0   0   0   1   1   0   1   0   0   0   1   0   0   0   0   1   0  328 [38.5/0.1]
   d   0   5   2   0   1   0   0   0   3   0   0   0   3   0   0   7   1   0   0   1   1   0   0   0   0   0  575 [12.5/0.1]
   n   0   1   1   0   2   0   0   0   0  44   0   0   2   0   0   4   0   0   0   0   0   0   0   0   0   0  736 [81.5/0.1]
  ay   0   0   0   0   0   0   0   0   0   0   3   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0  206 [60.0/0.0]
   z   0   4   1   0   4   0   1   0   0   1   0   0   5   2   0   9   1   0   1   0   2   0   0   1   0   0  391 [ 0.0/0.2]
  ih   2   2   0   0   0   0   0   0   0   0   0   0   4   0   0   0   1   0   0   0   0   0   0   0   0   0  660 [44.4/0.0]
   l   1   1   0   0   0   0   0   0   0   0   0   0   0  70   0   0   2   0   0   0   0   1   0   0   1   0   0  492 [92.1/0.0]
  aa   1   1   0   0   0   0   0   0   0   0   0   0   0   0  23   0   0   0   0   0   0   0   0   0   0   0  435 [92.0/0.0]
  sh   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  139 [ 0.0/0.0]
  er   0   0   1   1   0   0   0   0   0   0   0   0   2   0   1   8   0   0   0   1   1   0   0   0   0   0  368 [ 6.7/0.1]
  jh   0   0   0   0   1   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0  109 [ 0.0/0.0]
  aw   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   62 [ 0.0/0.0]
   p   0   1   0   0   0   0   0   0   0   0   0   0   1   0   0  66   0   0   0   0   0   0   0   0   0   0  270 [97.1/0.0]
   s   0   3   2   0   0   0   0   0   0   1   0   0   2   1   0   7   1   0   0   0   1   0   0   0   0   0  627 [ 0.0/0.1]
  uw   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   0   0   0   0   0   0   0   0   0  210 [ 0.0/0.0]
   k   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   2  12   0   1   0   0   0   0   0   0   0  465 [70.6/0.0]
  eh   0   1   0   0   1   0   0   0   0   0   0   0   1   0   0   1   0   4   1   0   0   0   0   0   0   0  317 [44.4/0.0]
  ng   0   1   1   0   1   0   0   0   0   1   0   0   2   0   0   5   1   0   0   0   0   0   0   0   0   0  140 [ 0.0/0.1]
   y   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  126
  ch   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   0  105 [ 0.0/0.0]
   w   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   5   0   0   0   0   0   0   0  229 [83.3/0.0]
  hh   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   0   0   0   0   0  144
  uh   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   57
   f   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   7   0   0   0   0   0  232 [70.0/0.0]
  ow   0   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0  160 [ 0.0/0.0]
   g   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0  178 [50.0/0.0]
  dh   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   3   0   0  188 [75.0/0.0]
  th   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   36
  oy   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   31
 sil   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0 756 775 [99.9/0.0]
Ins    0   0   0   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
====================================================================
```

From Table 5.5, it can be seen that the differences found in Table 5.2 do persist into the audio-visual results. At 40, 30 and 20dB, the lipspeakers in order of decreasing audio-visual accuracy are 2, 3, 1. The same order was observed in Table 5.2. However, at 10dB and 0dB, the order becomes 2, 1, 3. This is the same order observed in the audio-only accuracies at these SNRs. Looking at the audio-only scores, the most likely reason for Lipspeaker 3's lower audio-visual accuracy is that her audio-only score at these SNRs is the lowest. This indicates that while visual-only performance is a good indicator of audio-visual performance in noise, the robustness of the speaker's audio to noise is also a factor.

Lipspeaker 2 has the highest audio-visual accuracy at each SNR, but also has the highest audio-only accuracy at each SNR. At -10dB, her audio-visual accuracy is 31.86%, which is higher than the accuracy of Lipspeakers 1 and 3 at 0dB. Her audio-only score at -10dB is not significantly higher than that of the other two lipspeakers, so the audio-visual score can in this case be attributed almost entirely to the visual component of the signal.

Figure 5.14: Audio-visual confusion matrix from -10dB test in Figure 5.10

```
====================== HTK Results Analysis =======================
  Date: Tue Oct 22 12:55:41 2013
  Ref : TCDTIMITphonemes0ffset.mlf
  Rec : Results\splitlResults\TCDTestRecognition\TCDTestRec61.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=378, N=378]
WORD: %Corr=38.31, Acc=25.86 [H=5512, D=2995, S=5880, I=1791, N=14387]
------------------------ Confusion Matrix -------------------------
       a   m   b   r   i   v   e   t   a   d   n   a   z   i   l   a   s   e   j   a   p   s   u   k   e   n   y   c   w   h   u   f   o   g   d   t   s
       h                   y       y               e               y       h       a   h   r   h   w               w       h   g       h       h   h       w       h   h   i
                                                                                                                                                                   l   Del [ %c / %e]
  ah  404  18  18   3   9   9   1  23  23  23  36   0   2  40  24  14   8  12   0   1  22   6  11  45  25   4   9   2   5  77   1   8   8   4  33   0   0 350 [43.5/3.6]
   m    3 181  29   2   3   1   0   3   5   3   6   0   0   7   5   3   0   1   0   0  28   0   1   4   4   3   0   0   3  14   0   2   1   0   7   0   0  67 [56.7/1.0]
   b    5  16 125   0   1   1   0   3   0   2   6   0   1   2   7   3   0   0   0   0  54   5   0   2   3   1   0   0   1   3   0   2   0   0   6   0   0  44 [50.2/0.9]
   r   17   5   7 233   4   9   1  12  15   4  17   0   2   7  11   3   1   5   0   2   4   4  12  14   6   0   7   0   5  32   2  17   3   2  15   0   0 162 [48.7/1.7]
  iy   29   4   6   2  86   1   4  13   6  14  26   0   3  56  14   2   8   0   1   0   8  11   0  28   6   3   4   0   2  47   0   5   0   4  20   1   0 118 [20.8/2.3]
   v    2   1   1   2   0 110   1   3   1   1   3   0   0   5   1   2   0   1   0   0   1   0   2   2   6   0   0   0   0   4   0  44   2   0   4   0   0  32 [55.3/0.6]
  ey    2   0   0   0   5   0  84   2   4   1   4   0   1  11   4   0   1   2   0   0   3   0   0   6  14   2   0   0   0  15   0   2   1   3   5   0   0  38 [48.8/0.6]
   t   11   9   9   3   0   4   0 342  12  38  23   0  11  18   3   2   6   0   0   0   9  15   7  18   6   4  17   2   3  37   0   3   0   1  39   0   0 200 [52.5/2.2]
  ae    8   1   1   0   0   0   0   4 222   1   1   1   7   2   3   4   0   0   0   0   0   2   0   3  15   0   1   0   0  13   0   2   1   0   1   0   0  48 [75.8/0.5]
   d   16   6   3   2   6   1   3  57   8 130  19   0   8  14   8   2   2   1   1   1   9  14   7  13   4   1  11   2   6  40   1   8   0   6  35   0   0 154 [29.2/2.2]
   n    5   8   5   2   7   5   3  26  11  22 239   1   8  17   8   6   6   6   2   0   2   5   6  25  11   4   6   0   5  67   2   2   2   5  31   0   0 230 [42.7/2.2]
  ay    3   2   0   1   4   0   0   4  18   1   2  76   0   8   1   4   0   0   0   1   8   1   0   4   9   0   0   1   1  14   0   1   0   1   5   0   0  41 [44.7/0.7]
   z   12   1   2   2   5   3   0  26   7  21  20   1  85   3   5   2   2   0   2   1   6  17   4  20   5   1   3   0   3  36   0   4   1   1  17   0   0 105 [26.7/1.6]
  th   25   6   8   2  12   7   2  15   3  13  25   0   4 240   8   1   4   2   1   0   7   7   2   7  13   4   4   0   2  51   0  12   1   0  16   0   0 165 [47.6/1.8]
   l   10   6   7   2   3   8   4  15  10   6  32   1   2  14 157   7   3   6   0   1   7   5   3  14   2   1   7   1   5  68   0   1   0   6  21   0   0 133 [36.1/1.9]
  aa   13   3   5   1   0   2   0   7   8   1   4   1   1   3   4 264   0   1   0   0   4   1   5   4   1   1   0   1   7  21   0   4   5   0   8   0   0  80 [69.5/0.8]
  sh    4   0   1   0   0   0   0   9   2   3   4   0   1   0   2   0  74   2   4   0   1   2   0   1   0   0   4   2   0   3   1   0   0   0   3   0   0  17 [60.2/0.3]
  er   29   5   6  22   5   3   1  14   3   8  15   0   4   8   8  10   4  33   2   0   7   6   4  15   6   1   1   1   2  32   2   5   1   3  17   0   0 100 [11.7/1.7]
  jh    1   0   0   2   0   0   0   6   2   2   7   0   1   2   1   1  14   2  20   0   1   2   2   2   1   1   3   3   0   7   0   0   0   0   4   0   0  25 [23.0/0.5]
  aw    1   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0  43   0   0   0   1   3   0   0   0   1   0   0   0   0   2   0   0  10 [81.1/0.1]
   p    3  18  30   0   1   2   0   2   3   5   2   0   0   8   1   0   0   2   0   0 208   0   1   0   2   0   0   0   2   7   1   1   0   0   4   0   0  35 [68.6/0.7]
   s   11   5   6   2   2   7   0  30   2  27  13   1  22   9   5   3   1   1   1   0   8 195   4   9   2   4   6   0   1  65   0  13   0   1  17   0   0 172 [41.2/1.9]
  uw   11   2   1   1   0   2   0   3   0   3   5   0   1   0   0   4   0   0   0   0   0   1   0 119   5   0   0   2   0   9   3   1   5   6   2   3   0   0  25 [63.0/0.5]
   k   15   4   2   3   2   3   3  12   5  15  18   2   4   7   3   1   0   1   0   0   7   5   1 143   6   3  12   0   5  47   0   5   2   9  22   0   0 115 [39.0/1.6]
  eh    6   6   1   4   1   1   1   3  25   2   3   0   2  10   5   1   1   1   0   2   2   0   0   7 157   0   1   0   1  17   0   5   0   0   5   0   0  56 [58.1/0.8]
  ng    2   1   2   2  14   3   1   3   2   4   7   0   0   3   2   0   1   2   0   0   3   2   0   3   0  11   1   1   1  18   0   2   1   0   7   0   0  53 [11.1/0.6]
   y    4   1   0   2   2   2   0   3   0   4   8   0   0   0   0   0   2   1   0   0   1   1   2   4   0   0  46   0   0   6   0   0   0   0   7   0   0  30 [47.9/0.3]
  ch    7   0   2   1   0   0   0   7   0   6   2   0   1   0   2   0  15   0   5   0   1   0   0   1   3   0   7   5   0  11   1   2   0   1   2   0   0  26 [ 6.1/0.5]
   w    1   0   3   1   0   0   0   0   0   1   2   0   0   1   1   1   1   0   0   0   2   0   3   0   0   0   0   0 203   5   0   2   0   0   2   0   0   6 [88.6/0.2]
  hh    1   0   2   1   0   0   0   7   0   1   1   0   1   1   3   0   1   0   0   0   1   1   0   1   0   2   0   0   1  94   0   1   1   0   3   0   0  24 [75.8/0.2]
  uh    4   0   0   0   0   2   0   1   0   2   4   0   0   1   1   3   0   0   0   0   1   0   0   4   0   0   0   1   0   0   3   6   1   1   0   2   0   0  20 [16.2/0.2]
   f    2   0   0   1   0   9   0   0   1   0   0   3   2   0   0   0   0   0   1   0   1   0   5   0   0   0   0   0   1   0 198   0   0   2   0   0  15 [87.2/0.2]
  ow    5   4   2   0   0   1   1   3   0   2   2   0   4   2   0   6   1   0   0   0   0   0   0   8   3   0   0   1   0   5   4   0   1  82   0   1   0   0  24 [59.4/0.4]
   g    4   1   1   0   7   1   1  10   3   1   4   2   1   7   2   1   1   1   0   0   0   1   1   3  16   5   1   4   0   0  23   0   3   1  12   9   0   0  53 [ 9.4/0.8]
  dh    0   0   0   0   1   1   0   3   1   2   4   0   1   2   2   1   0   0   0   0   1   0   0   2   2   0   3   0   0   5   0   4   0   0 129   1   0  27 [78.2/0.3]
  th    2   0   0   0   0   1   0   2   0   1   1   0   0   2   2   0   0   0   0   0   1   1   0   0   1   0   0   0   0   6   0   1   0   0   7   1   0   7 [ 3.4/0.2]
  oy    6   0   1   1   0   0   0   0   0   0   1   1   0   1   0   2   0   0   0   0   0   0   0   4   0   1   0   0   2   3   0   0   0   0   0   0   0   8 [ 0.0/0.2]
 sil   29  15  11   7   6   2   1  31  16  27  34   2  11  14  17  12  10   1   0   1  16  21   7  41  18   4   7   0   5 155   2  15   4   9  46   0 755 180 [55.8/4.1]
Ins    73  39  42  17  16  31   9  97  62  60 139   7  33  65  61  34  19  12  17   1  50  44  32 120  47  17  32   7  25 380   4  50  17  13 118   0   1
===================================================================
```

Table 5.5: Differences between audio-visual/audio-only accuracy averages of each lipspeaker and overall average at each SNR. Positive differences are above-average.

SNR	40dB		30dB		20dB		10dB		0dB		-10dB	
	AV	Audio	AV	Audio	AV	Audio	AV	Audio	AV	Audio	AV	Audio
Overall average %	45.23	62.38	44.11	55.88	41.62	39.51	36.41	22.82	30.67	16.03	26.18	8.07
Diff. from overall % (Lipspkr1)	-5.71	-9.88	-4.93	-4.69	-3.10	3.94	-1.94	0.88	-2.76	0.37	-3.02	-0.97
Diff. from overall % (Lipspkr2)	5.76	5.31	6.29	4.87	6.08	2.79	6.34	1.98	5.77	1.10	5.68	1.22
Diff. from overall % (Lipspkr3)	-0.06	4.57	-1.37	-0.18	-2.97	-6.73	-4.41	-2.85	-3.01	-1.47	-2.67	-0.25

Figure 5.15: Audio-visual performance of each lipspeaker at each SNR

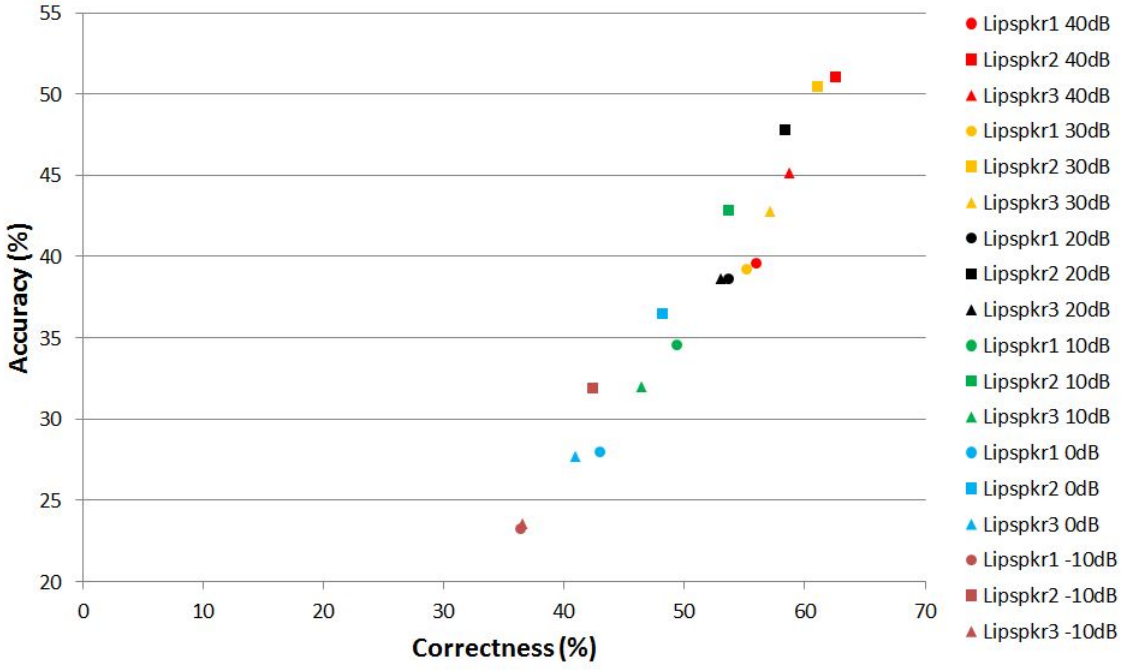
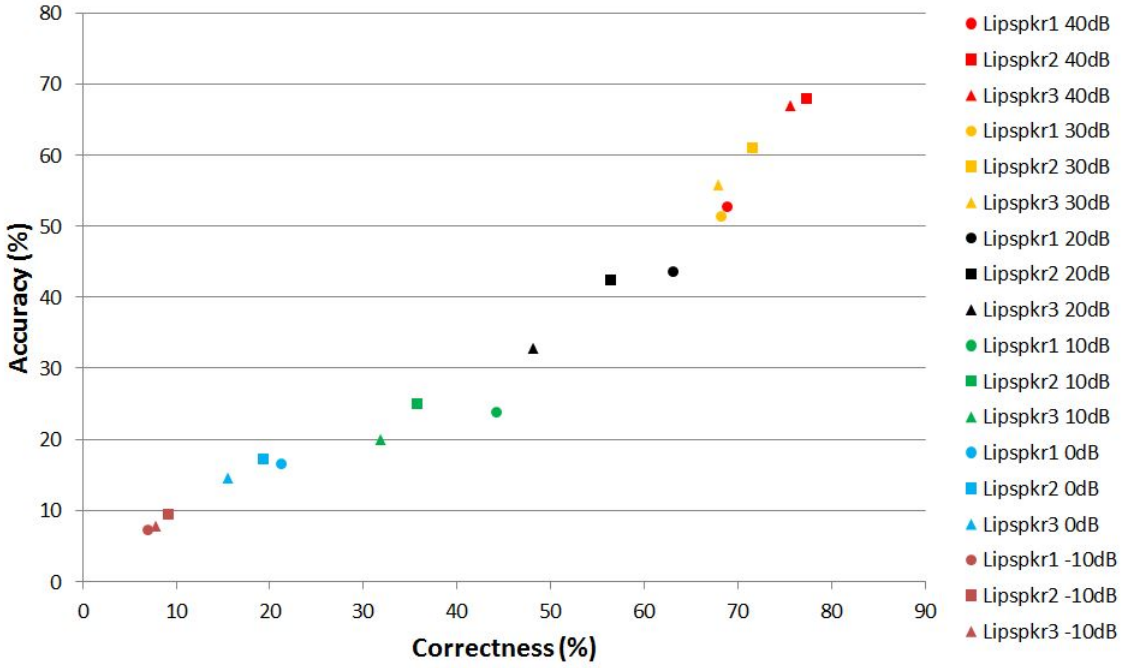


Figure 5.16: Audio-only performance of each lipspeaker at each SNR



5.4 Summary

This chapter has introduced the secondary "lipspeaker" part of TCD-TIMIT, which consists of 1131 TIMIT sentences recorded by 3 lipspeakers. Experiments run on the lipspeaker data are discussed and compared to results obtained on the main "volunteer" part of TCD-TIMIT from Chapter 4. A brief overview of the lipspeakers and the motivation for their inclusion is given in Section 5.1.

Section 5.2 details the visual-only experiments run on the lipspeaker data. The lipspeaker data was first tested on a volunteer-trained recognizer. These results were then compared to results from a recognizer trained and tested with equal amounts of volunteer and lipspeaker data. Next, a recognizer was trained and tested on lipspeaker data only, and these results were compared to volunteer results on a volunteer-only recognizer. Volunteer data was then tested on the lipspeaker-trained recognizer.

The audio-visual experiments undertaken with the lipspeaker data are discussed in Section 5.3. Using the experiment setup from Section 4.3.2, audio-visual and audio-only lipspeaker performance was evaluated using increasingly-noisy audio. The results are compared to the corresponding volunteer results from Section 4.3.2, and the individual lipspeaker performances are also discussed.

6 Conclusions

This thesis has introduced TCD-TIMIT, a new corpus for audio-visual speech recognition. The creation of this corpus was motivated by the lack of corpora suitable for visual and audio-visual continuous speech recognition. This is a commonly-cited issue among researchers in the field. The goals for TCD-TIMIT were for it to have high-quality audio and video footage, a set of time-aligned transcription files, and a set of baselines which can be used as a reference by other researchers. As such, there are two main topics discussed in this thesis: the creation of TCD-TIMIT, and the baseline experiments run on the data afterwards.

6.1 Database Creation

TCD-TIMIT is intended for continuous speech recognition research, hence the speakers were recorded reading full sentences. The sentences are taken from TIMIT, a database developed by Texas Instruments and MIT in the 1980s for continuous ASR. Compared to TIMIT, which consists of 630 volunteers reading 10 sentences each, fewer volunteers (59 volunteers and 3 lipspeakers) were recruited for TCD-TIMIT, but each volunteer read 98 sentences (lipspeakers read 377).

The speakers recorded for TCD-TIMIT are split into two groups in the database. The main group consists of 59 regular speakers (referred to as "the volunteers"). A second, smaller group consists of 3 professionally trained lipspeakers (referred to as "the lipspeakers"). The lipspeaker part of the database was recorded to test the hypothesis that automatic lipreading systems may find lipspeakers easier to lipread than regular speakers.

Video footage was recorded from two angles (straight and 30°). The decision to record two views was based on research which argues that there is useful visual information in angled views of a speaker's face. The footage was processed to create audio and video clips for each sentence (final count: 6913 clips in two views). An automated clipping process was created which managed to clip roughly 90% of the sentences correctly. The rest were clipped manually.

With the sentences clipped, a set of time-aligned phoneme-level label files was then created. TIMIT's phoneme-level label files were initially converted into TCD-TIMIT label files using the forced alignment process (Section 3.3.2). However, experiments run using these force-aligned files (Section 3.3.4) cast doubt on their suitability. As a result, a new set of force-aligned files was created using a tool called P2FA [90]. Despite results from experiments run using these force-aligned files not outperforming the previous results by a significant margin, the label files more accurately represent the pronunciations in TCD-TIMIT.

On the visual side, viseme-level label files were created from the phoneme-level files using a phoneme-to-viseme map (given in Table 2.4). The limitations of this approach are discussed in Section 2.2.1, but it is the simplest approach and is commonly used in the literature. With the viseme-level label files obtained, corresponding visual feature files were extracted to begin baseline experiments. The extraction method followed was originally developed by Cappelletta [7]. Based on results obtained by Cappelletta and others in the literature, the DCT was chosen as the parametrization method for all subsequent visual experiments. With the necessary label and feature files obtained, baseline audio, visual and audio-visual experiments were then run on TCD-TIMIT.

6.2 Baseline Results

The baseline experiments run on TCD-TIMIT all made use of HMMs to model units of speech. A short overview of the theory behind HMMs is given in Section 2.1.3. HTK was the toolkit used to create and test the HMMs. A full explanation of how HTK was used is given in Appendix B. Experiments were first run on the main (volunteer) part of TCD-TIMIT, and afterwards run on the secondary (lipspeaker) part.

6.2.1 Volunteer Experiments

For the audio-only baselines, experiments were run on TCD-TIMIT and on TIMIT for comparison. The purpose of the TIMIT results was to provide a reference to aid in evaluating whether the TCD-TIMIT results were robust. As such, the TIMIT results themselves needed to be verified, so they were compared with other TIMIT baselines in the literature (Section 4.1.2). Both sets of results were deemed to be reasonable.

To run the visual-only baseline experiments, 44-length DCT coefficients and 4-state HMMs were chosen after experimentation (Figure 4.3). Overall baseline results were lower than others

in the literature, but the trends in the results were found to be similar.

The audio-visual baseline experiment set up was to evaluate performance on increasingly-noisy audio, a common experiment in the literature. An audio-only recognizer was trained and tested in the same manner for comparison. The audio-visual recognizer was found to be more accurate than the audio-only recognizer below an SNR of 20dB. Above 20dB, the high dimensionality of the vectors and dominant influence of the visual stream led to the audio-visual recognizer having lower accuracy than its audio-only counterpart.

6.2.2 Lipspeaker Experiments

The purpose of the lipspeaker section of TCD-TIMIT was to investigate whether lipspeakers have a performance advantage over non-lipspeakers in automatic visual speech recognition systems. Lipspeaker data was first tested on a volunteer-trained recognizer, then a new recognizer was trained and tested on equal amounts of data from each lipspeaker and volunteer. Lipspeaker performances were found to be below-average in both cases.

The next step taken was to train and test a recognizer entirely on lipspeaker data and compare its performance to a recognizer trained and tested entirely on volunteer data. Results from the lipspeaker-trained recognizer surpassed those of the volunteer-trained recognizer by a large margin (Table 5.1). There are some caveats to the comparison: the lipspeaker-trained recognizer was trained on only 3 speakers compared to the volunteer recognizer's 56, and saw 251 sentences from each speaker compared to the volunteer recognizer's 67. Following these results, the volunteer data was tested on the lipspeaker-trained recognizer. It was speculated that training on lipspeaker data might produce a more robust recognizer for any speaker. However, the volunteers' results were lower than results from a speaker-independent volunteer-trained recognizer, offering no evidence to support this speculation.

Finally, audio-visual baselines were obtained on the lipspeakers using the same audio-visual experiment run previously on the volunteers. The results show that the lipspeaker-trained audio-visual recognizer begins to outperform its audio-only counterpart at an SNR of 22dB. This represents a 2dB gain over the volunteer-trained recognizer. Audio-visual results were higher for lipspeakers at every SNR compared to volunteers. At the final SNR of -10dB, volunteer and lipspeaker audio-only accuracy was roughly 8%, while volunteer audio-visual accuracy was 18.32% and lipspeaker audio-visual accuracy was 25.86%. This concluded the baseline experiments run on TCD-TIMIT.

One general trend observed was that individuals who had performed well in the visual-only experiments performed better than others in the audio-visual experiments. The converse is also true for poor visual-only speakers. This can be seen in the volunteer and lipspeaker results. Another was that the best-performing visemes in the volunteer and lipspeaker experiments were consistently /D/, /A/, /C/ and /F/. According to the Jeffers map (Table 2.4), these visemes all involve relatively distinctive lip shapes and positions. In contrast, the worst-performing visemes

were /J/ and /K/. These are ranked as having the worst visibility in the Jeffers map, because they are reliant on tongue positions. To recognize these visemes more reliably, a visual feature set would need a method of extracting this tongue information.

6.3 Future Work

The most immediate task at hand is to make TCD-TIMIT available for other researchers. Post-distribution, there are some specific experiments that would expand on the work done in this thesis.

- A more robust experiment comparing lipspeakers and volunteers could be run. The problem at present is that there is more data per lipspeaker than per volunteer. As a solution, a small number of the best and/or worst-performing volunteers (Table 4.8) could be brought back to record more sentences.
- The baseline experiments run on the straight camera's footage could be re-run on the 30° camera's footage to compare the results. No experiments have been attempted on this footage as of yet.
- The intention is to supply a set of phoneme and viseme-level label files with the database, but a number of other phoneme and viseme sets have been proposed in the literature. The label files could be altered to use a different phoneme/viseme set and the baseline experiments re-run.

As well as the experiments above, TCD-TIMIT can and hopefully will be used to investigate all parameters of VSR/AVSR: statistical models, ROI extraction methods, visual feature parameters, training regimes, audio-visual integration strategies etc. These are the many open questions of AVSR. The hope is that TCD-TIMIT will prove itself a useful tool in answering them.

Bibliography

[1] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4277–4280, 2012.

[2] E. Bailly-baillire, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Marithoz, J. Matas, K. Messer, F. Pore, and B. Ruiz. The BANCA database and evaluation protocol. In *In Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA03*, pages 625–638. Springer-Verlag, 2003.

[3] W. Branford. *The elements of English.* Routledge and Kegan Paul, London, 1967.

[4] F. Brugnara, D. Falavigna, and M. Omologo. Automatic segmentation and labeling of speech based on Hidden Markov Models. *Speech Commun.*, 12(4):357–370, Aug. 1993.

[5] I. R. BS.1770-3. Algorithms to measure audio programme loudness and true-peak audio level. `http://www.itu.int/rec/R-REC-BS.1770`, 2010.

[6] D. Burnham, E. Ambikairajah, and J. A. et al. A blueprint for a comprehensive Australian English auditory-visual speech corpus.

[7] L. Cappelletta. What Is a Viseme? Exploring the Visual Side of Automatic Speech Recognition. Master's thesis, Trinity College Dublin, College Green, Dublin 2, Ireland, 2012.

[8] T. Chen. Audiovisual speech processing. *Signal Processing Magazine, IEEE*, 18(1):9–21, 2001.

[9] C. Chibelushi, F. Deravi, and J. S. D. Mason. A review of speech-based bimodal recognition. *Multimedia, IEEE Transactions on*, 4(1):23–37, 2002.

[10] A. G. Chitu and L. Rothkrantz. Building a data corpus for audio-visual speech recognition. In *Euromedia2007*, page 88–92, apr 2007.

[11] CMU. The Carnegie Mellon University pronouncing dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

[12] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120:2421, 2006.

[13] K. H. Davis, R. Biddulph, and S. Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

[14] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, 1980.

[15] DeafHear.ie. "Lipspeakers". `http://www.deafhear.ie/DeafHear/lipSpeakers.html`, July 2013.

[16] K. Driel. Building a Visual Speech Recognizer. Master's thesis, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands, 2009.

[17] N. P. Erber. Interaction of audition and vision in the recognition of oral speech stimuli. *J Speech Hear Res*, 12(2):423–425, 1969.

[18] FFmpeg.org. `http://www.ffmpeg.org/`.

[19] M. Filppula. *The Grammar of Irish English: Language in Hibernian Style.* Routledge Studies in Germanic Linguistics. Taylor & Francis, 2002.

[20] N. Fox, B. O'Mullane, and R. Reilly. The realistic multi-modal VALID database and visual speaker identification comparison experiments. In *5th International Conference on Audio- and Video-Based Biometric Person Authentication*, AVBPA-2005 Proceedings, New York, 2005.

[21] S. Furui. *Digital Speech Processing, Synthesis, and Recognition.* New York and Basel: Marcel Dekker, Inc., 1989.

[22] G. Galatas, G. Potamianos, and F. Makedon. Audiovisual speech recognition incorporating facial depth information captured by the Kinect. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 2714–2717, 2012.

[23] G. Galatas, G. Potamianos, A. Papangelis, and F. Makedon. Audio visual speech recognition in noisy visual environments. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '11, pages 19:1–19:4, New York, NY, USA, 2011. ACM.

[24] T. Gan. *Bimodal Speech Recognition.* PhD thesis, Universitt Hamburg, Von-Melle-Park 3, 20146 Hamburg, 2012.

[25] T. Ganchev. *Contemporary Methods for Speech Parameterization*. SpringerBriefs in electrical and computer engineering. Springer New York, 2011.

[26] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. DARPA TIMIT acoustic phonetic continuous speech corpus CDROM, 1993.

[27] A. C. Gimson. *Gimson's pronunciation of English*. Arnold, London, 5. ed., 7. imp. edition, 1998.

[28] R. Goecke and J. B. Millar. A detailed description of the AVOZES data corpus, 2004.

[29] K. Gorman. Automatic speech segmentation with HTK. `http://www.ling.upenn.edu/~kgorman/papers/segmentation/.speechseg.html`.

[30] J. Gowdy, A. Subramanya, C. Bartels, and J. Bilmes. DBN based multi-stream models for audio-visual speech recognition. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 1, pages I–993–6 vol.1, 2004.

[31] T. J. Hazen, K. Saenko, C.-H. La, and J. R. Glass. A segment-based audio-visual speech recognizer: data collection, development, and initial experiments. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, pages 235–242, New York, NY, USA, 2004. ACM.

[32] M. Heckmann, K. Kroschel, C. Savariaux, and F. Berthommier. DCT-based video features for audio-visual speech recognition. In *INTERSPEECH*, 2002.

[33] R. Hickey. *Irish English: History and Present-Day Forms*. Studies in English Language. Cambridge University Press, 2007.

[34] A. Hines and N. Harte. Error metrics for impaired auditory nerve responses of different phoneme groups. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[35] J.-P. Hosom. *Automatic Time Alignment of Phonemes Using Acoustic-Phonetic Information*. PhD thesis, Oregon Graduate Institute of Science and Technology, 2000.

[36] J.-P. Hosom. Speaker-independent phoneme alignment using transition-dependent states. *Speech Commun.*, 51(4):352–368, Apr. 2009.

[37] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[38] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[39] R. Jakobson, C. G. M. Fant, and M. Halle. *Preliminaries to Speech Analysis: The Distinctive Features and Their Correlates*. The MIT Press, 1969.

[40] J. Jeffers and M. Barley. *Speechreading (lipreading)*. Thomas, 1971.

[41] S. Kapadia, V. Valtchev, and S. Young. MMI training for continuous phoneme recognition on the TIMIT database. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 491–494 vol.2, 1993.

[42] A. Karpov, A. Ronzhin, K. Markov, and M. Zelezn. Viseme-dependent weight optimization for CHMM-based audio-visual speech recognition. In T. Kobayashi, K. Hirose, and S. Nakamura, editors, *INTERSPEECH*, pages 2678–2681. ISCA, 2010.

[43] F. Kelly. Hidden Markov model based continuous speech recognition, April 2009. Undergraduate Honors Thesis.

[44] T. Kleinschmidt, D. Dean, S. Sridharan, and M. Mason. A continuous speech recognition evaluation protocol for the AVICAR database. In *International Conference On Signal Processing and Communication Systems*, Gold Coast, Australia, 2007.

[45] K. Kumar, T. Chen, and R. Stern. Profile view lip reading. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–429–IV–432, 2007.

[46] L. F. Lamel, R. Kassel, and S. Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Proceedings of the DARPA Speech Recognition Workshop*, pages 100–110, 1986.

[47] Y. Lan, R. Harvey, B.-J. Theobald, E.-J. Ong, and R. Bowden. Comparing visual features for lipreading. In *In AVSP-2009*, pages 102–106, 2009.

[48] Y. Lan, B.-J. Theobald, and R. Harvey. View independent computer lip-reading. In *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo*, ICME '12, pages 432–437, Washington, DC, USA, 2012. IEEE Computer Society.

[49] K.-F. Lee and H.-W. Hon. Speaker-independent phone recognition using hidden Markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(11):1641–1648, 1989.

[50] X. Lin, H. Yao, X. Hong, and Q. Wang. HIT-AVDB-II: A new multi-view and extreme feature cases contained audio-visual database for biometrics. In *Advances in Intelligent Systems Research*, JCIS-2008 Proceedings, pages 432–437, Harbin Institute of Technology Shenzhen Graduate School, 2008. JCIS.

[51] C. Lopes and F. Perdigao. Phoneme recognition on the TIMIT database. In *Speech Technologies*, pages 285–302. InTech, 2011.

[52] P. Lucey and G. Potamianos. Lipreading using profile versus frontal views. In *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 24–28, 2006.

[53] J. Mason, F. Deravi, C. Chibelushi, and S. Gandon. DAVID - digital audio visual integrated database. Technical report, Speech and Image Research Group at University of Wales Swansea, 1996.

[54] I. Matthews, T. Cootes, J. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(2):198–213, 2002.

[55] I. Matthews, G. Potamianos, C. Neti, and J. Luettin. A comparison of model and transform-based visual features for audio-visual LVCSR. In *Proc. Int. Conf. Multimedia Expo*, pages 22–25, 2001.

[56] H. Mcgurck and J. W. Macdonald. Hearing lips and seeing voices. *Nature*, 264(246-248), 1976.

[57] Merriam-Webster.com. `http://www.merriam-webster.com/dictionary/phone`, Jun 2013.

[58] K. Messer, J. Matas, J. Kittler, and K. Jonsson. XM2VTSDB: the extended M2VTS database. In *Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77, 1999.

[59] J. Movellan. Visual Speech Recognition with Stochastic Networks. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in neural information processing systems*, volume 7, pages 851–858. MIT Press Cambridge, San Mateo, CA, 1995.

[60] C. Neti, G. Potamianos, J. Leuttin, I. Matthews, H. Glotin, D. Vergyri, J. Sisson, A. Mashari, and J. Zhou. Audio-visual speech recognition. Technical report, CLSP Summer Workshop, Johns-Hopkins University, Baltimore, MD, 2000.

[61] A. M. Noll. Short-Time "Cepstrum" Pitch Detection. *Acoustical Society of America Journal*, 36:1030, 1964.

[62] G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(3):423–435, 2009.

[63] A. Pass, J. Zhang, and D. Stewart. An investigation into features for multi-view lipreading. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2417–2420, 2010.

[64] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy. CUAVE: a new audio-visual database for multimodal human-computer interface research. In *In Proc. ICASSP*, pages 2017–2020, 2002.

[65] B. L. Pellom and J. H. Hansen. Automatic segmentation and labeling of speech recorded in unknown noisy channel environments. *Speech Comm., November*, 1998.

[66] E. Petajan. Automatic lipreading to enhance speech recognition. In *Proceedings of the IEEE Communication Society Global Telecommunications Conference*, volume 1, pages 265 – 272 vol.1, 1984.

[67] D. Pogue. Talk to the machine: Progress in speech-recognition software. `http://www.scientificamerican.com/article.cfm?id=talk-to-the-machine`, 2010.

[68] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003.

[69] G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-visual automatic speech recognition: An overview. *Issues in Visual and Audio-Visual Speech Processing*, 22:23, 2004.

[70] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[71] R. Rose and P. Momayyez. Integration of multiple feature sets for reducing ambiguity in ASR. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–325–IV–328, 2007.

[72] T. Saitoh and R. Konishi. A study of influence of word lip reading by change of frame rate. In *In AVSP-2010*, 2010.

[73] C. Sanderson. *Biometric person recognition: Face, speech and fusion*. VDM Publishing, 2008.

[74] P. Scanlon, R. Reilly, and P. d. Chazal. Visual feature analysis for automatic speechreading. In *AVSP 2003-International Conference on Audio-Visual Speech Processing*, 2003.

[75] F. Schiel. A tutorial to HTK (part I + II). `http://www.phonetik.uni-muenchen.de/forschung/publikationen/Schiel_HTK.txt`.

[76] R. Seymour, D. Stewart, and J. Ming. Comparison of image transform-based features for visual speech recognition in clean and corrupted videos. *J. Image Video Process.*, 2008:14:1–14:9, Jan. 2008.

[77] S. Siniscalchi, P. Schwarz, and C.-H. Lee. High-accuracy phone recognition by combining high-performance lattice generation and knowledge based rescoring. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–869–IV–872, 2007.

[78] K. Sjölander. An HMM-based system for automatic segmentation and alignment of speech. In *In Proceedings of Fonetik 2003*, pages 93–96, 2003.

[79] V. D. Society. Victorian deaf society information sheet. `http://www.vicdeaf.com.au/files/editor_upload/File/Information%20Sheets/Speechreading%20_Visual%20Cues_.pdf`, 2010.

[80] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[81] C. Sui, S. Haque, R. Togneri, and M. Bennamoun. A 3D audio-visual corpus for speech recognition. In *Proc. of SST*, 2012.

[82] Q. Summerfield. Some preliminaries to a comprehensive account of audio-visual speech perception. In Dodd, editor, *Hearing by eye: The psychology of lip-reading*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1987.

[83] K. Vertanen. Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments. Technical report, Cavendish Laboratory, University of Cambridge, 2006.

[84] A. Vorwerk, X. Wang, D. Kolossa, S. Zeiler, and R. Orglmeister. WAPUSK20 - a database for robust audiovisual speech recognition. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *LREC*. European Language Resources Association, 2010.

[85] S. . Wilcox. MPEG encoding basics. `http://www.media-matters.net/docs/resources/Digital%20Files/MPEG/MPEG%20Encoding%20Basics.pdf`.

[86] Y. W. Wong, S. I. Chng, K. P. Seng, L.-M. Ang, S. W. Chin, W. J. Chew, and K. H. Lim. A new multi-purpose audio-visual UNMC-VIER database with multiple variabilities. *Pattern Recognition Letters*, 32(13):1503 – 1510, 2011.

[87] S. Young. The general use of tying in phoneme-based HMM speech recognisers. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 569–572 vol.1, 1992.

[88] S. J. Young. The general use of tying in phoneme-based HMM speech recognisers. In *Proceedings of the 1992 IEEE international conference on Acoustics, speech and signal processing - Volume 1*, ICASSP'92, pages 569–572, Washington, DC, USA, 1992. IEEE Computer Society.

[89] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.

[90] J. Yuan and M. Liberman. Speaker identification on the SCOTUS corpus. In *Proceedings of Acoustics 2008*, 2008.

[91] X. Zhang, C. C. Broun, R. M. Mersereau, and M. A. Clements. Automatic speechreading with applications to human-computer interfaces. *EURASIP J. Appl. Signal Process.*, 2002(1):1228–1247, Jan. 2002.

A

Phonetic Traits of Hiberno-English

While obtaining force-aligned phoneme label files for TCD-TIMIT using P2FA, it was felt that the pronouncing dictionary in use (the CMU dictionary) could benefit from some Hiberno-English pronunciations. The CMU dictionary was based on American pronunciations. All but 4 speakers in TCD-TIMIT have Irish accents, with the majority of these being "neutral" Dublin accents. By "neutral" it is meant that some of the strongest characteristics of Dublin accents are not present or only faintly present in the speakers' accents. Nevertheless, it was felt that the accents were different enough to American accents to warrant trying to add Hiberno-English pronunciations to the dictionary. A set of rules were sought that could be applied to words already in the dictionary to add an alternative "Irish" pronunciation below these.

The books "Irish English: History and Present-Day Forms" by Hickey [33] and "The Grammar of Irish English: Language in Hibernian Style" by Filppula [19] were referred to in an attempt to build rules for Hiberno-English. Also, emails were exchanged with Dr. Amelia Kelly, a researcher who has previously phonetically described a dialect of Hiberno-English and worked with speech synthesis of Irish accents. Between these three sources, a set of rules was created and applied to the CMU dictionary. The rules are given in Table A.1.

Even if the rules produce useless alternative pronunciations for some words, it will not affect P2FA's forced alignment, because the most likely word is always chosen. The only reason not to have many pronunciations for each word is the additional time it will take P2FA to determine the most likely one.

Table A.1: Hiberno-English Rules Applied to CMU Dictionary. The full CMU phoneme set can be seen in Table 3.3.

Example Word	Original Phonetic Transcription	Rule Applied	Alternate Phonetic Transcription
this	DH IH1 S	DH ->D	D IH1 S
I'm	AY1 M	AY ->OY	OY1 M
three	TH R IY1	TH ->T	T R IY1
validate	V AE1 L AH0 D EY0 T	T ->AH1	V AE1 L AH0 D EY0 AH1
butter	B AH1 T ER0	AH1 AH1 ->UH0	B UH0 ER0

B
HTK

HTK is a commonly-used toolkit to set up and test HMMs. It was used in this work to create the HMM-based audio, visual and audio-visual speech recognizers for all experiments. This appendix details the exact HTK workflows that were used.

There are many good resources, both printed and on the web, for training and testing HMMs with HTK. Some of the most useful references found during this project include Kyle Gorman's "Automatic speech segmentation with HTK" online tutorial [29], a tutorial webpage from the phonetics department at LMU Munich [75] and the HTK book itself [89].

B.1 Audio-only Recognizers

This section details the HTK workflow used for the audio-only recognition experiments in Chapters 3, 4 and 5.

To begin, the necessary files were collected. These are listed below:

- A set of audio files (WAV files were used) for training and testing.
- A phoneme-level label file for each WAV file. These can be contained in one large HTK-format Master Label File (MLF).
- A prototype HMM file. The prototype file used is given in Figure B.2.
- A configuration file. The configuration file used is given in Figure B.1.

Figure B.1: Config file used to create MFCC files

```
#MFCC config settings
SOURCEFORMAT = WAV          # The format of the audio files
TARGETFORMAT = HTK          # "HTK" output format
TARGETKIND = MFCC_D_A_Z     # D = Deltas, A = , Z = Cepstral Mean Normalization
TARGETRATE = 100000.0       # get coefficients every 10000ns (sample rate 100Hz)
SAVEWITHCRC = T             # use check bit
WINDOWSIZE = 250000.0       # 25000ns window
USEHAMMING = T              # use Hamming window
PREEMCOEF = 0.97            # 1st order preemphasis
NUMCHANS = 26               # 26 channels filteration
CEPLIFTER = 22              # 22 cepstral filters
NUMCEPS = 12                # 12 MFCC coefficients
ENORMALISE = F              # do not normalize intensity
```

Figure B.2: Proto file used to create HMMs

```
~h "proto"
<BEGINHMM>
<VecSize> 36 <MFCC_D_A_Z>
<NumStates> 5
<State> 2
<Mean> 36
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 36
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 3
<Mean> 36
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 36
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<State> 4
<Mean> 36
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
<Variance> 36
1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
<TransP> 5
0.0 0.5 0.5 0.0 0.0
0.0 0.4 0.4 0.2 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<ENDHMM>
```

- A file containing the phoneme set in use. This is not strictly necessary, but if the phonemes are not in a file they must be passed one by one as parameters to HTK functions.

After collecting these files, the WAV files were converted into MFCC feature files. The configuration file "wavconfig", given in Figure B.1, was used to specify how the MFCCs were to be created. The HTK command for the conversion is HCopy. To avoid passing each WAV file in as a parameter one-by-one, they were placed in a HTK script file (allWavs.scp). Each line of the script file consists of the path to a WAV file, a space, then the path to the corresponding MFCC file which is to be created. The HCopy command was:

```
# HCopy -C wavconfig -S allWavs.scp
```

HCompV was then used to initialize the means and variances of the prototype matrix (proto, Figure B.2) to the global means and covariances of the training set (trainSet.scp). The corresponding label files were given in "allLabels.mlf". The command was:

```
# HCompV -C config -f 0.01 -m -S trainSet.scp -M hmm1 -X rec -I allLabels.mlf proto
```

Variance information is output by HCompV into a file called "vFloors". Following the advice in Kyle Gorman's tutorial [29], this was copied into a new file (macros) with some additional header information. Then, for each phoneme in the phoneme list file (monophones0.list), the "proto" file (Figure B.2) was copied, the word "proto" in the copy was replaced by the name of that phoneme, and the name of the copy was also changed to the name of that phoneme. The outcome is a HMM file for each phoneme ("aa.txt", "ae.txt" etc.), although their parameters are currently all the same.

The next HTK command, HInit, initializes one phoneme at a time. To initialize every phoneme, a script was written to parse each line of the phoneme list file (monophones0.list) and run HInit on that phoneme's HMM file. Note also that the "SOURCEFORMAT" field in the configuration file of Figure B.1 was changed to "HTK". The HInit command (shown here for /aa/) was as follows:

```
# HInit -C config -S trainSet.scp -H hmm1/macros -H hmm1/aa.txt -M hmm1/0 -X rec -I allLabels.mlf
-l "aa" aa.txt
```

The Baum-Welch re-estimation function HRest was then called in the same manner (again shown here only for /aa/):

```
# HRest -C config -S trainSet.scp -H hmm1/0/macros -H hmm1/0/aa.txt -M hmm1/1 -X rec -I allLabels.mlf
-l "aa" aa.txt
```

At this point the HMM files for each phoneme were all combined in one large Master Macro File (MMF). This was more convenient. The embedded version of the Baum-Welch algorithm HERest was then run three times. The first call is given below:

```
# HERest -S trainSet.scp -H hmm1/1/macros -H hmm1/1/hmm.mmf -X rec -I allLabels.mlf -M hmm1/2
monophones0.list
```

The next modification was to accommodate for short pauses, denoted with an /sp/ phoneme. Up to this point, the only silence phoneme was /sil/. This concept of having different silence phonemes for long and short pauses is from HTK's own tutorial [89]. The implementation used was based on Kyle Gorman's implementation [29]. A new one-state HMM called /sp/ was created, and its centre state was tied to that of /sil/. However, to actually make use of this /sp/ phoneme, the next step would be to change label files to a set containing instances of /sp/. Since this set was not available, the /sp/ model created in this step was not actually used. Nevertheless, it was created, so the HHEd command used is given below. "sil.hed" is given in

Figure B.3: Script file input to HHEd to tie /sil/ to /sp/

```
1  AT 2 4 0.2 {sil.transP}
2  AT 4 2 0.2 {sil.transP}
3  AT 1 3 0.3 {sp.transP}
4  TI silst {sil.state[3],sp.state[2]}
```

Figure B.3 and "monophones1.list" is just the list of phonemes from "monophones0.list" with the addition of /sp/.

```
# HHEd -H hmm1/3/hmm.mmf -M hmm1/4 sil.hed monophones1.list
```

A number of iterations of increasing mixtures and embedded re-estimation runs were then performed. Five re-estimation runs were performed for each mixture incrementation. The mixture incrementations were as follows: 2, 3, 5, 7, 9, 11, 14, 17, 20, 23, 26, 29, 31. The format of the HERest calls was the same as the example given above. A typical HHEd command is given below (in this example, "2mix.hed" contained the script commands for HHEd to increase the number of mixtures to 2).

```
# HHEd -H hmm1/10/macros -H hmm1/10/hmm.mmf -M hmm1/11 2mix.hed monophones0.list
```

The training complete, HTK's Viterbi algorithm function was used to generate output label files on a test set. The format of the HVite call was:

```
# HVite -o SM -C config -H hmm1/88/macros -H hmm1/88/hmm.mmf -i outputLabels.mlf -w phn_net
-S testSet.scp phn.dict monophones0.list
```

There are a few additional files used here. "phn.dict" is a "dictionary" file for the phonemes. This is a holdover from HTK's original use as a word-level recognition tool. For word-level recognition, HTK would consult a transcription dictionary to find which words match the sequence of phonemes it produced. For phoneme-level recognition, the phonemes themselves are the desired output, so a phoneme "dictionary" is simply a file mapping each phoneme to itself. "phn_net" is a phoneme network file derived from the label files and a phoneme language model file. Language model files can convey useful information about the likelihood of certain phonemes after others. In this case, the simplest language model possible was used, making every sequence of phonemes equally possible, however the model does specify that the utterances begin and end with /sil/. HParse was run on this language model to create the "phn_net" file.

With a set of output label files obtained, finally HResults was called to compare the output to the ground-truth "allLabels.mlf" file. The HResults call was as follows:

```
# HResults -I allLabels.mlf monophones0.list outputLabels.mlf
```

```
1 AT 3 2 0.2 {S.transP}
2 AT 2 3 0.2 {S.transP}
```

(a) 3-state HMMs)

```
1 AT 4 2 0.2 {S.transP}
2 AT 4 3 0.2 {S.transP}
3 AT 3 2 0.2 {S.transP}
4 AT 2 4 0.2 {S.transP}
```

(b) 4-state HMMs

```
1 AT 5 2 0.2 {S.transP}
2 AT 5 3 0.2 {S.transP}
3 AT 5 4 0.2 {S.transP}
4 AT 4 2 0.2 {S.transP}
5 AT 4 3 0.2 {S.transP}
6 AT 3 2 0.2 {S.transP}
7 AT 2 5 0.2 {S.transP}
```

(c) 5-state HMMs

Figure B.4: Silence viseme state-tying commands for 3, 4 and 5-state HMMs

B.2 Visual and Audio-Visual Recognizers

For visual and audio-visual recognizers, Cappelletta's [7] workflow was followed. Although visemes are referred to below, audio-visual recognizers used phonemes. The workflow was the exact same in both cases. Also worth noting is that the audio-only recognizers for the noisy audio experiments of Sections 4.3 and 5.3 were trained using this workflow.

Cappelletta's workflow differed slightly from the audio-only workflow described in Section B.1. HCopy was not used for the visual feature files. A Matlab script written by Mike Brookes converted the DCT vectors to HTK-compatible visual feature files instead. HCompV was not called. The prototype HMM was simply copied for each viseme. HInit was the first HTK function called. In this case, an additional parameter was specified, "-v 0.0001", which set the variance floor to 0.0001. After running HInit, the call to HRest was skipped. The viseme HMMs were combined in one large MMF and HERest was called 5 times. The format of the HERest command was the same as the one given in Section B.1.

The modification for short pauses was also different. No /sp/ viseme was added. Instead, states were simply tied to allow the silence viseme to be traversed quicker. The HHEd script files had to be changed depending on whether 3, 4 or 5-state HMMs were in use. These files are given in Figure B.4.

After tying the states of the silence viseme, the next step was to run HRest on the silence HMM alone. In order to do this, the silence viseme was extracted from the MMF containing the other viseme HMMs, HRest was run on it, and it was then re-inserted into the MMF. The HRest call is given below:

```
# HRest -i 30 -S trainSet.scp -H hmm1/S/S.mmf -I allLabels.mlf -l "S" S.txt
```

At this point, the iterations of embedded re-estimation and mixture incrementation began. The mixture incrementations were as follows: 2, 3, 5, 7, 9, 11, 14, 17, 20. HERest was run 5 times after each incrementation. The HHEd commands and scripts were the same as in Section B.1, and the HERest command was virtually the same apart from the additional parameter of "-v 0.0001".

For results, the HVite and HResults calls were the same as Section B.1.

C

Consent Form for Database Subjects

Consent Form

TCD Audio-Visual TIMIT Database

Principal Investigator: Dr. Naomi Harte

Unique ID

Background:

Humans use both visual and audio cues to understand speech. When listening to a person, we both hear their speech and watch their lips to aid in our understanding. Speech recognition by computers (e.g. Siri on the iPhone or Dragon Dictate) uses only speech cues. When surroundings become noisy, humans begin to rely more and more on reading a person's lips to augment the poor quality of what they hear. We are filming this database to help us research and understand what visual information is useful to extract from speakers to improve speech recognition by machines in noisy environments.

Aim:

We aim to collect video and audio of 60 speakers, each talking 100 sentences in English. This video will be edited and packaged to make a single database of speakers available to the audio-visual speech recognition community.

Procedure:

You will be given a unique ID number and asked to sign this consent form. The person in charge of filming the session will give you complete instructions. You will sit in front of a green screen for the filming. A monitor will be placed conveniently to allow you read the sentences. Please speak in a normal manner with minimal head movement. The person filming may ask you to repeat sentences. You need to start each sentence with your mouth in a closed, rested position.

Data Protection:

Your part in this study will be kept anonymous. A list of names and corresponding Unique Ids will be kept seperately and not distributed. All hard copy data for this experiment will be stored in a secure area with access limited to the investigators team. All electronic data will be stored in password protected files.

All videos will be edited down to a sequence of sentences. The video material from these sentences will form the database. The database may be distributed in the future to other research groups for academic reserach only. No profit shall be made from the distribution of the database. The following information for a speaker will be shared: age, gender, native/non native english speaker.

There is no obligation to participate and you may withdraw from this study at any time. If you do decide to withdraw during the filming, please make this clear to the person filming. Any recording will be deleted and not used.

Sigmedia

DECLARATION:

I have read, or had read to me, the information leaflet for this project and I understand the contents. I have had the opportunity to ask questions and all my questions have been answered to my satisfaction. I freely and voluntarily agree to be part of this research study, though without prejudice to my legal and ethical rights. I understand that I may withdraw from the study at any time and I have received a copy of this agreement.

GENDER (M/F):.....

DATE OF BIRTH (DD/MM/YYYY):.....

PARTICIPANT'S NAME:……...

PARTICIPANT SIGNATURE:……...

DATE:……...

(IF UNDER 18) SIGNATURE OF

PARENT/GUARDIAN:.....

Contact: Dr. Naomi Harte, Eoin Gillen, Dept. of Electronic & Electrical Engineering, Printing House, Trinity College Dublin, Dublin 2. Email: nharte@tcd.ie ogiollae@tcd.ie

Sigmedia

D

Individual Visual-Only Results

Results for each volunteer used in the experiments of Sections 4.2 and 4.3 are given in Tables D.1 and D.2. "Vis only" refers to the 4-state, 44-coefficient speaker-dependent experiment run in Section 4.2.1. "Audio-Visual" refers to the speaker-dependent audio-visual experiments on noisy audio run in Section 4.3.2. "Corr" is the correctness percentage, and "Acc" is the accuracy percentage.

Table D.1: Part 1 of visual-only and audio-visual results (%) for each volunteer from experiments of Sections 4.2 and 4.3. All results are given as percentages.

	Vis Only		Audio-Visual											
SNR	-		40dB		30dB		20dB		10dB		0dB		-10dB	
Spkr.	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc
01M:	40.58	33.09	52.12	34.75	49.23	30.69	42.18	26.16	33.69	20.46	26.35	19.02	22.39	16.6
02M:	16.68	16.49	46.71	36.68	46.63	36.94	42.45	32.95	34.9	26.2	27.44	21.4	18.74	15.99
03F:	50.87	40.10	51.55	39.86	50.04	39.42	45.88	35.96	38.62	28.96	32.42	24.62	26.93	19.84
04M:	42.16	36.81	55.86	39.57	52.64	39.57	47.63	35.36	41.36	29.72	30.98	22.11	24.44	18.53
05F:	40.12	32.91	49.82	34.2	46.77	31.42	39.68	27.2	34.56	23.61	24.15	19.3	15.08	13.82
06M:	42.32	36.17	41.49	31.25	41.39	32.7	36.32	27.99	33.88	24.64	28.26	21.65	21.83	18.3
07F:	41.08	36.26	49.69	37.51	47.77	35.58	43.3	31.55	37.42	27.34	28.4	21.91	20.86	17.44
08F:	43.68	36.11	53.79	38.73	50.62	36.91	44.01	32.31	39.02	28.67	32.6	23.3	27.42	20.61
09F:	49.03	38.03	54.42	40.68	50.58	38.8	45.5	33.27	40.5	29.08	31.4	23.02	26.49	19.27
10M:	46.07	36.77	48.03	38.84	45.4	36.59	41.56	32.46	37.62	28.33	31.61	25.7	24.11	19.32
11F:	51.09	38.37	54.46	36.25	50.64	35.43	46.45	31.06	38.98	25.96	31.97	19.58	27.05	17.3
12M:	51.90	39.67	49.76	35.79	47.4	33.81	42.97	30.88	37.11	27.67	30.97	21.72	24.83	18.22
13F:	47.48	37.60	52.67	37.3	49.86	39.27	47.05	37.3	43.3	32.8	33.55	24.74	25.4	18.93
14M:	45.44	34.00	54.41	35.63	52.98	36.24	48.67	32.75	42.09	28.34	33.68	21.66	23.41	17.04
15F:	36.37	32.00	52.24	39.31	48.01	35.59	42.27	30.35	36.01	25.19	27.81	20.88	25.27	18.6
16M:	44.24	32.72	49.46	39.41	46.93	38.44	43.12	34.05	38.24	29.56	32.88	25.37	24.59	20.49
17F:	46.97	35.20	50.99	37.34	49.87	37.34	45.98	34.75	40.1	31.2	33.28	24.29	23.08	18.5
18M:	45.96	40.37	51.47	44.71	51.18	44.22	47.35	39.61	42.25	35	35.88	28.73	28.04	24.12
19M:	45.71	34.66	48.67	39.68	46.21	37.88	42.23	36.08	36.46	31.25	29.92	25.47	23.2	20.64
20M:	44.88	40.24	46.44	34.27	43.07	32.75	37.87	29.46	32.11	24.66	26.26	19.86	22.1	16.81
21M:	41.92	34.96	49.48	38.81	44.41	33.22	40.53	29.69	34.68	24.96	28.4	21.77	22.29	18.59
22M:	50.44	40.39	47.89	35.7	45.56	35.25	44.22	33.54	39.46	29.96	31.03	22.33	23.95	19.64
23M:	45.47	33.08	51.54	39.48	47.68	37.26	45.08	33.98	40.35	29.44	35.42	24.42	28.19	20.75
24M:	41.49	35.58	55.61	41	53.91	38.08	52.03	35.72	46.28	28.93	36.48	24.88	25.35	19.89
25M:	39.74	35.58	49.95	37.66	46.51	37.56	41.79	34.32	36.28	29.7	29.11	23.4	18.49	16.42
26M:	47.79	33.86	56.3	39.93	54.24	37.79	49.53	32.31	40.27	26.91	32.39	21.51	21.85	16.71
28M:	43.57	32.78	45.05	32.8	41.56	29.59	38.83	27.33	31.1	23.09	24.98	20.64	19.7	16.87
29M:	29.61	28.77	43.84	31.81	41.83	30.95	37.15	28.94	31.33	22.92	25.5	19.1	19.77	15.47
30F:	44.68	32.03	49.59	36.41	45.84	36.14	40.99	31.47	36.69	26.26	29.09	22.69	24.15	19.03
31F:	47.84	37.13	59.29	40.53	57.64	41.17	52.24	35.59	42.82	27.17	33.85	20.31	26.08	16.83
32F:	45.30	35.80	52.27	29.82	46.47	27.59	42.11	25.94	36.5	21.01	28.17	17.91	21.49	15
33F:	46.02	40.99	53.7	35.03	51.48	34.47	47.5	33.73	40.2	27.45	32.26	21.44	25.88	18.67
34M:	30.80	29.22	46.25	32.6	43.04	30.86	38.37	27.38	32.88	21.43	25.27	15.84	17.95	15.48
36F:	42.29	37.28	51.2	39.12	46.32	36.06	43.01	34.74	36.72	28.04	30.19	22.75	24.73	19.02
37F:	43.99	39.30	48.15	32.94	48.69	33.75	46.26	31.86	39.51	27.45	33.21	20.97	25.38	16.74
38F:	46.74	34.42	51.28	39.74	47.44	36.67	42.56	33.42	37.18	28.97	30.94	23.16	24.79	20.17

Table D.2: Part 2 of visual-only and audio-visual results for each volunteer from experiments of Sections 4.2 and 4.3. All results are given as percentages.

	Vis Only		Audio-Visual											
SNR	-		40dB		30dB		20dB		10dB		0dB		-10dB	
Spkr.	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc	Corr	Acc
39M:	49.56	39.86	55.06	40.82	50.49	37.42	46.28	35.09	39.66	29.1	32.77	23.46	26.59	20.32
40F:	47.84	33.60	54.09	29.09	48.51	28.35	42.29	24.44	36.15	21.84	28.44	17.75	23.7	15.8
41M:	41.53	36.97	50.36	42.44	48	39.53	44.08	33.88	36.79	29.33	30.69	23.22	25.41	19.67
42M:	28.60	25.90	52.88	40.09	49.41	39.82	45.39	35.07	37.99	28.4	29.59	22.47	22.56	17.81
43F:	47.10	41.10	48	39.57	46.41	38.33	43.03	33.54	37.8	29.81	31.94	26.18	25.29	21.47
44F:	45.21	39.71	59.77	45.39	59.3	46.9	54.61	43.14	46.05	35.06	35.62	26.13	27.16	21.52
45F:	35.85	30.14	48.36	33.01	46.23	32.92	42.41	28.13	35.67	23.51	27.77	18.63	16.68	14.55
46F:	33.30	28.69	52.65	38.41	50.98	39.1	47.74	34.58	41.06	29.57	30.16	22.3	18.57	15.72
47M:	23.83	23.24	44.24	34.45	41.43	33.18	37.62	30.19	32.18	26.11	28.74	21.49	23.12	18.4
48M:	36.44	31.30	55.54	40.51	52.72	37.08	46.13	32.07	38.22	25.75	28.21	21.79	24.17	18.89
49F:	49.62	41.90	52.34	40.31	48.7	36.33	42.82	32.01	37.46	27.42	32.09	21.45	24.57	17.73
50F:	41.69	36.86	48.76	40.4	45	37.56	39.67	32.32	34.53	28.56	29.2	23.42	22.41	19.19
51F:	46.78	39.08	53.47	41.57	50.5	39.13	46.71	36.43	41.48	33.63	33.54	25.79	25.07	20.02
52M:	33.39	28.78	48.51	37.26	45.99	36.64	41.9	33.02	37.03	28.69	28.62	22.56	21.23	17.3
54M:	32.97	26.86	46.88	33.99	46.38	34.19	43.61	31.71	37.86	25.87	32.41	23.19	23.09	18.83
55F:	47.15	37.59	54.52	37.43	51.28	40.28	46.46	35.17	39.39	29.96	31.04	22.99	26.42	20.14
56M:	40.65	34.08	49.24	36.44	44.62	34.4	39.38	30.04	34.04	26.04	28.27	21.42	23.2	17.6
57M:	24.92	22.89	40.8	30.96	40.99	30.87	38.95	29.99	32.13	26.68	25.41	21.71	17.33	15.58
58F:	45.05	34.63	51.73	42.6	48.17	38.46	43.56	35.29	37.6	30.67	31.25	24.42	26.06	21.15
59F:	38.62	30.84	52.93	25.83	51.49	26.82	46.98	27.18	40.5	24.66	30.06	20.88	22.5	19.26