Soft Fusion methods for Multi Modal Speech applications

By

Dheeraj Kumar

To the

DEPARTMENT OF ELECTRICAL ENGINEERING

Indian Institute of Technology, Kanpur

April, 2010

Soft Fusion methods for Multi Modal Speech applications

A Thesis Submitted

In Partial Fulfillment of the Requirement

for the Degree of

Bachelor-Master of Technology (Dual)

by

Dheeraj Kumar

To the

DEPARTMENT OF ELECTRICAL ENGINEERING

Indian Institute of Technology, Kanpur

April, 2010

Certificate

This is to certify that the thesis entitled "**Soft Fusion methods for Multi Modal Speech applications**" by Dheeraj Kumar (Roll No. Y5827167) has been carried out under my supervision and guidance. This work has not been submitted elsewhere for the award of any degree.

26/4/10

Dr. Rajesh M Hegde
Asst. Professor
Department of Electrical Engineering,
Indian Institute of Technology,
Kanpur

IIT Kanpur
April 2010

Dedicated
To
My Family

Acknowledgements

I express my deepest sense of gratitude and regards to my thesis supervisor Dr. Rajesh M Hegde for his constant motivation, kind guidance and invaluable suggestion that have helped this thesis work to take a meaningful shape. His positive intent was the main force behind all the work. I am grateful to him for helping me develop critical thinking to excel in the field of research. It has been a great experience working under his guidance.

My friend at the hostel have always supported me in countless ways and have provided me a cheerful environment, I would like to thank them all for their support. Last but not the least I am extremely grateful to my parents and all other family members who have been a pillar of inspiration all along.

Dheeraj Kumar

Abstract

The complementary nature of audio and video information is well established. Video information about mouth shape and position can be used to interpret audio information in a better way. In this thesis, decisions made from audio and video information separately are late fused using Dempster Schafer (DS) theory, which provides a soft belief function for fusing information from independent modalities.

Speaker diarization is the problem of finding out speaking times of each speaker and grouping together homogeneous segments. This is an increasingly relevant problem in meeting room scenarios and for automatic meeting documentation. In this thesis, speaker diarization using audio only information is performed using Bayesian Information Criteria (BIC) and video based diarization is performed using Hidden Markov Model (HMM) modeling of speaking and non speaking segments and later the two decisions are fused using DS theory.

Speech recognition is the problem of finding out what is being said by listening to or by seeing someone speak or both. Applications of speech recognition are numerous and include better human-computer interface and speech controlled applications. In this thesis, speech recognition is performed using HMM modeling of audio and video features, and later the decisions made using these modalities are fused using DS theory.

Chapter 2 describes how humans make use of information gathered from multiple modalities like vision, hearing, touch, etc. to make better decisions based on integration of information from all these senses. Chapter 3 describes soft belief function and how is it used to combine audio and video information. The problem of "Speaker

Diarization" and "Audio Visual Speech Recognition" are explained in detail in Chapter 4 & 5. Chapter 6 describes the databases on which experiments are performed. Experimental results are shown in Chapter 7 before concluding in Chapter 8.

Based on the results of the experiments conducted for speaker diarization and audio visual speech recognition on standard databases, it can be concluded that video information when fused with audio information using a soft belief function such as DS theory gives better results than unimodal information based speaker diarization and speech recognition.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Speech recognition is a very interesting as well as challenging area of research. It's applications which include voice dialing, call routing, remote appliance control and content-based spoken audio search, simple data entry, preparation of structured documents and in speech-to-text processing are immense and much helpful to all sections of society. It also finds applications in diverse fields of health care and military. Motivation for this work comes from the human perception of speech [11]. We not only listens to speech but also see it. Audio is highly susceptible to noise, so performance of speech based systems falls dramatically in noisy environments. For the speech based systems to approach human level of performance and for speech to become a truly pervasive user interface we need some non traditional approaches to increase the performance of such systems. Visual speech is one such approach for making large improvements in high noise environments and has the potential of being channel and task independent [3].

1.2 Multi modal speech processing

Video information is quite useful in understanding speech. Lipreading is useful to all sighted people, including those with normal hearing and those with impaired hearing [7]. Biological systems, such as human sensory systems, can robustly and accurately localize objects even in the presence of significant amount of noise. One of the reasons behind this ability is the fact that they rely on the integration of several different senses instead of just a single sense [5]. Sense integration is effective because it allows the perception system to be applicable in a greater number of situations than would be possible with a single sense alone. Also, a given source of noise is likely to affect only one of the senses. For example, a vision system is unaffected by background sound sources in the environment, just as a sound localization system is unaffected by rapidly varying room lighting.

Visible articulators, primarily the lips, teeth and tongue, are among those whose configuration determines and is correlated with the resonances of vocal tract.

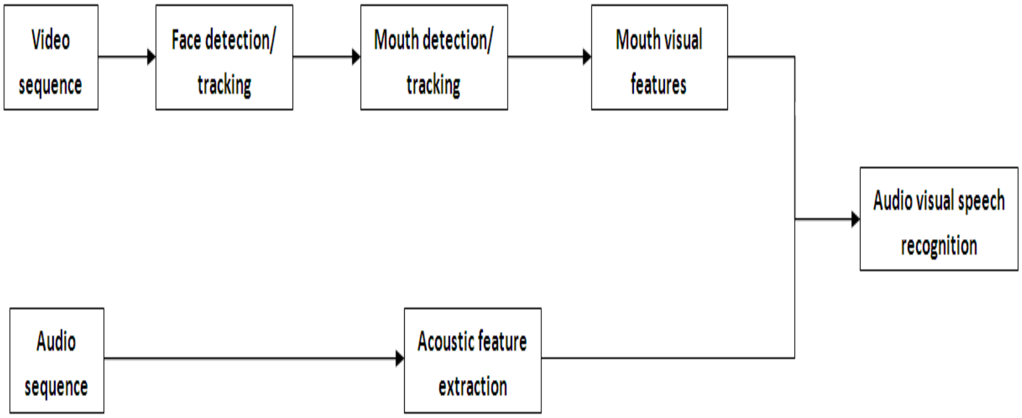
Video sequence
Face detection/ tracking
Mouth detection/ tracking
Mouth visual features
Audio visual speech recognition
Audio sequence
Acoustic feature extraction

Figure 1.1: *Block diagram of audio visual speech recognition system*

Multi-modal speech recognition, in which acoustic features and other information are jointly used, has been investigated and found to increase robustness and thus improve the accuracy of audio-visual speech recognition systems [4]. Most of the multi-modal methods use visual features, typically lip information, in addition to the acoustic features. In most of the studies, a lip contour is extracted from images by mouth tracking and pattern matching techniques [8].

In this work, audio and video modalities has been used to carry out the experiments.

1.3 Multi modal information fusion

Fusing different data channels and different representations can bring robustness and increased accuracy to learning algorithms. The mathematical and theoretical models developed in this field allow for a two-way information flow between computer science and cognitive sciences. Experimental results have shown that two modalities need not be in perfect temporal synchrony for fusion. If there is a fixed spatial or temporal distance between two percepts, and the percepts show structural similarities across these distances, this may lead to fusion as well.

In general, we can categorize methods of audio-visual information fusion into two broad categories: feature fusion (or early integration) and decision fusion (or late integration) [1], which are shown in Figure 1.2. In the former approach, the features of the two modalities are concatenated to form a composite feature vector, which is used as the classifier for recognition. In the latter approach, the features of each modality are used for recognition separately and, then, the outputs of the two classifiers are combined for the final recognition result.

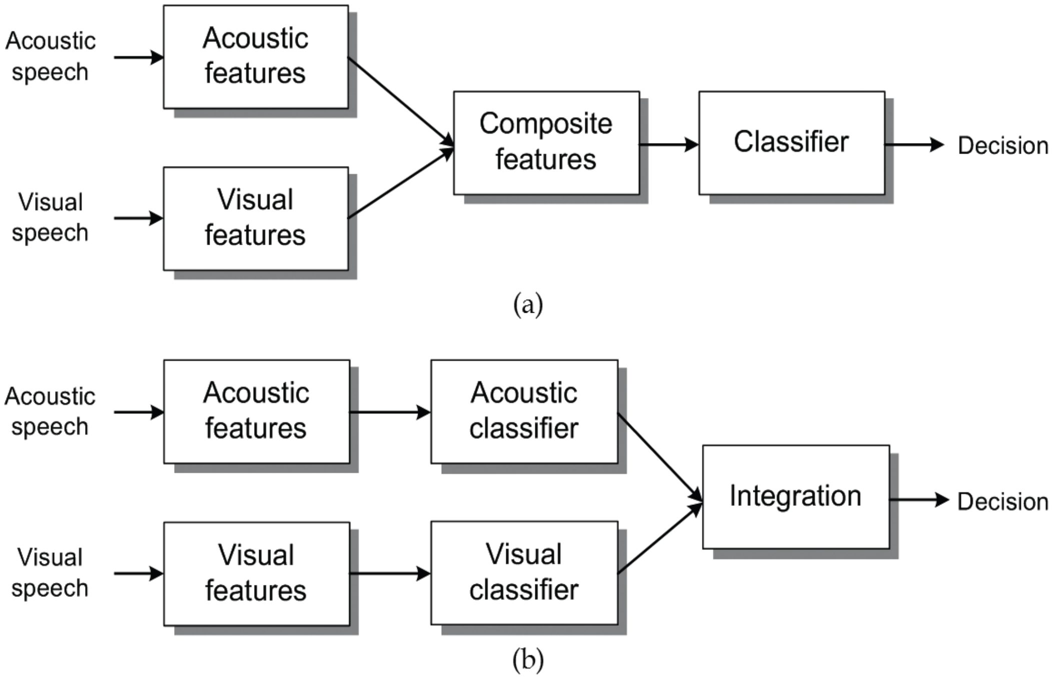

Figure 1.2: *Models for integrating acoustic and visual information. (a) Feature fusion. (b) Decision fusion*

In the decision fusion approach [17] it is relatively easy to employ an adaptive weighting scheme for controlling the amount of the contribution of the two modalities to the final recognition according to the noise level in the speech, which is because the acoustic and the visual signals are processed independently. The decision fusion allows flexible modeling of the temporal coherence of the two information streams, whereas the feature fusion assumes a perfect synchrony between the acoustic and the visual feature sequences. It is known that there exists an asynchronous characteristic between the acoustic and the visual speech: The lips and the tongue sometimes start to move up to several hundred milliseconds before the acoustic speech signal, which makes it difficult to synchronize the two signals.

1.4 Applications of audio visual speech processing addressed in this thesis

There are a lot of applications of audio visual speech processing. In this thesis, main focus is on two applications of audio visual speech processing viz “Speaker Diarization”' ' and “Audio Visual Speech Recognition”

1.4.1 Speaker diarization

In the context of audio document indexing and retrieval, speaker diarization is the process which detects speakers turns and regroups those uttered by the same speaker. It is generally based on a first step of segmentation (often preceded by a speech detection phase) that consists of partitioning the regions of speech into segments (each segment must be as long as possible and must contain ideally the speech of only one speaker), followed by a clustering step that consists of giving the same label to segments uttered by the same speaker. Ideally, each cluster corresponds to only one speaker and vice versa. Most of the systems operate without specific a priori knowledge of speakers or their number in the document, but in spite of tremendous progress, they generally need specific tuning and parameter training.

1.4.2 Audio visual speech recognition

Audio Visual Speech Recognition (AVSR) is a technique that uses image processing capabilities in lip reading systems to aid audio based speech recognition systems in recognizing indeterministic phones or giving preponderance among near probability decisions. Each system, lip reading and audio based speech recognition works separately and then the information gathered from them is fused together to make a better decision.

Aim of AVSR is to exploit the human perceptual principle of sensory integration (currently, the joint use of audio and visual information) to improve the recognition of human activity (e.g. speech recognition, speech activity, speaker change, etc.), intent (e.g. speech intent) and identity (e.g: speaker recognition), particularly in the presence of acoustic degradation due to noise and channel, and the analysis and mining of multimedia content.

2

Audio visual speech processing by humans

2.1 Speech production system in humans

Speech is a natural form of communication for human beings. Continuous speech is a set of complicated audio signals, which are usually classified as voiced or unvoiced, but in some cases they can be something between these two classes. Voiced sounds consist of fundamental frequency (F_0) and its harmonic components produced by vocal cords (vocal folds). The vocal tract modifies this excitation signal causing formant (pole) and sometimes antiformant (zero) frequencies.

Speech production mechanism is composed of four stages:

- Language processing, in which an utterance contents are converted into phonemic symbols in the brain's language center.
- Motor command generation to vocal organs in brain's motor center.
- Articulatory movement of the vocal organs based on these motor commands for the production of speech.
- Emission of air sent from the lungs in the form of speech.

When speaking, the air flow is forced through the glottis between the vocal cords and the larynx to the three main cavities of the vocal tract, the pharynx

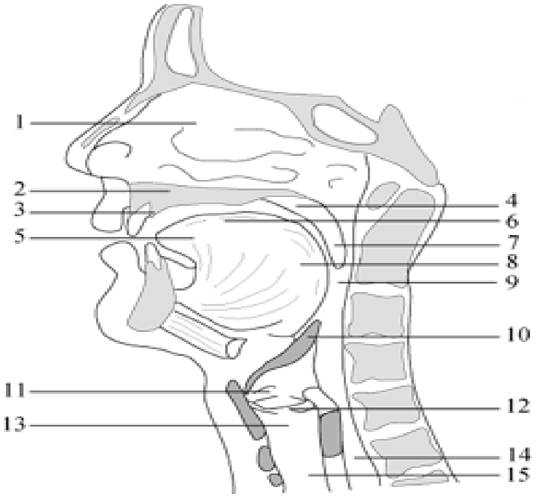

Figure 2.1: *Human vocal organs (1) Nasal cavity, (2) Hard palate, (3) Alveoral ridge, (4) Soft palate (Velum), (5) Tip of the tongue (Apex), (6) Dorsum, (7) Uvula, (8) Radix, (9) Pharynx, (10) Epiglottis, (11) False vocal cords, (12) Vocal cords, (13) Larynx, (14) Esophagus, and (15) Trachea*

and oral and nasal cavities. From the oral and nasal cavities the air flow exits through the mouth and nose respectively. The V-shaped opening between the vocal cords, called the glottis, is the most important sound source in the vocal system. Vocal cords may act in several different ways during speaking. The most important function is to modulate the air flow by rapidly opening and closing, causing buzzing sound from which vowels and voiced consonants are produced. The fundamental frequency of vibration depends on the mass and tension and is about 110 Hz, 200 Hz, and 300 Hz for men, women, and children, respectively.

2.2 Speech perception

Speech perception is the process by which the sounds of language are heard, interpreted and understood [21]. It is closely linked to the fields of phonetics and phonology in linguistics and cognitive psychology and perception in psychology. Usually speech perception is studied in the context of pattern recognition. Signals are presented to humans and they make some objective judgment about what they heard.

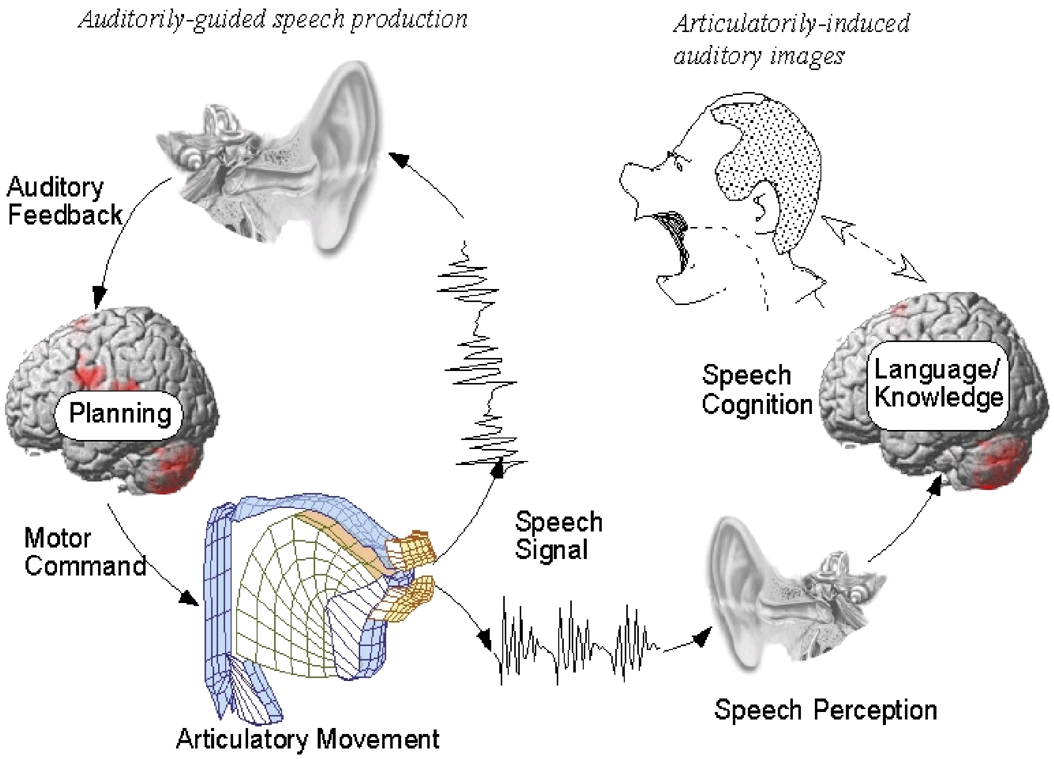

Figure 2.2: *Human speech production and perception process*

The process of perceiving speech begins at the level of the sound signal and the process of audition [11]. After processing the initial auditory signal, speech sounds are further processed to extract acoustic cues and phonetic information. This speech information can then be used for higher-level language processes, as in word recognition.

Most of the linguistic interactions occur within a face-to-face context, in which acoustic (speech) as well as visual information (mouth movements) are involved in message comprehension. Though humans are able to understand words without any visual input, audio-visual perception improves language comprehension, specially when the acoustic information is noisy. Integration of information on verbal messages provided by the two sensory (acoustical and visual) modalities could be based on similar patterns of time-varying features common to both the acoustical and the visual input. More specifically, the timing of changes in vocalization is visible as well as audible in terms of their time-varying patterns.

2.3 Units in audio-visual speech processing

As alphabets are the basic unit of any written language, in the same way there are alphabets defined for spoken language and visual speech. These are called phonemes and visemes [23] respectively.

2.3.1 Units in audio speech processing

A phoneme is the smallest segmental unit of sound employed to form meaningful contrasts between utterances. A phoneme is a group of slightly different sounds which are all perceived to have the same function by speakers of the language. Every language has a set of phonetic alphabet and a different set of possible phonemes and their combinations. In some languages, each letter in the spelling system represents one phoneme. However, in English language there is a poor match between spelling and phonemes and it has about 40 phonemes. The phonetic alphabet is usually divided in two main categories, vowels and consonants. Vowels are always voiced sounds and are produced with the vocal cords in vibration, while consonants may be either voiced or unvoiced. Vowels have considerably higher amplitude than consonants and they are also more stable and easier to analyze and describe acoustically. Because consonants involve very rapid changes, they are more difficult to synthesize properly.

2.3.2 Units in visual speech processing

As phonemes are basic units of audio speech, in the same way visemes are defined as basic units of visual speech [23]. A viseme is a representational unit used to classify speech sounds in the visual domain. This term was introduced based on the interpretation of the phoneme as a basic unit of speech in the acoustic domain. A viseme describes the particular facial and oral positions and movements that occur alongside the voicing of phonemes.

Phonemes and visemes do not always share a one-to-one correspondence. often, several phonemes share the same viseme. thirty two visemes are required in order to produce all possible phoneme with the human face. Visemes can help with understanding speech - if the phoneme is distorted or muffled, the viseme

Table 2.1: *Visemes as phoneme classes*

Viseme	Phoneme class
0	silence
1	f v w
2	s z
3	S Z
4	p b m
5	g k x n N r j
6	t d
7	l
8	I e:
9	E E:
10	A
11	@
12	i
13	O Y y u 2: o: 9 9: O:
14	a:

accompanying it can help to clarify what the sound actually was. Thus, visual and auditory components work together while communicating orally.

2.4 Multi modal integration by humans

Humans have five major senses: vision, hearing, touch, smell, and taste. The actual number of sensory modalities that an organism has is hard to find. Each sensory modality can be further divided into sub modalities. In vision, the processing of color, shape, orientation, motion, and depth may occur somewhat separately, and form somewhat independent sub modalities of vision.

Multi modal integration (multi sensory integration) is the study of how information from the different sensory modalities, such as sight, sound, touch, smell, self-motion and taste, may be integrated by the human nervous system. Such integration may result in unified perceptual experiences that are coherent across sensory modalities. Multi modal integration also deals with how different sensory modalities interact with one another and alter each other's processing.

In the case of speech perception, video plays a major role in understanding

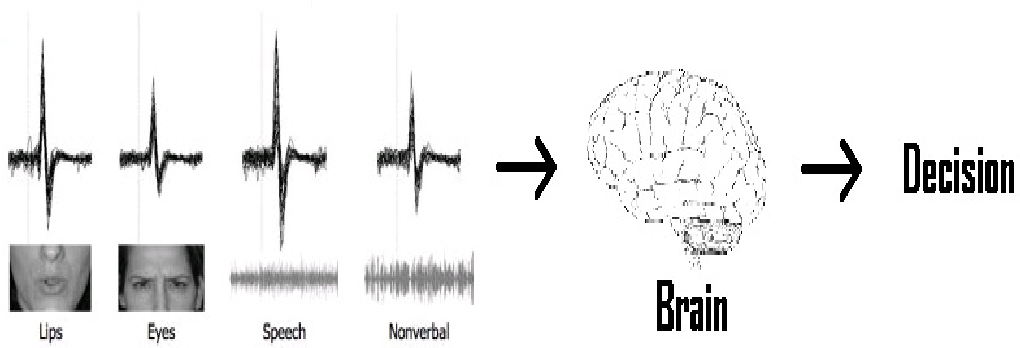

Figure 2.3: *Multi modal information fusion by humans*

what the speaker is speaking, specially when the communication is taking place in noisy environment. This is because of the fact that audio is highly susceptible to noise but video is not, so in noisy environments such as crowded areas, or in case of loud background noise, audio quality is too low, in such cases one can infer what the speaker is speaking by seeing the lip movement of the speaker.

3

Multi modal information fusion using a soft belief function

A multimedia content is composed of several streams that carry information in audio, video or textual channels. Classifying and clustering multimedia contents require extraction and combination of information from these different modalities [13]. The streams constituting a multimedia content are naturally different in terms of scale, dynamics and temporal patterns. These differences make combining the information sources using classic combination techniques difficult.

Depending on the type of information in a certain modality, different levels of information fusion [22] are defined. These are described as below:

- Sensor level fusion:

  Consolidation of raw data obtained from multiple sensors or multiple snaps from a single sensor.

- Feature level fusion:

  Consolidation of the feature sets originating from multiple modalities into a single feature set by normalization or transformation schemes.

- Score-level fusion:

  Consolidation of the score outputs from different modalities to generate a new score to be used for decision making.

- Rank-level fusion:

  Consolidates the rank output from different modalities to derive a consensus rank for each hypothesis.

- Decision-level fusion:

  Combining final recognition decisions of different modalities.

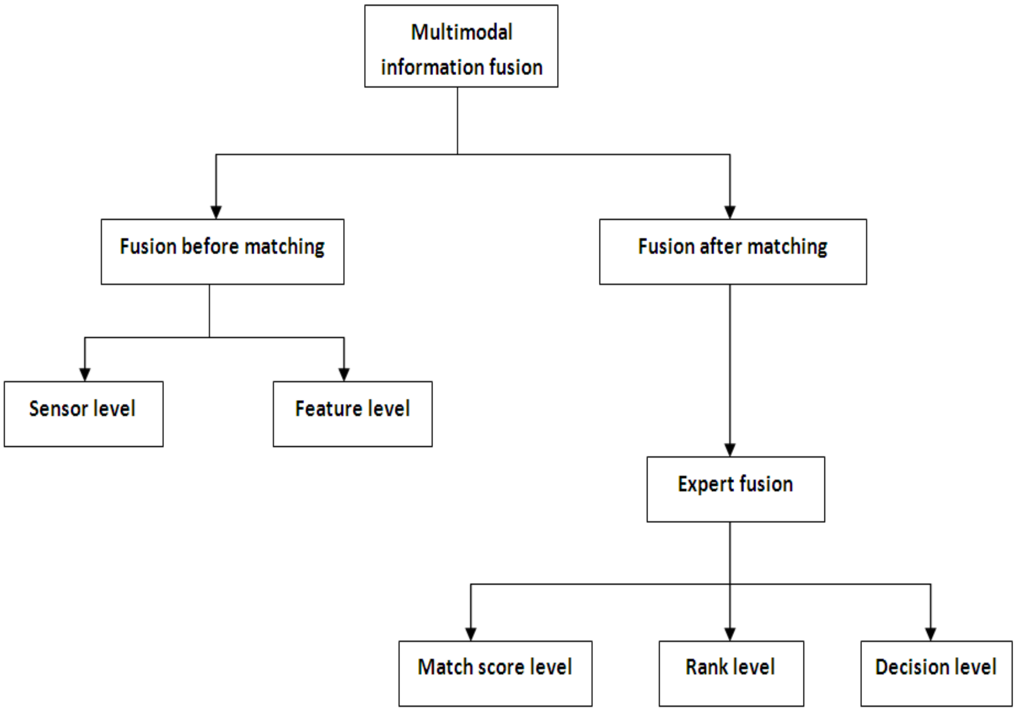

Figure 3.1: *Levels of multi modal information fusion*

Various methods have been proposed to fuse the multi modal information obtained from audio and video modalities, which are being described in Section 3.1.

3.1 Conventional information fusion schemes

A number of techniques are available for audio-visual information fusion, which can be broadly grouped into feature fusion and decision fusion methods. The first ones are the simplest, as they are based on training a traditional HMM classifier on the concatenated vector of the audio and visual features, or any appropriate transformation of it. Decision fusion methods combine the single-modality (audio-only and visual-only) HMM classifier outputs to recognize audio-visual speech. Specifically, class conditional log-likelihoods from the two classifiers are linearly combined using appropriate weights that capture the reliability of each classifier, or data stream. This likelihood recombination can occur at various levels of integration, such as the state, phone, syllable, word, or utterance level.

3.1.1 Concatenative feature fusion

In this information fusion scheme [17], feature vectors obtained from audio and video modalities are concatenated and the concatenated vector is used as a single feature vector.

Let us denote the time synchronous audio and visual feature vectors at instant t, by $O_S^{(t)} \in R^{D_s}$ of dimension D_s , where $s = A, V$ respectively. The joint audio-visual feature vector is then simply the concatenation of the two, namely

$$O^{(t)} = [O_A^{(t)^T}, O_V^{(t)^T}]^T \in R^D \tag{3.1}$$

where $D = D_A + D_V$

These feature vectors are then trained to generate HMMs as if generated from single modality and are used for the purpose of speech recognition.

3.1.2 Hierarchical fusion using feature space transformations

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

It makes a reasonable choice to consider Linear Discriminant Analysis (LDA) as a means of obtaining such a dimensionality reduction. Indeed, LDA achieves the best discrimination among the classes of interest on the basis of the data alone, without any a-priori bias in favor of any of the two feature streams. Similarly LDA is followed by an Maximum Likelihood Linear Transform (MLLT) based data rotation, in order to improve maximum likelihoodness of the data [17]. This method amounts to a hierarchical LDA / MLLT application on the original audio and visual features, as depicted in Figure 3.2, and it is therefore referred to as HiLDA (hierarchical LDA).

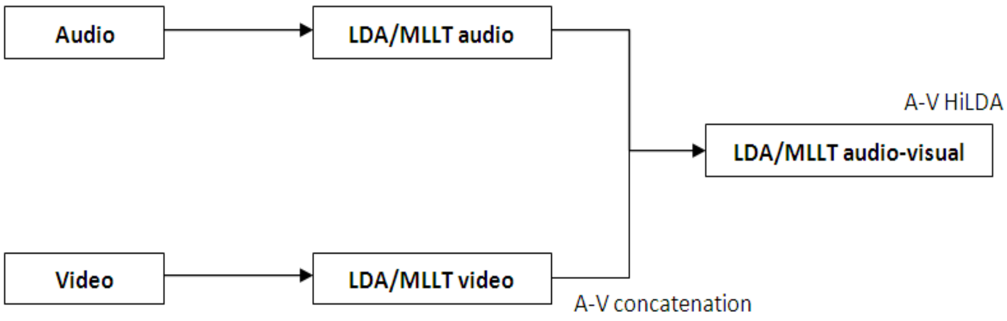

Figure 3.2: *Hierarchical LDA / MLLT feature extraction (AV-HiLDA)*

The final audio-visual feature vector is

$$O_{HiLDA}^{(t)} = P_{MLLT} \times P_{LDA} \times O^{(t)} \tag{3.2}$$

Where Matrices P_{LDA} and P_{MLLT} denote the LDA projection and MLLT rotation matrices.

3.1.3 Decision fusion with adaptive weighting scheme

Decision fusion approach is a good choice for designing a noise-robust AVSR system. Decision fusion in HMM-based AVSR systems is performed by utilizing the outputs of the acoustic and the visual HMMs for a given audiovisual speech datum and then fuse them adaptively to obtain noise-robustness over various noise environments [14].

When the acoustic and the visual features (O_A and O_V) of a given audio-visual speech datum of unknown class are obtained, the recognized utterance class C^*

is given by

$$C^{*} = arg\max_{i}\{\gamma \times logP(O_A|\lambda_A^i) + (1-\gamma) \times logP(O_V|\lambda_V^i)\} \tag{3.3}$$

where λ_A^i and λ_V^i are the acoustic and the visual HMMs for the i^{th} class, respectively, and $logP(O_A|\lambda_A^i)$ and $logP(O_V|\lambda_V^i)$ are their outputs (log-likelihoods). The integration weight γ determines how much the final decision relatively depends on each modality. It has a value between 0 and 1, and varies according to the amount of noise contained in the acoustic speech. When the acoustic speech is clean, the weight should be large because recognition with the clean acoustic speech usually outperforms that with the visual speech; on the other hand, when the acoustic speech contains much noise, the weight should be sufficiently small. Therefore, for noise-robust recognition performance over various noise conditions, it is important to automatically determine an appropriate value of the weight according to the noise condition of the given speech signal.

3.1.4 Coupled HMM based method for information fusion

A Coupled Hidden Markov Model (CHMM) is a generalization of the HMM suitable for a large variety of multimedia applications that integrate two or more streams of data [16]. A coupled HMM can be seen as a collection of HMMs, one for each data stream, where the discrete nodes at time t for each HMM are conditioned by the discrete nodes at time t-1 of all the related HMMs.

Parameters of a CHMM are defined as follows

$$\pi_o^c(i) = P(q_t^c = i) \tag{3.4}$$

$$b_t^c(i) = P(O_t^c|q_t^c = i) \tag{3.5}$$

$$a_{i|j,k}^c = P(q_t^c = i|q_{t-1}^0 = j, q_{t-1}^1 = k) \tag{3.6}$$

where q_c^t is the state of the couple node in the c^{th} stream at time t. In a continuous mixture with Gaussian components, the probabilities of the observed

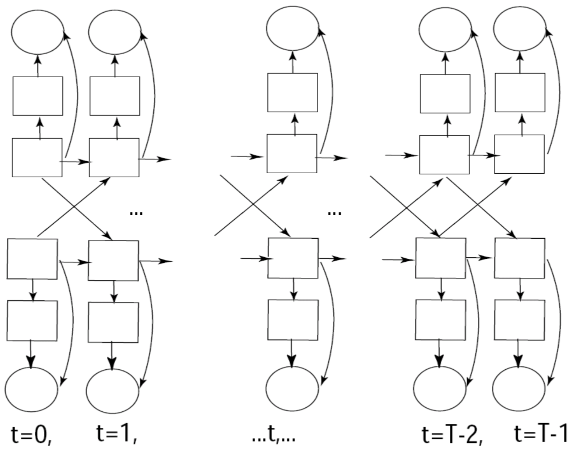

Figure 3.3: *The audio-visual coupled HMM*

nodes are given by:

$$b_t^c(i) = \sum_{m=1}^{M_i^c} w_{i,m}^c N(O_t^c, \mu_{i,m}^c, U_{i,m}^c) \tag{3.7}$$

where $\mu_{i,m}^c$ and $U_{i,m}^c$ are the mean and covariance matrix of the i^{th} state of a coupled node, and m^{th} component of the associated mixture node in the c^{th} channel. M_i^c is the number of mixtures corresponding to the i^{th} state of a coupled node in the c^{th} stream and the weight $w_{i,m}^c$ represents the conditional probability $P(s_t^c = m|q_t^c = i)$ where s_t^c is the component of the mixture node in the c^{th} stream at time t.

3.2 Soft fusion using Dempster-Schafer theory

Soft information fusion refers to a more relaxed and more flexible system to combine information from audio and video modalities for making better decision.

The Dempster Shafer (DS) theory is a mathematical theory of evidence [20]. It allows one to combine evidence from different sources and arrive at a degree of belief (represented by belief function) that takes into account all the available evidences. This theory was first developed by Arthur P. Dempster and Glenn Shafer.

Probability theory can effectively model uncertainty. However, there is some information probability cannot describe. For example, ignorance. Consider the following example: If we have absolutely no information about the coin, in probability theory, we will assume that it would be 50% head and 50% tail. However, in another scenario, if we know the coin is fair, so we know for a fact that it would be 50% head and 50% tail. Therefore, in the two different scenarios, we arrive at the same conclusion. So representing ignorance in probability theory becomes a problem. Dempster Shafer theory can effectively solve this problem. In Dempster Shafer Theory, for the ignorance scenario, the belief of Head and the belief of Tail would be 0, whereas for the fair coin scenario, the belief of Head would be 0.5, and that of Tail would also be 0.5.

DS theory is a generalization of the Bayesian theory of subjective probability. While the Bayesian theory requires probabilities for each question of interest, belief functions allow us to have degrees of belief for one question on probabilities of a related question.

Dempster Shafer (DS) theory of evidence, allows the representation and combination of different measures of evidence. Let

$$\Theta = (\theta_1, ..., \theta_k) \tag{3.8}$$

be a finite set of mutually exclusive and exhaustive hypothesis referred as singletons and Θ is referred as a frame of discernment. A basic probability assignment is a function m such that

$$m : 2^{\Theta} \rightarrow [0, 1] \tag{3.9}$$

where

$$\sum_{A \subset \Theta} m(A) = 1, \; m(\Phi) = 0 \tag{3.10}$$

If $\neg A$ is complementary set of A then by DS Theory

$$m(A) + m(\neg A) < 1 \tag{3.11}$$

Which is in contrast to probability theory. This divergence from probability is called “Ignorance”. The function assigning sum of masses of all the subsets of the set of interest is called the belief function and is given by

$$Bel(A) = \sum_{B \subset A} m(B) \tag{3.12}$$

Belief function assigned to each subset of θ is a measure of total belief in the preposition represented by the subset.

3.3 Decision making using soft belief functions

This section describes the proposed methodology of using the DS theory to combine decisions from the audio and video modalities. The degree of belief for a decision is determined from subjective probabilities obtained from the two modalities and then are combined using Dempster's rule making a reasonable assumption that the modalities are independent.

The DS theory defines a mass distribution based on the reliability of the individual modalities. Hence we first formulate a problem being looked upon using individual modalities and then fuse them using DS theory. Let

X_{audio} : The audio feature based decision.
X_{video} : The video feature based decision.

Then let us take a two hypothesis problem (H_1 or H_2) of two exclusive and exhaustive classes, which we are looking to classify with the help of above feature vectors.

Both X_{audio} and X_{video} can hypothesize as H_1 or H_2. Thus the focal elements of both the features are H_1, H_2 and Ω, where Ω is the whole set of classes $\{H_1, H_2\}$. The unimodal source reliabilities provide us with a certain degree of trust that we should have on the decision of that modality. The reliabilities of audio and video based decisions is based on the number of times the X_{audio} and X_{video} classifies the given data correctly. At a particular time interval, the audio and video features give a certain probability of classification. If P(X_{audio} = H_1) = p_1, then the mass distribution is $m_{audio}(H_1)$= xp_1. Similarly, the mass assigned to H_2 is $m_{audio}(H_2)$ = x(1-p_1). The remaining mass, is allocated to the whole set

Table 3.1: *Reliability of the unimodal Features*

Classification Feature	Reliability
Audio : X_{audio}	x
Video : X_{video}	y

of discernment, $m_{audio}(\Omega)$ = 1 - x. Similarly we assign a mass function for the video feature based decision. Since the audio and video feature based decisions are independent of each other, we can use Dempster's rule of combination for arriving at a joint decision. When the decisions of the two modalities are inconsistent, their product of masses is assigned to a single measure of inconsistency, say k. The confusion-matrix as obtained for the audio-video features combined is shown in Table 3.2.

Table 3.2: *The confusion-matrix of probability mass functions for multi modal features*

	$m_v(H_1)= yp_2$	$m_v(H_2) = y(1-p_2)$	$m_v(\Omega) = 1-y$
$m_a(H_1)=xp_1$	$m_{a,v}(H_1)=xyp_1p_2$	$k = xyp_1(1-p_2)$	$m_{a,v}(H_1)=x(1-y)p_1$
$m_a(H_2)=x(1-p_1)$	$k = xyp_2(1-p_1)$	$m_{a,v}(H_2)= xy(1-p_1)(1-p_2)$	$m_{a,v}(H_2)=x(1-y)(1-p_1)$
$m_a(\Omega)= 1-x$	$m_{a,v}(H_1)= (1-x)yp_2$	$m_{a,v}(H_2)=(1-x)y(1-p_2)$	$m_{a,v}(\Omega)=(1-x)(1-y)$

From Table 3.2, We define total inconsistency k as

$$k = xyp_1(1-p_2) + xyp_2(1-p_1) \tag{3.13}$$

Hence the combined belief in hypothesis H_1 and H_2, obtained from the multiple modalities (speech and video) can now be formulated as

$$Bel(H_1) = \frac{xyp_1p_2 + xp_1(1-y) + (1-x)yp_2}{(1-k)} \tag{3.14}$$

$$Bel(H_2) = \frac{xy(1-p_1)(1-p_2) + x(1-p_1)(1-y) + (1-x)y(1-p_2)}{(1-k)} \tag{3.15}$$

Note that the mass functions have been normalized by the factor (1-k). The Belief function hence computed in Equation 3.14 & 3.15, gives a soft decision measure for choosing a better hypothesis from the two possible classifications.

This two hypothesis problem can be extended to multiple hypothesis problem. Let us consider a multiple hypothesis problem having N possible hypothesis:

H_1: Hypothesis 1
H_2: Hypothesis 2
...
H_N: Hypothesis N

Recognition probabilities from individual modalities are given by Equation 3.16

$$P(X_{audio} = H_i) = A_i; P(X_{Video} = H_i) = V_i; 1 \leq i \leq N \tag{3.16}$$

The confusion-matrix for this N hypothesis problem is shown in Table 3.3.

Table 3.3: *The confusion-matrix of probability mass functions for multi modal features*

	$m_v(H_1)= yV_1$	$m_v(H_2) = yV_2$	...	$m_v(H_N)= yV_N$	$m_v(\Omega) = 1 - y$
$m_a(H_1)=xA_1$	$m_{a,v}(H_1)=xyA_1V_1$	$k = xyA_1V_2$	...	$k = xyA_1V_N$	$m_{a,v}(H_1)=x(1 - y)A_1$
$m_a(H_2)=xA_2$	$k = xyA_2V_1$	$m_{a,v}(H_2)= xyA_2V_2$	...	$k = xyA_2V_N$	$m_{a,v}(H_2)=x(1 - y)A_2$
...	...	...	...	...	...
$m_a(H_N)=xA_N$	$k = xyA_NV_1$	$k = xyA_NV_2$	...	$m_{a,v}(H_N)= xyA_NV_N$	$m_{a,v}(H_N)=x(1 - y)A_N$
$m_a(\Omega)= 1 - x$	$m_{a,v}(H_1)= (1 - x)yV_1$	$m_{a,v}(H_2)=(1 - x)yV_2$	...	$m_{a,v}(H_N)=(1 - x)yV_N$	$m_{a,v}(\Omega)=(1 - x)(1 - y)$

From Table 3.3, the total inconsistency k is given by

$$k = \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} xyA_iV_j \tag{3.17}$$

Hence the combined belief in hypothesis $H_k, 1 \leq k \leq N$, obtained from the multiple modalities (speech and video) can now be formulated as

$$Bel(H_k) = \frac{xyA_kV_k + x(1 - y)A_k + (1 - x)yV_k}{(1 - k)} \tag{3.18}$$

4

Speaker diarization using a soft belief function

“Speaker Diarization” is the process of partitioning an input audio stream into homogeneous segments according to the speaker identity. It can enhance the readability of an automatic speech transcription by structuring the audio stream into speaker turns and, when used together with Speaker recognition systems, by providing the speaker’s true identity. It is used to answer the question “who spoke when?”. Speaker Diarization is a combination of “Speaker Segmentation” and “Speaker Clustering”. It first aims at finding speaker change points in an audio stream, and then grouping together speech segments on the basis of speaker characteristics.

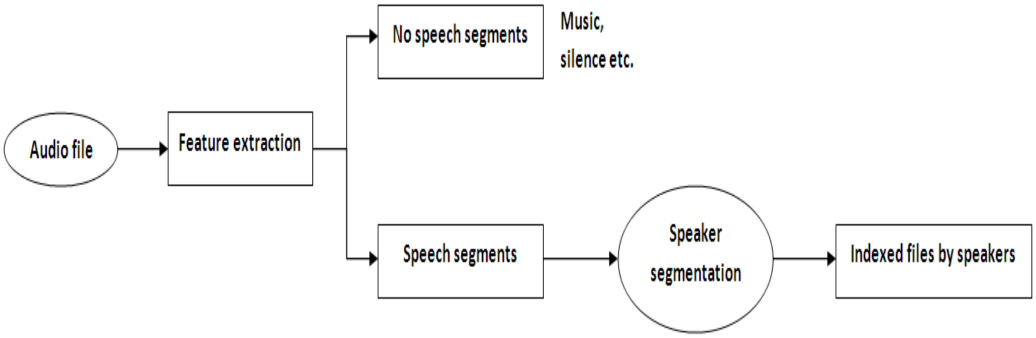

Figure 4.1: *Block diagram of an audio based speaker diarization system*

4.1 Audio based speaker diarization

This section describes speaker diarization being achieved using audio information only. This section would include basics of audio processing, audio feature vector extraction and then application of speaker diarization.

4.1.1 Speech feature extraction

Mel Frequency Cepstral Coefficients (MFCC), are most commonly used feature vectors for speech processing.

In speech processing, Mel Frequency Cepstrum(MFC) represents short term power spectrum of the sound derived from the linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. This frequency warping allows for better representation of sound.

The mel scale which was proposed by Stevens, Volkman and Newman in 1937 is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The reference point between this scale and normal frequency measurement is defined by equating a 1000 Hz tone, 40 dB above the listener's threshold, with a pitch of 1000 mels. Above about 500 Hz, larger and larger intervals are judged by listeners to produce equal pitch increments. As a result, four octaves on the hertz scale above 500 Hz are judged to comprise about two octaves on the mel scale.

conversion of the frequency between mel and Hertz is done using the formula

$$\mathrm{m} = 2595 \times \log_{10}(\frac{\mathrm{f}}{700} + 1) \tag{4.1}$$

$$\mathrm{f} = 700 \times (10^{\mathrm{m}/2595} - 1) \tag{4.2}$$

A cepstrum is the result of taking the Fourier transform (FT) of the decibel spectrum as if it was a signal. the power cepstrum (of a signal) is the squared magnitude of the Fourier transform of the logarithm of the squared magnitude of the Fourier transform of the signal.

$$\text{power cepstrum of signal} = |\text{F}\{\log(|\text{F}\{\text{signal}\}|^2)\}|^2 \tag{4.3}$$

procedure to extract MFCC's from an audio signal is as follows:

- Take the Fourier transform of the windowed excerpt of the audio signal.
- Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
- Take the log of power at each of the mel frequencies.
- Take the discrete cosine transform of the list of mel log powers considering them to be a signal.
- The MFCCs are the amplitudes of the resulting spectrum.

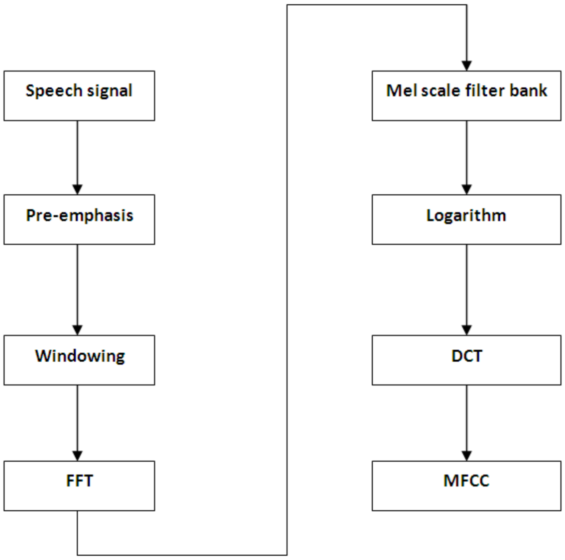

Figure 4.2: *Block diagram describing MFCC calculation from speech signal*

MFCCs are commonly used as feature vectors in speech and speaker recognition systems, which can automatically recognize numbers spoken into a telephone and can recognizing people from their voice. MFCCs are also increasingly finding use in music information retrieval applications such as genre classification and audio similarity measures.

One of the shortcomings of MFCCs is that they are not very robust in the presence of additive noise, and so it is common to normalize their values in speech recognition systems to nullify the influence of noise.

4.1.2 Traditional audio based speaker diarization methods

Speaker diarization can be achieved by speaker clustering [2]. This problem is formulated in an ergodic HMM framework with minimum duration constraint.

If $X = \{x_1, x_2, ..., x_T\}$ is the audio data to be segmented, we want to find the optimal number of clusters K_s^* and there respective Gaussian mixture models (GMM) $\lambda_{K_s^*}$, that produces the best segmentation of the data X according to

$$(\lambda_{K_s^*}, K_s^*) = arg \max_{\lambda_{K_s}, K_s} P(X, q_{best} | \lambda_{K_s}, K_s) \tag{4.4}$$

where q_{best} is the Viterbi path with the highest likelihood. There is one state q for each speaker cluster. Thus, we want to find the set of clusters and their acoustic models that maximize the likelihood of the data, as well as the associated speaker segmentation based on this HMM topology.

The algorithm starts with over-clustering the data, i.e. clustering the data in terms of more than the expected number of classes (large initial value for K_s). This is followed by an agglomerative clustering approach where best candidate clusters are merged in an iterative fashion.

One more approach for speaker diarization in a meeting room scenario using audio information only is to determine direction of arrival using a multichannel microphone array [18]. The signals coming from the different microphones are Wiener filtered to improve the SNR. Then, one of the signals (microphones), the one with the highest SNR, is selected as a reference channel. The Time Delay Of

Arrival (TDOA) between each of the other channels and the reference channel is calculated. In order to estimate the TDOA between segments corresponding to two microphones, a modified version of the Generalized Cross Correlation with phase transform ($GCC_{PHAT}(f)$) is used.

4.1.3 Bayesian information criteria and its application to unimodal speaker diarization

4.1.3.1 Bayesian information criteria(BIC)

Bayesian Information Criterion (BIC) also known as Schwarz criterion is a criterion for model selection among a class of parametric models with different numbers of parameters. Choosing a model to optimize BIC is a form of regularization. When estimating model parameters using maximum likelihood estimation, it is possible to increase the likelihood by adding additional parameters, which may result in over fitting. The BIC resolves this problem by introducing a penalty term for the number of parameters in the model.

The BIC was developed by Gideon E. Schwarz, who gave a Bayesian argument for adopting it. It is very closely related to the Akaike Information Criterion (AIC).

BIC value is given by the formula:

$$\mathrm{BIC}(\mathcal{M}_i) = log(\mathcal{L}(\mathcal{X}_i, \mathcal{M}_i)) - \lambda \frac{1}{2} \#(\mathcal{M}_i) log(N_i) \tag{4.5}$$

where:

- $\mathcal{L}(\mathcal{X}_i, \mathcal{M}_i)$ = likelihood of the data.
- N_i = number of frames in the acoustic segment under consideration.
- $\#(\mathcal{M}_i)$ = number of free parameters to estimate corresponding to the model $\mathcal{M}_i$.
- λ is a free parameter dependent on the data being modeled.

Given any two estimated models, the model with the higher value of BIC is the one to be preferred. Higher BIC implies either fewer explanatory variables, better fit, or both. It is important to note that the BIC can be used to compare estimated models only when the numerical values of the dependent variable are identical for all estimates being compared.

Few important characteristics of BIC are as follows:

- It does not depend on the prior.
- It can measure the efficiency of the parameterized model in terms of predicting the data.
- It has a penalty term for the complexity of the model where complexity refers to the number of parameters in model.
- It is equal to the minimum description length criterion but has a negative sign.
- It can be used to choose the number of clusters according to the intrinsic complexity present in a particular dataset.

BIC is been widely used for model identification in time series and linear regression. It can be applied quite widely to any set of maximum likelihood-based models. However, it should be noted that in many applications (for example, selecting a black body or power law spectrum for an astronomical source), BIC simply reduces to maximum likelihood selection because the number of parameters is equal for the models of interest.

4.1.3.2 Speech based unimodal speaker diarization

The BIC (Bayesian Information Criterion) for segmentation and clustering based on MOG (Mixture Of Gaussian) is used for the purpose of speech based unimodal speaker diarization. The likelihood distance is calculated between two segments to determine whether they belong to the same speaker or not. The distances used for acoustic change detection can also be applied to speaker clustering in order to infer whether two clusters belong to the same speaker. For a given acoustic segment $\mathcal{X}_i$, the BIC value of a particular model $\mathcal{M}_i$, indicates how well the model fits the data, and is determined by Equation 4.5

In order to detect the audio scene change between two segments with the help of BIC, one can define two hypothesis.

Hypothesis 0 is defined as:

$$\mathrm{H}_0 : \mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_N \sim \mathcal{N}(\mu, \Sigma) \tag{4.6}$$

which considers the whole sequence to consist no speaker change.

Hypothesis 1 is defined as:

$$\mathrm{H}_1 : \mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_L \sim \mathcal{N}(\mu_1, \Sigma_1); \mathrm{x}_{L+1}, \mathrm{x}_{L+2}, ..., \mathrm{x}_N \sim N(\mu_2, \Sigma_2) \tag{4.7}$$

is the hypothesis that a speaker change occurs at time L.

A check of whether the hypothesis H_0 better models the data as compared to the hypothesis H_1, for a mixture of Gaussian case can be done by computing a function similar to the generalized likelihood ratio as

$$\begin{aligned} \Delta\mathrm{BIC}(\mathcal{M}_i) &= \log(\mathcal{L}(\mathcal{X}, \mathcal{M})) - (\log(\mathcal{L}(\mathcal{X}_i, \mathcal{M}_i)) \\ &+ \log(\mathcal{L}(\mathcal{X}_j, \mathcal{M}_j))) - \lambda\Delta\#(i, j)\log(\mathrm{N}) \end{aligned} \tag{4.8}$$

where $\Delta\#(i, j)$ is the difference in the number of free parameters between the combined and the individual models.

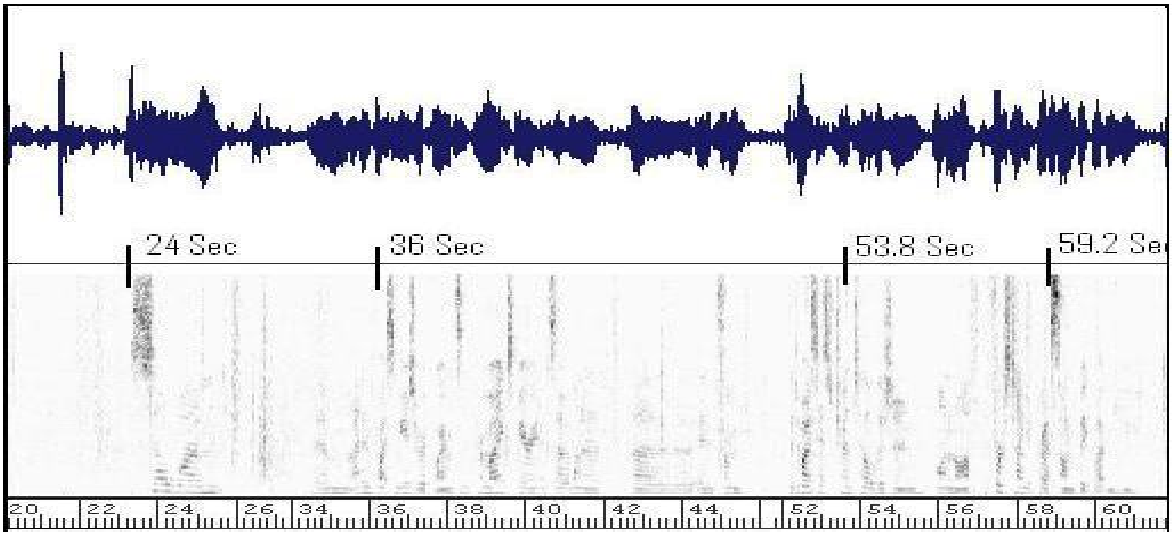

Figure 4.3: *Speech based unimodal speaker change detection*

When the BIC value based on mixture of Gaussian model exceeds a certain threshold, an audio scene change is declared. Note that the Mel Frequency Cepstral Coefficients (MFCC) are used as feature vectors in all the experiments.

Figure 4.3 illustrates a sample speaker change detection plot with speech information only using BIC. The illustration corresponds to the data from the AMI multi modal corpus. Speaker changes have been detected at 24, 36, 53.8 and 59.2 seconds.

4.2 Modeling visual speech activity using HMMs

Visual speech activity can be effectively modeled using Hidden Markov Models (HMMs).

4.2.1 Basics of HMM

A hidden Markov model (HMM) is a statistical model in which the process being modeled is assumed to be a Markov process with unobserved state. An HMM can be considered as the simplest dynamic Bayesian network [19].

A Markov process, named after the Russian mathematician Andrey Markov, is a time-varying random phenomenon for which the conditional probability distribution of future states of the process depend only upon the present state; that is, given the present, the future does not depend on the past.

The HMM is a powerful statistical tool for modeling generative sequences that can be characterized by an underlying process generating an observable sequence. HMMs have found application in many areas including signal processing, handwriting recognition, gesture recognition and in particular speech processing. These are also been applied with success to low level NLP tasks such as part-of-speech tagging, phrase chunking, and extracting target information from documents.

Markov models are of two types, regular Markov models and hidden Markov models. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. Figure 4.4 describes an example of a Markov process.

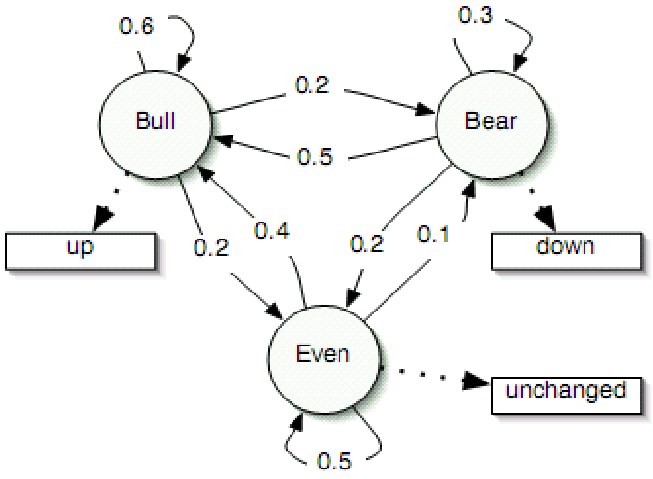

Figure 4.4: *Example of a regular Markov process*

This model describes a simple model for a stock market index. It has three states, Bull, Bear and Even, and three index observations up, down, unchanged. The model is a finite state one, with probabilistic transitions between states. Given a sequence of observations, one can easily find out the state sequence that had produced those observations. For example, for the observation sequence of up-down-unchanged, the state sequence should be: Bull-Bear-Even, and the probability of the sequence is simply the product of the transitions, in this case 0.2×0.2.

In a hidden Markov model, the state is not directly visible, but output dependent on the state is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Here the adjective “hidden” refers to the state sequence through which the model passes, not to the parameters of the model. Figure 4.5 describes how we can extend the previous model to HMM case.

This new model now allows all observation symbols to be emitted from each state with a finite probability. This change makes the model much more expressive. In this case, a bull market would have both good days and bad days, but there would be more good ones. The key difference is that now if we have

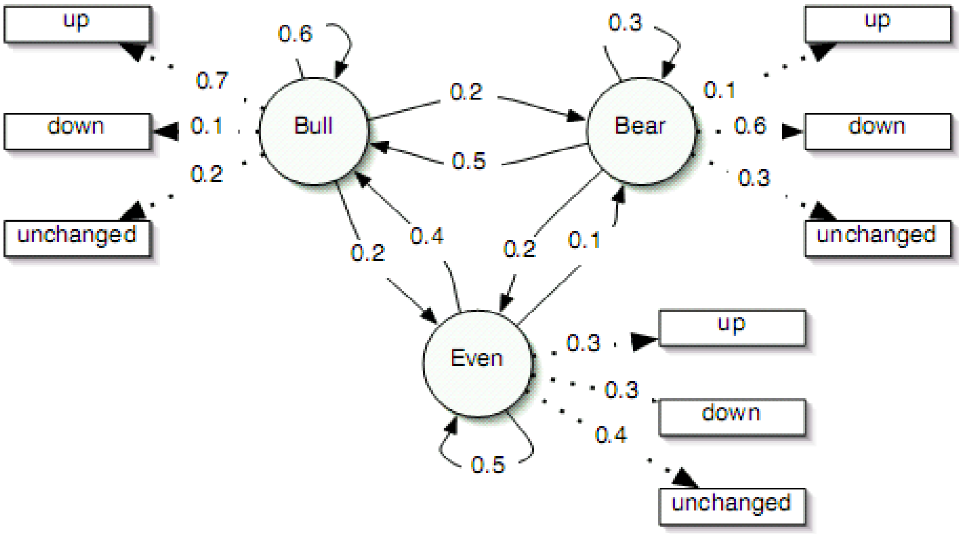

Figure 4.5: *Example of a hidden Markov process*

the observation sequence up-down-down then we cannot say exactly which state sequence may have produced these observations and thus the state sequence is hidden. We can however calculate the probability of the model producing this sequence, as well as which state sequence was most likely to have produced the observation.

4.2.2 Definitions and notations of a HMM

The formal definition of an HMM is as follows

$$\lambda = (A, B, \pi) \tag{4.9}$$

S represents state alphabet set, having a total of N states and V represents observation alphabet set, having a total of M observations

$$S = (s_1, s_2, ..., s_N) \tag{4.10}$$

$$V = (v_1, v_2, ..., v_M) \tag{4.11}$$

We define Q as a fixed state sequence of length T, and corresponding observation sequence be O

$$Q = (q_1, q_2, ..., q_T) \tag{4.12}$$

$$O = (o_1, o_2, ..., o_T) \tag{4.13}$$

A represents a transition array, storing the probability of transition from one state to another.The state transition probabilities are independent of time

$$A = [a_{ij}],\ a_{ij} = P(q_t = s_j | q_{t-1} = s_i) \tag{4.14}$$

$$a_{ij} \geq 0 \tag{4.15}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{4.16}$$

B represents observation array, storing the probability of observation k being produced from the state i, which is independent of time t

$$B = [b_i(k)],\ b_i(k) = P(o_t = v_k | q_t = s_i) \tag{4.17}$$

The initial probability array is represented by π

$$\pi = [\pi_i],\ \pi_i = P(q_1 = s_i) \tag{4.18}$$

$$\sum_{i=1}^{N} \pi_i = 1 \tag{4.19}$$

There are two assumptions being made by the model. The first one, called the Markov assumption, states that the current state is dependent only on the previous state, this represents the memory property of the model

$$P(q_t | q_{t-1}, q_{t-2}, ..., q_1) = P(q_t | q_{t-1}) \tag{4.20}$$

The other one, independence assumption states that the output observation at time t depends only on the current state, and is independent of previous observations and states

$$P(o_t|o_{t-1}, o_{t-2}, ..., o_1, q_t, q_{t-1}, q_{t-2}, ..., q_1) = P(o_t|q_t) \tag{4.21}$$

4.2.3 Three basic problems of HMM

For the HMMs to be useful in the real world, three basic problems needed to be solved, these are:

1. Evaluation problem: Given an observation sequence $O = o_1, o_2, ..., o_T$ and a model $\lambda = (A, B, \pi)$, how to effectively compute $P(O|\lambda)$, the probability to observe the sequence for the given HMM.

2. Decoding problem: Given the observation sequence $O = o_1, o_2, ..., o_T$ and the model $\lambda = (A, B, \pi)$, how to choose a corresponding state sequence $Q = q_1, q_2, ..., q_T$, which is optimal in some meaningful sense.

3. Learning problem: How to adjust the model parameters $\lambda = (A, B, \pi)$, to maximize $P(O|\lambda)$.

4.2.3.1 The evaluation problem

Given an HMM, and a sequence of observations, we would like to compute P(O|λ), the probability of occurrence of observation sequence given the model. This problem could be viewed as that of evaluating how well a model predicts a given observation sequence, and thus allow us to choose the most appropriate model from a set.

The probability of observing the observation sequence O for a specific state sequence Q is

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(o_t|q_t, \lambda) = b_{q_1}(o_1) \times b_{q_2}(o_2) \times ... \times b_{q_T}(o_T) \tag{4.22}$$

And the state sequence probability is

$$P(Q|\lambda) = \pi_{q_1} \times a_{q_1q_2} \times a_{q_2q_3} \times ... \times a_{q_{T-1}q_T} \tag{4.23}$$

So the probability of observation given the model is

$$P(O|\lambda) = \sum_{Q} P(O|Q, \lambda)P(Q|\lambda) = \sum_{q_1 \cdots q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1q_2} b_{q_2}(o_2)...a_{q_{T-1}q_T} b_{q_T}(o_T) \tag{4.24}$$

This expression allows us to evaluate the probability of occurrence of O given λ, but evaluating it this way is computationally expensive, of the order of exponential in T ($2TN^T$).

A better approach is to realize that many redundant calculations would be made while directly evaluating above equation, and therefore caching calculations can lead to reduced complexity. We implement the cache as a trellis of states at each time step, calculating the cached values (called α) for each state as a sum over all states at the previous time step. α is the probability of the partial observation sequence $o_1, o_2, ..., o_t$ and state s_i at time t. We define the forward probability variable as

$$\alpha_t(i) = P(o_1 o_2 ... o_t, q_t = s_i|\lambda) \tag{4.25}$$

The algorithm for this process is called the forward algorithm and is as follows

1. Initialization
$$\alpha_1(i) = \pi_i b_i(o_1),\ 1 \leq i \leq N \tag{4.26}$$

2. Induction
$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(o_{t+1}),\ 1 \leq t \leq T-1,\ 1 \leq j \leq N \tag{4.27}$$

3. Termination
$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{4.28}$$

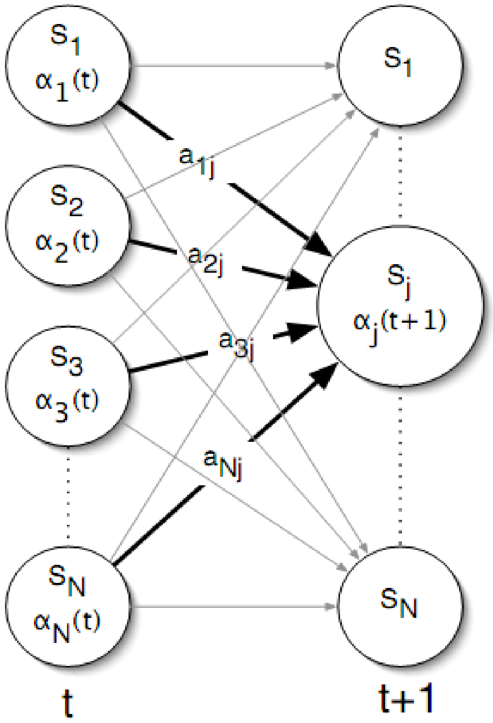

Figure 4.6: *Induction step for forward algorithm*

The induction step is the key to the forward algorithm and is depicted in figure 4.6. For each state s_j , $\alpha_j(t)$ stores the probability of arriving at that state having observed the observation sequence until time t.

The forward algorithm reduces the complexity of calculations involved to N^2T from $2TN^T$, which is the complexity for direct calculation.

In a similar manner we can define a backward variable $\beta_t(i)$ as the probability of partial observation sequence from t+1 to the end, given state s_i at time t and the model λ.

$$\beta_t(i) = P(o_{t+1}o_{t+2}...o_T|q_t = s_i, \lambda) \tag{4.29}$$

We can solve for $\beta_t(i)$ as follows

1. Initialization

$$\beta_T(i) = 1,\, 1 \leq i \leq N \tag{4.30}$$

2. Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j),\ t = T-1, T-2, ..., 1,\ 1 \leq i \leq N \qquad (4.31)$$

Computation of $\beta_t(i)$, $1 \leq t \leq T$, $1 \leq i \leq N$, requires calculations of the order of N^2T.

4.2.3.2 The decoding problem

The aim of decoding is to discover the hidden state sequence that is most likely to have produced a given observation sequence. One solution to this problem is to use the Viterbi algorithm to find the single best state sequence for an observation sequence. The Viterbi algorithm is another trellis algorithm which is very similar to the forward algorithm, except that the transition probabilities are maximized at each step, instead of being summed.

For the solution of the decoding problem we define a variable $\gamma_t(i)$, as the probability of being in state i at time t, given the observation sequence O and model λ.

$$\gamma_t(i) = P(q_t = s_i | O, \lambda) \qquad (4.32)$$

γ can be expressed in terms of forward backward variables as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \qquad (4.33)$$

Using $\gamma_t(i)$, individually most likely state q_t can be found out using

$$q_t = \max_{1 \leq i \leq N} [\gamma_t(i)] \qquad (4.34)$$

For Viterbi decoding we define a variable $\delta_t(i)$ as the probability of the most probable state path for the partial observation sequence

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P(q_1, q_2, ..., q_{t-1}, q_t = s_i, o_1, o_2, ..., o_t | \lambda) \qquad (4.35)$$

The Viterbi algorithm is as follows

1. Initialization

$$\delta_1(i) = \pi_i b_i(o_1),\, 1 \leq i \leq N,\, \psi_1(i) = 0 \tag{4.36}$$

2. Recursion

$$\delta_t(j) = \max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}]b_j(o_t),\, 2 \leq t \leq T,\, 1 \leq j \leq N \tag{4.37}$$

$$\psi_t(j) = arg \max_{1\leq i\leq N}[\delta_{t-1}(i)a_{ij}],\, 2 \leq t \leq T,\, 1 \leq j \leq N \tag{4.38}$$

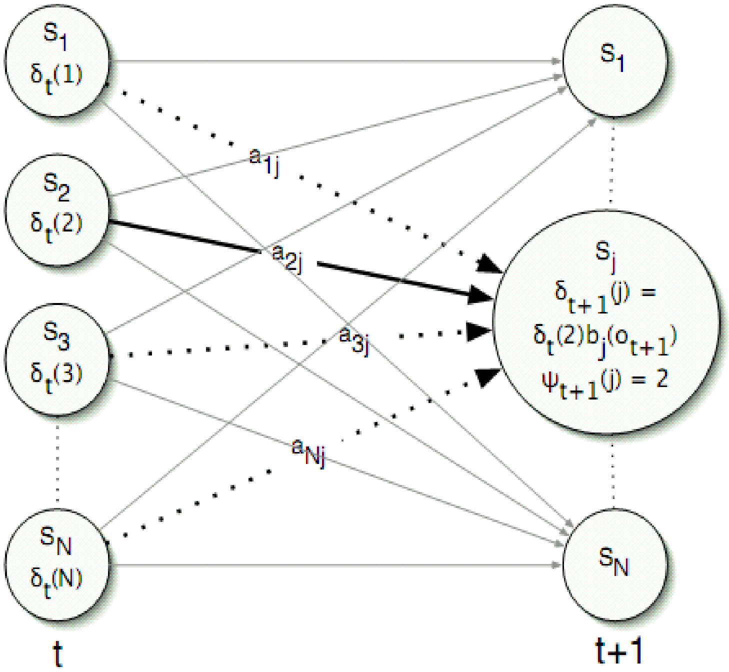

Figure 4.7: *Recursion step of the Viterbi algorithm*

3. Termination

$$P^* = \max_{1\leq i\leq N}[\delta_t(i)] \tag{4.39}$$

$$q_T^* = arg \max_{1 \leq i \leq N} [\delta_t(i)] \tag{4.40}$$

4. Optimal state sequence backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*),\, t = T-1, T-2, ..., 1 \tag{4.41}$$

4.2.3.3 The learning problem

Given a set of examples from a process, we would like to be able to estimate the model parameters $\lambda = (A, B, \pi)$ that best describe that process. There are two standard approaches to this task, dependent on the form of the examples, which are referred as supervised and unsupervised training. If the training examples contain both the inputs and outputs of a process, we can perform supervised training by equating inputs to observations, and outputs to states, but if only the inputs are provided in the training data then we must used unsupervised training to guess a model that may have produced those observations.

For any finite observation sequence as training data, there is no optimal way to estimate the model parameters, but we can choose $\lambda = (A, B, \pi)$ such that $P(O|\lambda)$ is locally maximized.

Baum-Welch method is one such iterative method to estimate model parameters. For this we define a variable $\xi_t(i, j)$, which gives the probability of being in state s_i at time t and s_j at time $t+1$, given the model and the observation sequence.

$$\begin{aligned}
\xi_t(i,j) &= P(q_t = s_i, q_{t+1} = s_j | O, \lambda) && (4.42)\\
&= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} && (4.43)\\
&= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} && (4.44)
\end{aligned}$$

$\gamma_t(i)$ can be expressed in terms of $\xi_t(i,j)$ as

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{4.45}$$

Adding $\gamma_t(i)$ over time variable t gives the expected number of times the state s_i is visited. Similarly adding $\xi_t(i,j)$ over t gives the expected number of transitions from state s_i to s_j.

Using above concepts, set of reasonable re estimation formula for π, A and B are

$$\bar{\pi}_i = \text{Expected frequency in state } s_i \text{ at time } (t=1) = \gamma_1(i) \tag{4.46}$$

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{Expected number of transitions from state } s_i \text{ to state } s_j}{\text{Expected number of transitions from state } s_i} && (4.47) \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} && (4.48) \end{aligned}$$

$$\begin{aligned} \bar{b}_j(k) &= \frac{\text{Expected number of times in state j, observing symbol } v_k}{\text{Expected number of times in state j}} && (4.49) \\ &= \frac{\sum_{t=1, s.t. v_t = o_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} && (4.50) \end{aligned}$$

4.2.4 Video based speaker diarization

This section describes how video only information can be used to determine speaker turn and group them together to realize the application of speaker diarization.

4.2.4.1 Traditional video based speaker diarization methods

One approach for speaker diarization using video information is done by agglomerative clustering of video features using GMMs [10]. To provide video features for speaker diarization, frame-based visual activity features, that have been shown to correlate well with speaking activity patterns are used. For each frame, the average motion vector magnitude over estimated skin blocks is calculated and used as a measure of individual visual activity for a camera view. Note that the averaging over estimated skin blocks is done to reduce the effect of background clutter and mitigate pose and scale variations. To detect skin blocks, a block-level skin-color detector, working mostly in the compressed domain is used. A GMM is used to model the distribution of (U,V) chrominance coefficients of skin-tone in the YUV color space where each Gaussian component is assumed to have a diagonal covariance matrix. In the Intra-frames, the likelihood of observed chrominance DCT coefficients are computed according to the GMM and threshold it to determine skin-color blocks. Skin blocks in the Inter-frames are inferred by using motion vector information to propagate skin-color blocks through the duration of the Group-Of-Picture (GOP). These values from all camera views are concatenated and used as the video feature vector for that frame.

Another method to achieve speaker diarization using video information is through detecting speaker locations [12]. The probability of speaker location is estimated with a face location detection and tracking algorithm called Sparse Template Condensation Tracker (STC Tracker), which is built into the recording system. When the tracker is initialized it can automatically build 3-D face templates, and it is robust as regards large head rotations of up to 60 degrees in the horizontal direction. In addition, it can track multiple faces simultaneously in real time by particle filtering with a GPU.

4.2.4.2 Face tracking

Face tracking extends face detection to video sequences. Any individual who appears in a video for any length of time generates a face track that is, a sequence of face instances across time. The goal of face tracking is to aggregate single-frame detection results into a collection of face tracks. In face tracking we exploit spatio-temporal continuity to associate face instances across frames, and iteratively update motion models for all face tracks, respecting shot boundaries in the process. It can serve as a front end to further analysis modules, such as face recognition, face expression analysis, gaze tracking and lip-reading, to name a few. Face tracking is also a core component to enable the computer to “see” the user in a Human-Computer Interface system.

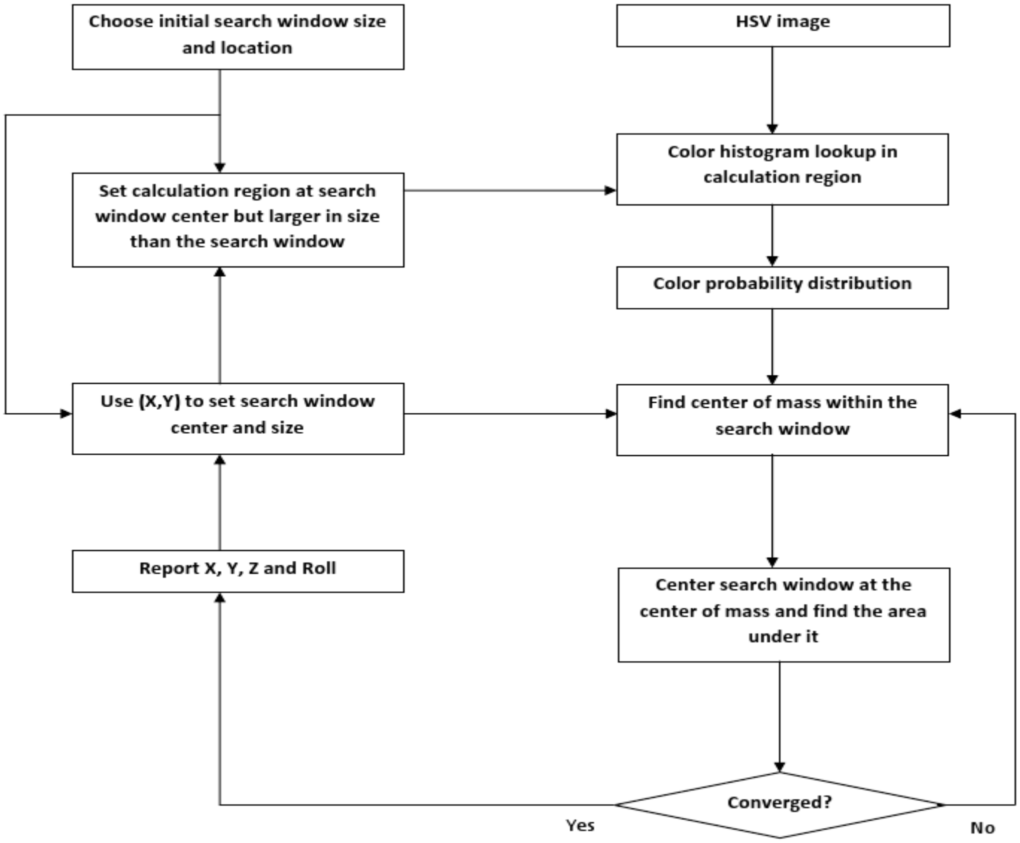
Choose initial search window size and location
HSV image
Set calculation region at search window center but larger in size than the search window
Color histogram lookup in calculation region
Color probability distribution
Use (X,Y) to set search window center and size
Find center of mass within the search window
Report X, Y, Z and Roll
Center search window at the center of mass and find the area under it
Converged?
Yes
No

Figure 4.8: *Block diagram for face tracking application*

4.2.4.3 Face tracking using CAMSHIFT algorithm

Camshift stands for “Continuously Adaptive Mean Shift”. This is the basis for the face-tracking algorithm in OpenCV. It combines the basic Mean Shift algorithm with an adaptive region-sizing step [6]. The kernel is a simple step function applied to a skin-probability map. The skin probability of each image pixel is based on color and is calculated using a method called histogram back projection. Color is represented as hue from the HSV color model. CAMSHIFT works by tracking the hue of an object, in this case, flesh color. The movie frames are all converted to HSV space before individual analysis. It's a robust, non-parametric, iterative technique for finding the mode of probability distributions. It is also computationally less expensive so can be used for tracking in real time applications.

The CAMSHIFT algorithm is based on the MEAN SHIFT algorithm. The MEAN SHIFT algorithm works well on static probability distributions but not on dynamic ones as in a movie. It is able to handle dynamic distributions by readjusting the search window size for the next frame based on the zeroth moment of the current frames distribution. This allows the algorithm to anticipate object movement to quickly track the object in the next scene. Even during quick movements of an object, CAMSHIFT is able to track it correctly.

The algorithm to implement CAMSHIFT is as follows:

1. Initial location of the 2D search window is computed.

2. The color probability distribution is calculated for a region slightly bigger than the mean shift search window.

3. Mean shift is performed on the area until suitable convergence. The zeroth moment and centroid coordinates are computed and stored.

4. The search window for the next frame is centered around the centroid and the size is scaled by a function of the zeroth movement.

5. Go to step 2.

It is to be noted that the initial search window is specified manually.

4.2.4.4 Feature extraction from video

Video features used for the purpose of speaker diarization are frame based video features. The features used is the histogram of the hue plane of the face pixels.

The face of the speaker is extracted from the video using CAMSHIFT algorithm. Then we determine the hue plane of the face region of each frame. The histogram of this hue plane in 32 bins is being used as video feature vector. To determine whether a segment belong to speaking or non-speaking, we train Hidden Markov Model(HMM) of speaking/non-speaking features.

4.2.4.5 Video based unimodal speaker diarization using HMMs

Unimodal speaker diarization based on video features uses frame based video features for speaker diarization. The features used is the histogram of the hue plane of the face pixels. Hue plane features of the whole face are used and not just of the lips. This is primarily because the face contains a considerable amount of information from the perspective of changes in the hue plane. It was also noted from initial experiments that the changes in the hue plane of the face pixels when a person is speaking compared to when silent are significant. This histogram is then used as feature vector for training models.

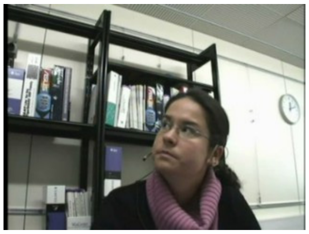

Figure 4.9: *Video frame of silent speaker*

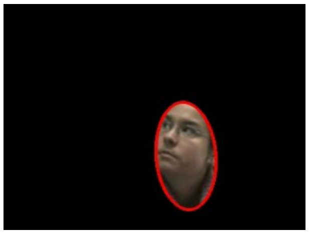

Figure 4.10: *Face extraction*

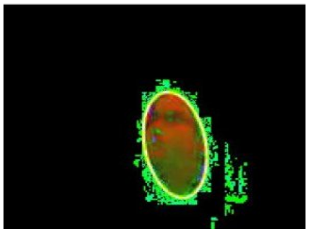

Figure 4.11: *Hue plane of the face*

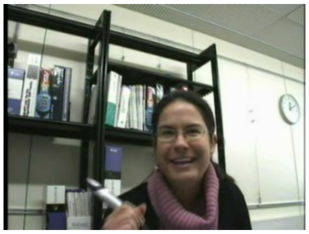

Figure 4.12: *Video frame of a talking speaker*

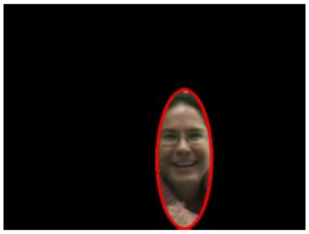

Figure 4.13: *Face extraction*

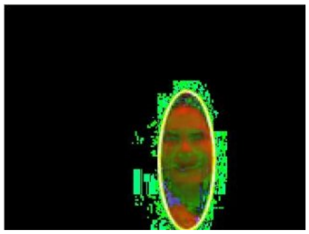

Figure 4.14: *Hue plane of the face*

Figures 4.9, 4.10, 4.11, shows a frame of the video of a silent speaker from the AMI database, whose skin colored pixels are tracked and then the hue plane of the frame extracted. In Figures 4.12, 4.13, 4.14, a similar set of results are illustrated for the same speaker and from the same video clip, when she is speaking.

Using the features extracted from the histogram of the hue plane, speaker diarization is now performed over a video segment of a certain duration by calculating the likelihood of the segment belonging to a model. The segment is classified as belonging to that speaker, for which the model likelihood is maximum. HMMs for each speaker are trained a priori using the video features. A speaker change is detected if the consecutive segments are classified as belonging to different models. The probability of speaker change is computed as the probability of two consecutive video segments belonging to two different models.

4.3 Speaker diarization using soft fusion of audio visual information

The soft fusion of audio visual information using DS theory is done as described in Section 3.3.

Here this is a two hypothesis problem being described as:

H_1: Speaker change detected
H_2: No speaker change detected

These hypothesis are to be tested using audio and video based decisions, X_{audio} and X_{video} respectively. Using Table 3.2, we can find out belief functions for the two hypothesis as

$$Bel(H_1) = \frac{xyp_1p_2 + xp_1(1-y) + (1-x)yp_2}{(1-k)} \tag{4.51}$$

$$Bel(H_2) = \frac{xy(1-p_1)(1-p_2) + x(1-p_1)(1-y) + (1-x)y(1-p_2)}{(1-k)} \tag{4.52}$$

The Belief function hence computed in Equation 4.51 and 4.52, gives a soft decision measure for speaker diarization.

5

Audio visual speech recognition using soft fusion methods

Audio visual speech recognition (AVSR) is a technique that uses image processing capabilities in lip reading to aid speech recognition systems in recognizing indeterministic phones or giving preponderance among near probability decisions. Ability to organize and intelligently combine sensory data derived from multiple sensors (modulated by perceptual relevance and sensory confidence) is crucial for building a robust model of objects and events in our environment, in spite of dramatically varying perceptual conditions. Audio visual speech recognition exploits human perceptual principle of sensory integration (Joint use of audio and visual information) to improve the recognition of human activity (e.g. speech recognition, speech activity, speaker change, etc.), intent (e.g. speech intent) and identity (e.g: speaker recognition), particularly in the presence of acoustic degradation due to noise and channel, and the analysis and mining of multimedia content.

The applications for this work include accurate transcription of human activity for improved human-computer interfaces, multimedia content mining and meeting transcription. Many applications of AVSR lies in the field of health care, defense, telephone based applications, and for people with disabilities.

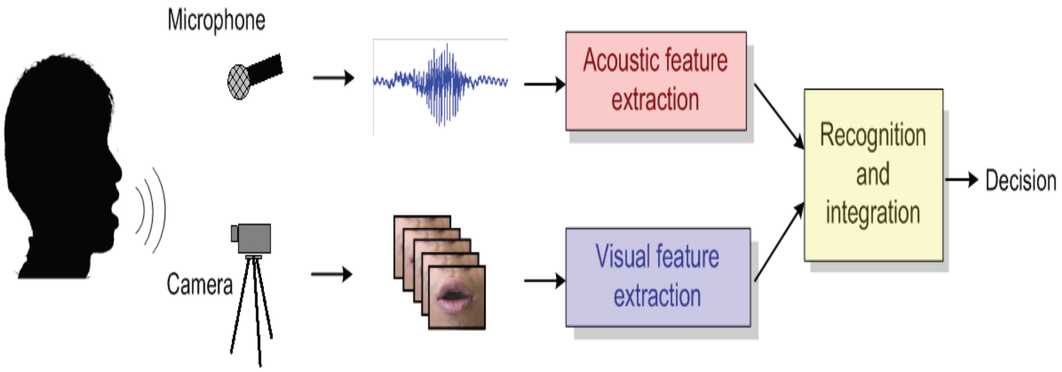
Microphone
Camera
Acoustic feature extraction
Visual feature extraction
Recognition and integration
Decision

Figure 5.1: *General procedure of an audio-visual speech recognition system*

5.1 Audio based speech recognition

Speech recognition (also known as automatic speech recognition(ASR)) converts spoken words to text. Speech recognition applications include voice dialing, call routing, remote appliance control and content-based spoken audio search, simple data entry (e.g., entering a credit card number), preparation of structured documents, speech-to-text processing (e.g., word processors or emails), and in aircraft cockpits (usually termed Direct Voice Input).

5.1.1 Types of automatic speech recognition systems

Speech recognition systems can be characterized by many parameters, some of which are shown in Table 5.1

Table 5.1: *Typical parameters used to characterize the capability of speech recognition systems*

Parameters	Range
Speaking modes	Isolated words to continuous speech
Speaking style	Read speech to spontaneous speech
Enrollment	Speaker dependent to speaker independent
Vocabulary	Small(<20 words) to large(>20000 words)
Language model	Finite state to context sensitive
Perplexity	Small(<10) to large(>100)
SNR	High(>30db) to low(<10db)
Transducer	Noise canceling microphone to telephone

Speech recognition is a difficult problem, because of the many sources of variability associated with the signal, such as the acoustic realizations of phonemes, the smallest sound units of which words are composed, are highly dependent on the context in which they appear.

5.1.2 Performance of speech recognition systems

The performance of speech recognition systems is usually specified in terms of accuracy and speed. Accuracy may be measured in terms of performance accuracy which is usually rated with word error rate (WER), whereas speed is measured with the real time factor.

5.1.2.1 HMM based speech recognition

Modern general-purpose speech recognition systems are generally based on Hidden Markov Models. These are statistical models which outputs a sequence of symbols or quantities. One possible reason why HMMs are used in speech recognition is that a speech signal could be viewed as a piecewise stationary signal or a short-time stationary signal. Speech could be approximated as a stationary process and could thus be thought of as a Markov process.

Another reason why HMMs are popular is because they can be trained automatically and are simple to use and computationally feasible. In speech recognition, the hidden Markov model would output a sequence of n-dimensional real-valued vectors (with n being a small integer, such as 10), outputting one of these every 10 milliseconds. The vectors would consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short time window of speech and de-correlating the spectrum using a cosine transform, then taking the first few (most significant) coefficients. The hidden Markov model will tend to have in each state a statistical distribution that is a mixture of diagonal covariance Gaussians which will give a likelihood for each observed vector. Each word, or (for more general speech recognition systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes.

Decoding of speech uses the Viterbi algorithm to find the best path, and here there is a choice between dynamically creating a combined hidden Markov model which includes both the acoustic and language model information, or combining it statically beforehand (the finite state transducer, or FST, approach).

5.1.2.2 Dynamic time warping (DTW) based speech recognition

Dynamic time warping is an approach that was historically used for speech recognition but has now largely been displaced by the more successful HMM-based approach. Dynamic time warping is an algorithm for measuring similarity between two sequences which may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another they were walking more quickly, or even if there were accelerations and decelerations during the course of one observation. DTW has been applied to video, audio, and graphics. Indeed, any data which can be turned into a linear representation can be analyzed with DTW.

A well known application has been automatic speech recognition, to cope with different speaking speeds. In general, it is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions, i.e. the sequences are "warped" non-linearly to match each other. This sequence alignment method is often used in the context of hidden Markov models.

5.1.3 Feature vector for audio modality

Speech recognition experiment, being performed on 33 speakers of GRID corpus, with each speaker having an utterance of 1000 sentences. This is being done for clean speech as well as speech corrupted with Additive White Gaussian Noise(AWGN) of SNR 10dB, 20dB and 30dB.

13 dimensional MFCC features of audio signal is used as audio feature vector. Filter bank parameters used for the calculation of MFCC's are as follows

- Lowest frequency = 133.3333 Hz
- Linear filters = 13

- Linear spacing = 66.66666666 Hz
- Log filters = 27
- Log spacing = 1.0711703 Hz
- FFT size = 512
- Cepstral coefficients = 13
- Window size = 256

HMM models for these MFCC vectors of each word utterance are trained for audio only speech recognition.

Figure 5.2, 5.3, 5.4 and 5.5 shows audio signal and corresponding MFCC feature vector contour for the utterance of the sentence “bin blue at f one soon”, for clean speech and speech corrupted with AWGN having SNR of 30dB, 20dB and 10dB respectively.

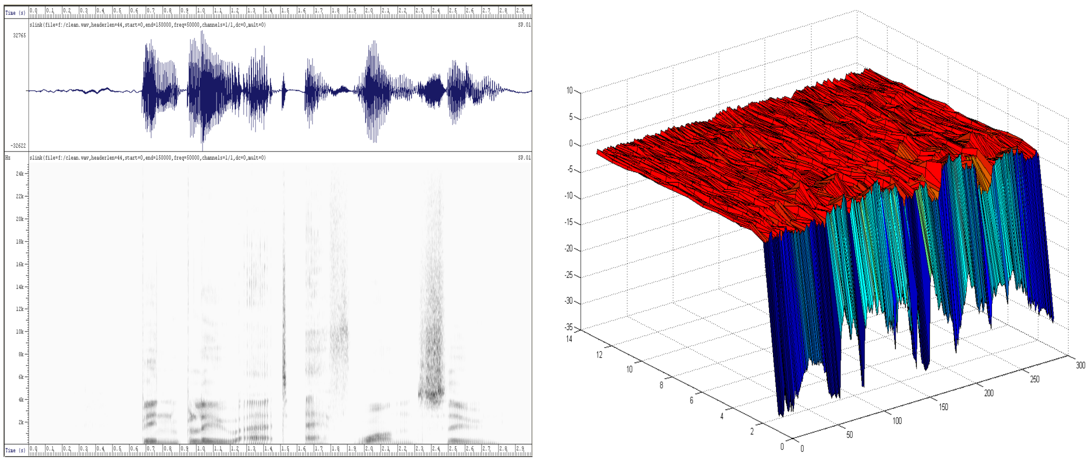

Figure 5.2: *Clean speech signal & its feature vector*

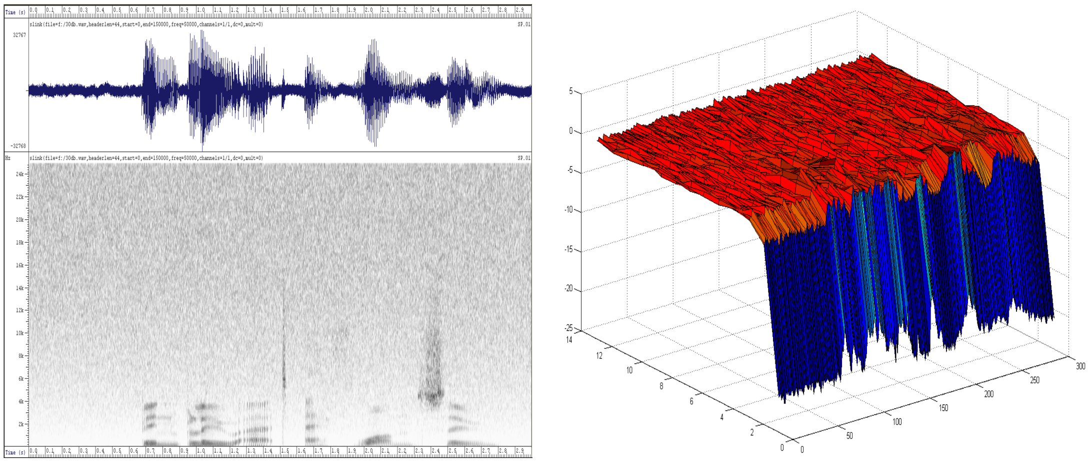

Figure 5.3: *Speech signal corrupted with AWGN having SNR of 30 dB & its feature vector*

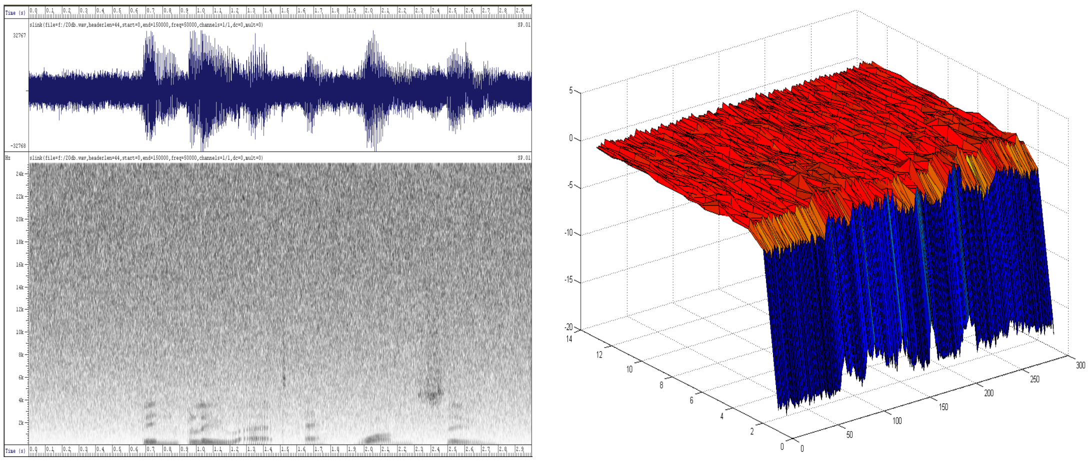

Figure 5.4: *Speech signal corrupted with AWGN having SNR of 20 dB & its feature vector*

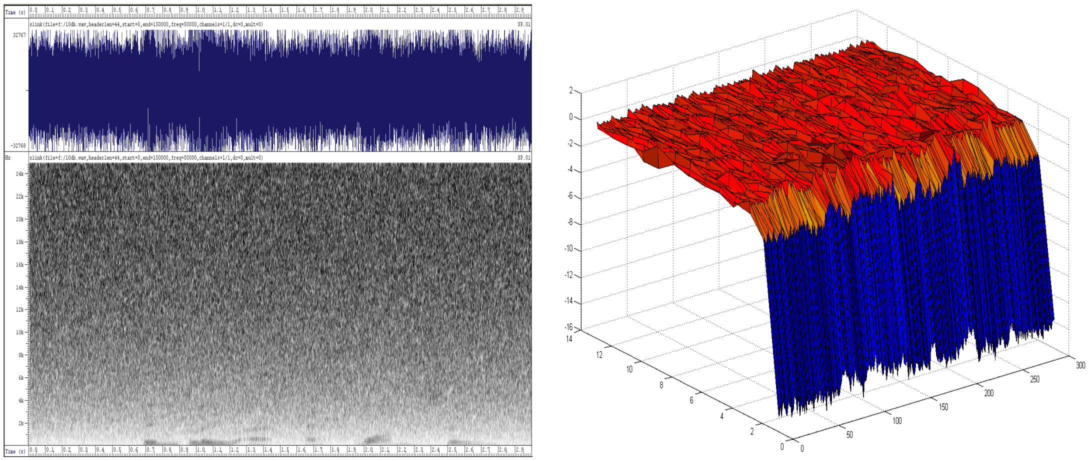

Figure 5.5: *Speech signal corrupted with AWGN having SNR of 10 dB & its feature vector*

5.2 Video based speech recognition

5.2.1 Visual speech

Visible Speech is a writing system invented in 1867 by Alexander Melville Bell, father of Alexander Graham Bell, the inventor of the telephone. Melville Bell was a teacher of the deaf and intended his writing system to help deaf students learn spoken language. Visible Speech was also the first notation system for the sounds of speech independent of a particular language or dialect and was widely used to teach students how to speak with a "standard" accent. Visible Speech symbols are intended to provide visual representation of the positions of speech organs needed to articulate individual sounds. Visible Speech is also known as the Physiological Alphabet.

Synchronous presentation of stimuli to the auditory and visual systems can modify the formation of a percept in either modality. For example, perception of auditory speech is improved when the speaker's facial articulatory movements are visible. Neural convergence onto multi sensory sites exhibiting supra-additivity has been proposed as the principal mechanism for integration.

5.2.2 Visual information as a supplement to audio information

The potential for improved speech recognition rates using visual features is well established in the literature on the basis of psychophysical experiments. Canonical mouth shapes that accompany speech utterances have been categorized, and are known as visual phonemes or "visemes". Visemes provide information that complements the phonetic stream from the point of view of confusability.

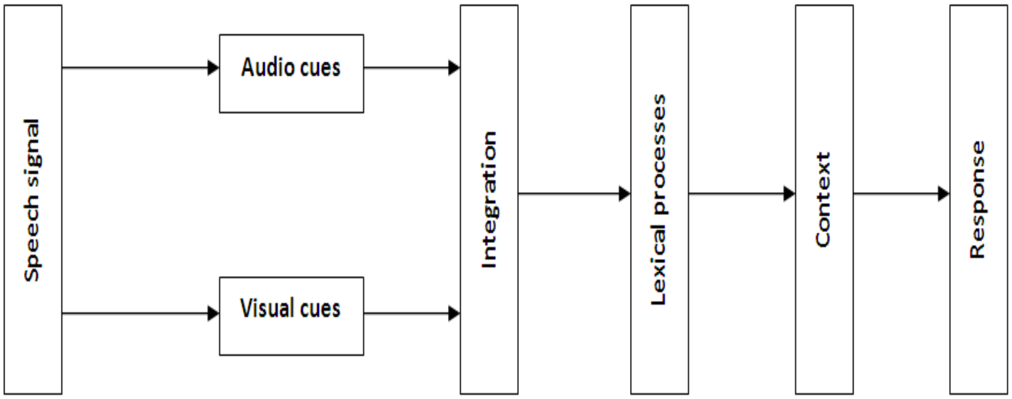

Figure 5.6: *Audio-visual speech recognition system schematic*

Candidate visual features are grayscale parameters of the mouth region; geometric model-based parameters such as area, height, width of mouth region; lip contours arrived at by curve fitting, spline parameters of inner/outer contour; and motion parameters obtained by 3D tracking. Gray scale parameters suffer from being sensitive to lighting conditions. Lip contour information, although invariant to lighting conditions, may not provide enough information of the inner articulators such as teeth and tongue. In general they can be grouped into three classes of features, High level lip contour based features, low level video pixel based one and a combination of these two.

5.2.3 Feature vector for video modality

We have used histogram of the lip region as video feature. To find out the video feature, lip region is assumed in the lower half of the face part. We have used 70×110 pixel sized region, in the lower part of the face as lip region. To find out video feature vector, fist we subtract RGB values of consecutive frames, so as to get motion vector video from the original video. Lip region is then extracted from this video and is converted to gray scale image by adding up the RGB values. A non-linear scale histogram of the pixel values of each frame, in 16 bins is found out and is used as feature vector. The 16 bin non linear scale used to find out histogram is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9-10, 11-12, 13-15, 16-19, 20-29, 30-49 & ≥ 50.

HMM models for these video features of each word utterance are trained for video only speech recognition.

Figure 5.7 shows lip region of a speaker when he is speaking the sentence “bin blue at f one soon.”

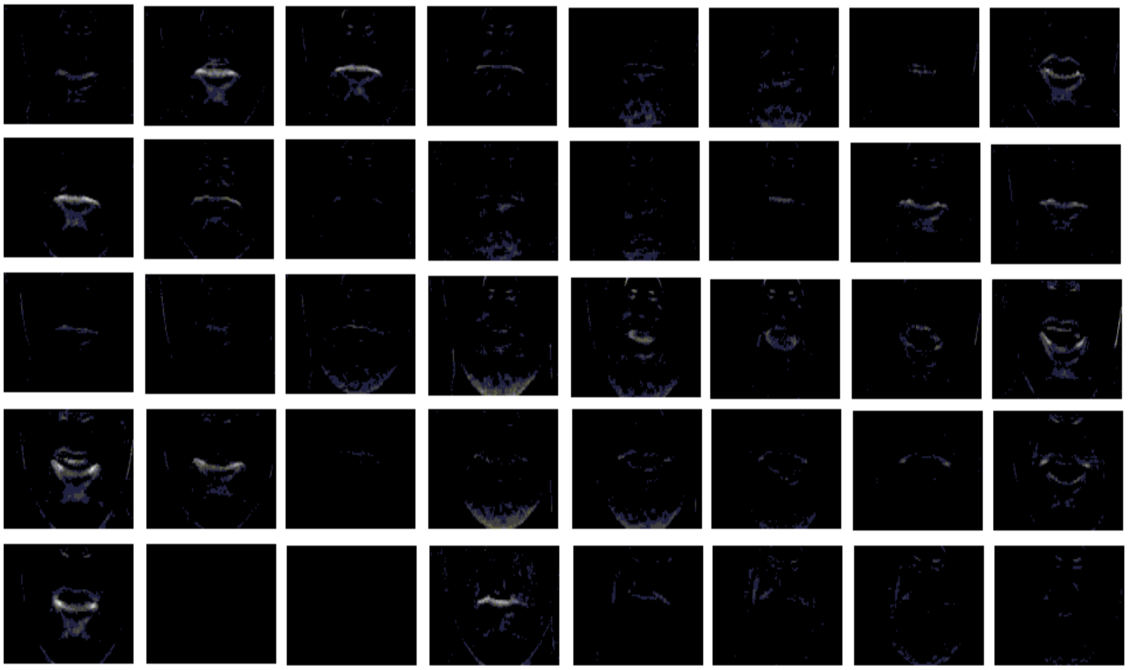

Figure 5.7: *Lip region features*

5.3 Audio visual information fusion

The primary challenge in AVSR is to obtain the performance which is equal to or better than the performance of any modality for various noise conditions. When the noise level is low, the acoustic modality performs better than the visual one and, thus, the audio-visual recognition performance should be at least as good as that of the acoustic speech recognition. When the noise level is high and the visual recognition performance is better than the acoustic one, the integrated recognition performance should be at least same as or better than the performance of the visual-only recognition. Besides, we expect the synergy effect of the two modalities by using AVSR systems. Thus, the goal of the second challenge in the use of audio-visual information for speech recognition is to improve the recognition performance with as high synergy of the modalities as possible.

Combining the two modalities should take full advantage of the individual modalities so that the integrated system shows a high synergy effect for a wide range of noise conditions. On the contrary, when the fusion is not performed appropriately, we cannot expect complementarity and synergy of the two information sources and, moreover, the integrated recognition performance may be even inferior to that of any of the unimodal systems, which is called “attenuating fusion” or “catastrophic fusion” [14].

5.3.1 Audio visual information fusion using DS theory

Speech recognition is a many hypothesis problem, depending on the number of words in the dictionary. To use DS theory to combine the decisions made from audio and video modalities, we first find out the reliability of audio and video data by carrying out recognition experiments on the training data itself and find our the percentage recognition. this gives us the reliability of audio and video data.

The scaled audio and video probabilities are found out using the formula

$$\text{probability} = 10^{\frac{\text{log probability}}{1000 \times \text{scale}}} \tag{5.1}$$

In Equation 5.1, the variable “scale” represents the weight being given to video modality, while making the combined decision. Recognition probability (log probability) that we obtain from recognizer is quite less, for audio it is of the order of -300 to -200 whereas for video is of the order of -3000 to -2000. Because of exponential function, for large values of scale, difference in the probabilities of different words is lager for video than audio. so large value of variable “scale” represents more weight being given to video than audio.

Audio visual speech recognition is an N hypothesis problem, where each utterance has N possible options to be classified into.

H_1: word 1
H_2: word 2
...
H_N: word N

Recognition probabilities from individual modalities are given by Equation 5.2

$$P(X_{audio} = H_i) = A_i; P(X_{Video} = H_i) = V_i; 1 \leq i \leq N \tag{5.2}$$

We want to find out most likely hypothesis by using X_{Audio} and X_{Video}, where

X_{audio} : The audio feature based decision.
X_{video} : The video feature based decision.

The reliability of audio and video based decision is as given in Table 5.2

Table 5.2: *Reliability of the unimodal features*

Classification Feature	Reliability
Audio : X_{audio}	x
Video : X_{video}	y

Then, as being described in Section 3.3, The confusion-matrix for this N hypothesis problem is shown in Table 5.3.

Table 5.3: *The confusion-matrix of probability mass functions for multi modal features*

	$m_v(H_1)= yV_1$	$m_v(H_2) = yV_2$	...	$m_v(H_N)= yV_N$	$m_v(\Omega) = 1 - y$
$m_a(H_1)=xA_1$	$m_{a,v}(H_1)=xyA_1V_1$	$k = xyA_1V_2$	...	$k = xyA_1V_N$	$m_{a,v}(H_1)=x(1-y)A_1$
$m_a(H_2)=xA_2$	$k = xyA_2V_1$	$m_{a,v}(H_2)= xyA_2V_2$	...	$k = xyA_2V_N$	$m_{a,v}(H_2)=x(1-y)A_2$
...	...	...	...	...	...
$m_a(H_N)=xA_N$	$k = xyA_NV_1$	$k = xyA_NV_2$	...	$m_{a,v}(H_N)= xyA_NV_N$	$m_{a,v}(H_N)=x(1-y)A_N$
$m_a(\Omega)= 1 - x$	$m_{a,v}(H_1)= (1-x)yV_1$	$m_{a,v}(H_2)=(1-x)yV_2$	...	$m_{a,v}(H_N)=(1-x)yV_N$	$m_{a,v}(\Omega)=(1-x)(1-y)$

From Table 5.3, the total inconsistency k is given by

$$k = \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{j=1}^{N} xyA_iV_j \tag{5.3}$$

Hence the combined belief in hypothesis $H_k, 1 \leq k \leq N$, obtained from the multiple modalities (speech and video) can now be formulated as

$$Bel(H_k) = \frac{xyA_kV_k + x(1-y)A_k + (1-x)yV_k}{(1-k)} \tag{5.4}$$

The Belief function hence computed in Equation 5.4, gives a soft decision measure for choosing a better hypothesis from the possible options.

6

Databases

6.1 Multi modal data acquisition test Bed

The experimental test bed at MiPS lab, IIT Kanpur is a typical meeting room setup which can accommodate four participants around a table. It is equipped with an eight channel linear microphone array and a four channel video array, capable of recording each modality synchronously.

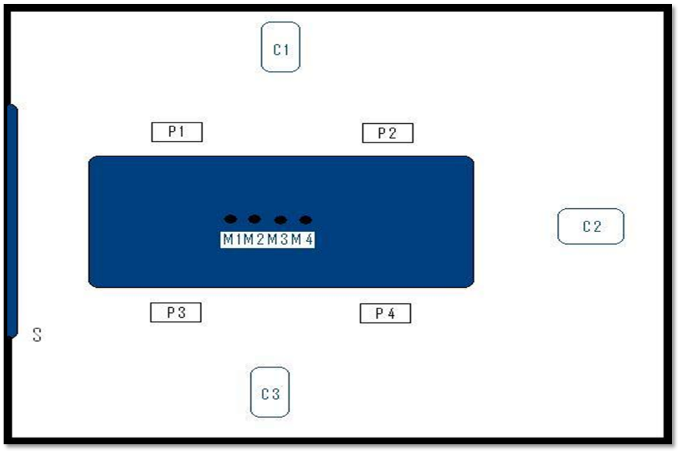

Figure 6.1: *Layout of the test bed at MiPS lab, IIT Kanpur*

Figure 6.1 represents layout of the test bed. Here C1, C2, C3 are three cameras; P1, P2, P3, P4 are four participants of the meeting ; M1, M2, M3, M4 represents four mics and S is the screen.

A manual timing pulse is generated to achieve multi modal synchronization. For the purpose of speaker diarization we use only one channel of audio data and 3 channel of video data with each camera focusing on the participants face. The multi modal data used in our experiments is eighteen minutes long, consisting of 3 speakers taking turns as in a dialogue, and the discussion was centered around varied topics like soccer, research, and mathematics. Figure 6.2 shows the snapshot of the test bed used for generating the multi modal data

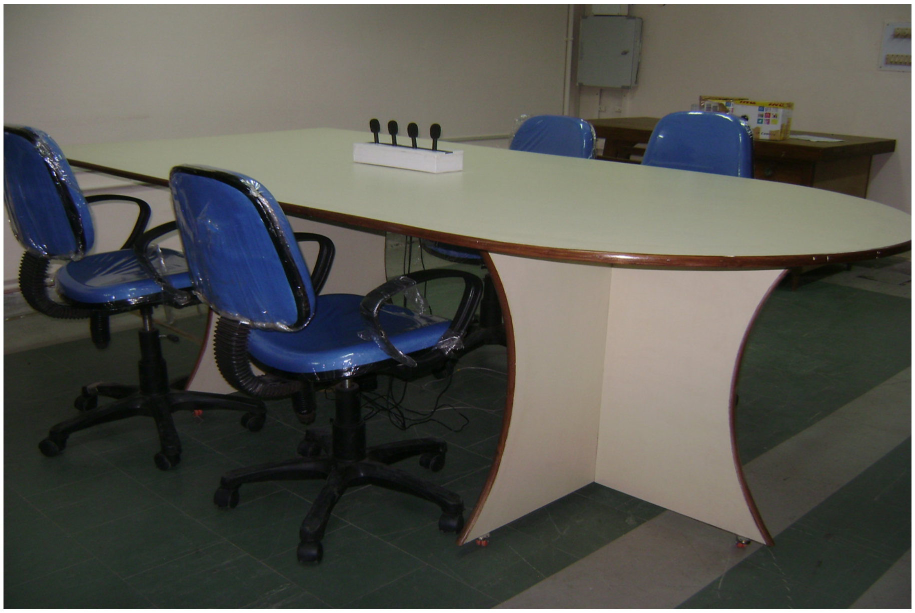

Figure 6.2: *Actual test bed used to acquire data*

6.2 AMI database

The AMI (Augmented Multi-party Interaction) project [15] is concerned with the development of technology to support human interaction in meetings, and to provide better structure to the way meetings are run and documented. The AMI Meeting Corpus contains 100 hours of meetings captured using many synchronized recording devices, and is designed to support work in speech and video processing, language engineering, corpus linguistics, and organizational psychology. It has been transcribed orthographically, with annotated subsets for everything from named entities, dialogue acts, and summaries to simple gaze and head movement.

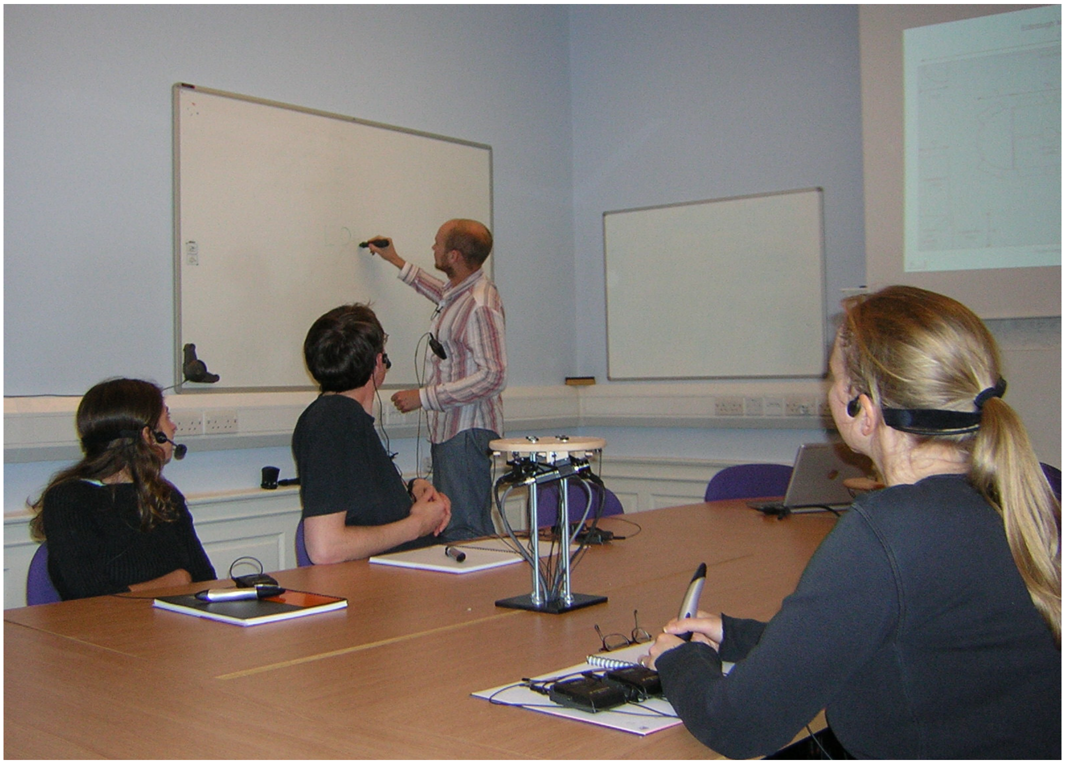

Figure 6.3: *AMI's instrumented meeting room*

Two-thirds of the corpus consists of recordings in which groups of four people played different roles in a fictional design team that was specifying a new kind of remote control. Controlling the data in this way allows us for better measures of how well the groups are doing, and to compare to new data where groups use these technologies that proves they help. However, it also limits the things people talk about. The remaining third of the corpus contains recordings of other types of meetings.

The project has a number of instrumented meeting rooms that enable the collection of multi modal meeting recordings. For each meeting, audio (captured from multiple microphones, including microphone arrays), video (coming from multiple cameras), slides (captured from the data projector), and textual information (coming from associated papers, captured handwritten notes and the white board) are recorded and time-synchronized. All of these streams are then available to be structured, browsed and queried within an easily accessible archive.

A subset of the multi modal data from the Augmented Multi-Party Interaction(AMI) corpus is used here to perform the experiments. It contains the annotated data of four participants. The duration of the meeting was around 30 minutes. The subjects in the meeting are carrying out various activities such as presenting slides, white board explanations and discussions round the table.

6.3 GRID corpus

GRID [9] is a large multi talker audio visual sentence corpus to support joint computational behavioral studies in speech perception. In brief, the corpus consists of high-quality audio and video (facial) recordings of 1000 sentences spoken by each of 34 talkers (18 male, 16 female). Sentences are of the form “put red at g nine now”. The corpus, together with transcriptions, is freely available for research use.

6.3.1 Sentence design

Each sentence consisted of a six word sequence of the form indicated in Table 6.1. Of the six components, three - color, letter, and digit were designated as keywords. In the letter position, “w” was excluded since it is the only multisyllabic English alphabetic letter. “Zero” was used rather than “oh” or “nought” to avoid multiple pronunciation alternatives for orthographic 0. Each talker produced all combinations of the three keywords, leading to a total of 1000 sentences per talker. The remaining components command, preposition, and adverb were fillers. Four alternatives were available in each filler position. Filler words were chosen to create some variation in contexts for the neighboring key words. Different gross phonetic classes (nasal, vowel, fricative, plosive, liquid) were used as the initial or final sounds of filler words in each position.

Table 6.1: *Sentence structure for the Grid corpus. Keywords are identified with asterisks*

Command	Color*	Preposition	Letter*	Digit*	Adverb
bin	blue	at	A-Z	1-9, 0	again
lay	green	by	excluding W		now
place	red	in			please
set	white	with			soon

6.3.2 Speaker population

The aim of speaker selection was to provide a sufficiently large number of speakers to allow users of the corpus to select subsets based on criteria such as intelligibility, homogeneity, and variety. Sixteen female and eighteen male talkers contributed to the corpus. Participants were staff and students in the Departments of Computer Science and Human Communication Science at the University of Sheffield. Student participants were paid for their contribution. All spoke English as their first language. All but three participants had spent most of their lives in England and together encompassed a range of English accents. Two participants grew up in Scotland and one was born in Jamaica. Ages ranged from 18 to 49 years. mean: 27.4 years.

6.3.3 Collection

Speech material collection was done under computer control. Sentences were presented on a computer screen located outside the booth, and talkers had 3 seconds to produce each sentence. Talkers were instructed to speak in a natural style. To avoid overly careful and drawn-out utterances, they were asked to speak sufficiently quickly to fit into the 3 seconds time window.

7

Performance evaluation

7.1 Speaker diarization experimental results

7.1.1 Evaluation methodology

The performance evaluation is carried out on two sets of multi modal data. One set is generated by the multi modal data acquisition test bed at MiPS Lab at IIT-Kanpur and the other is the AMI multi modal data corpus.

To facilitate for the synchronization of multi modal data i.e, the video frame rate of 25 fps, and the speech sampling rate of 44100 Hz, we consider 1 second segment interval for evaluating speaker change detection and subsequent speaker diarization. This avoids replicating video frames for a certain number of times as is done conventionally to synchronize multi modal data. The results obtained are compared with the annotated data of the AMI Corpus. The multi modal data recorded from the test bed has video frame rate of 30 fps and is manually annotated. Speaker diarization performance is usually evaluated in terms of Diarization Error Rate (DER), which is essentially a sum of three terms namely, missed speech (speech in the reference but not in the hypothesis), false alarm speech (speech in the hypothesis but not in the reference), and speaker match error (reference and hypothesized speakers differ). Hence the DER is computed as

$$DER = \frac{FA + MS + SMR}{SPK}\% \tag{7.1}$$

where missed speaker time (MS) is the total time when less speakers are detected than what is correct, false alarm speaker time (FA) is the total time when more speakers are detected than what is correct, speaker match error time (SMR) is the total time when some other speaker is speaking rather than the speaker detected and scored speaker time (SPK) is the sum of every speakers utterance time as indicated in the reference.

7.1.2 Separability analysis for multi modal features

The Bhattacharyya distance (BD), which is a special case of the Chernoff distance is a probabilistic error measure and relates more closely to the likelihood maximization classifiers that we have used for performance evaluation.

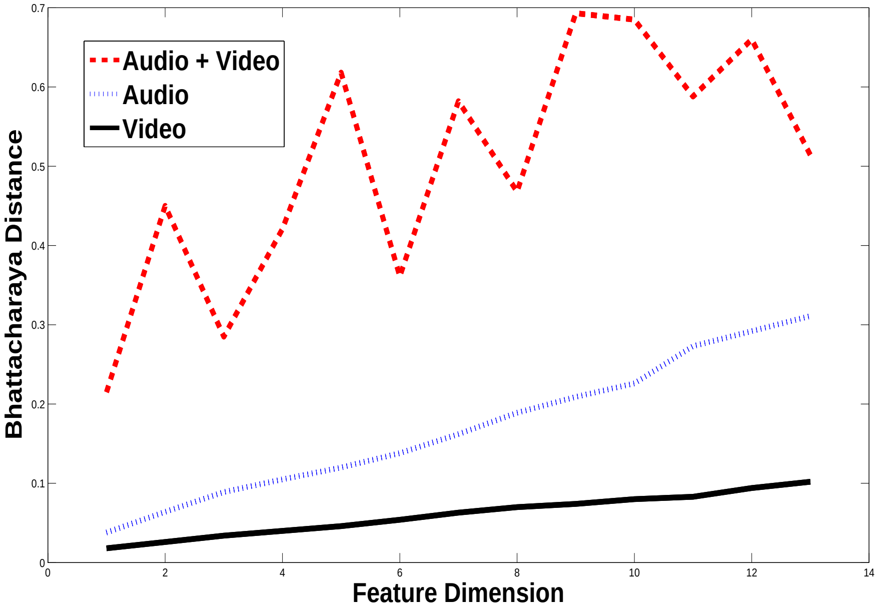

Figure 7.1: *Separability analysis results as the BD versus the feature dimension for unimodal and multi modal features*

Figure 7.1 illustrate the separability analysis results as the BD versus the feature dimension for both unimodal (speech only & video only) and multi modal (speech+video) features in Figure 7.1. The complementarity of the multi modal features when compared to unimodal speech features can be noted from Figure 7.1.

7.1.3 Experimental results

The reliability of each feature is determined by its speaker change detection performance on a small development set created from the unimodal speech or video data set.

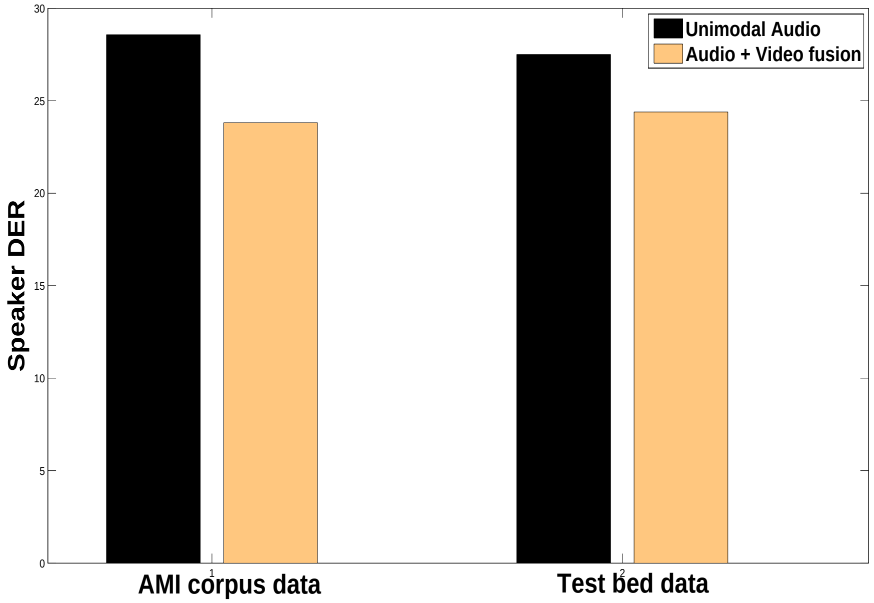

Figure 7.2: *Speaker DER using unimodal audio and multi modal information fusion on the two data sets*

The reliability values of the audio and video features computed from the development data set are given in Table 7.1, for the two corpora used in our experiments. The speaker diarization error rates (DER) for both the multi modal corpora used is also shown in Figure 7.2. It can be noted that reasonable reduction in DER is noted when the multi modal information fusion methodology based on DS theory is used.

Table 7.1: *Reliability of the unimodal information as computed from their feature vectors on the two multi modal data sets*

Unimodal Feature	Reliability on AMI corpus	Reliability on Test Bed data
Audio : X_{audio}	90.47	87.50
Video : X_{video}	87.50	78.04

7.2 Audio visual speech recognition experimental results

A typical GRID corpus sentence consists of 6 words. These are as described below

- Word 1: bin | lay | place | set;
- Word 2: blue | green | red | white;
- Word 3: at | by | in | with;
- Word 4: a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | x | y | z;
- Word 5: zero | one | two | three | four | five | six | seven | eight | nine;
- Word 6: again | now | please | soon;

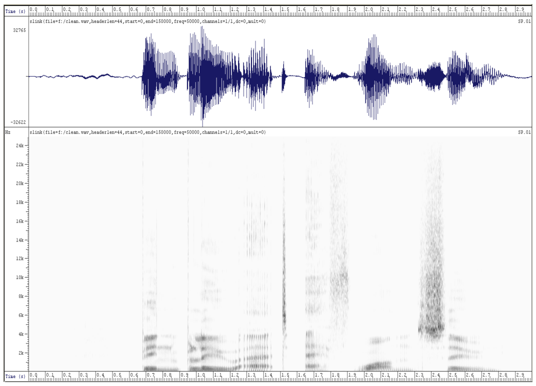

Figure 7.3: *Clean speech signal and its spectrogram*

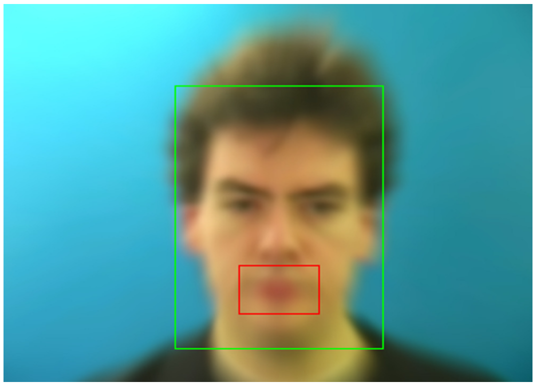

Figure 7.4: *Noisy video signal*

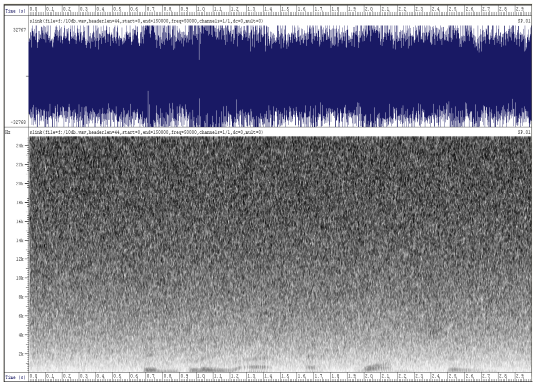

Figure 7.5: *Noisy speech signal and its spectrogram*

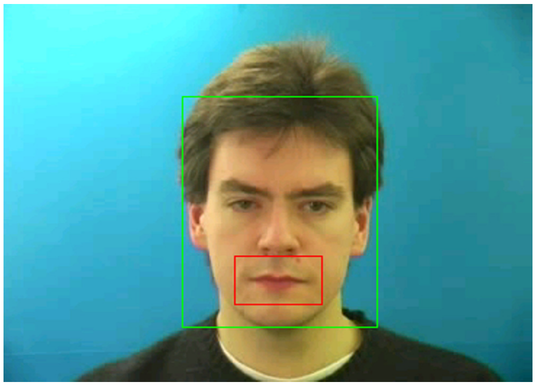

Figure 7.6: *Clean video signal*

We have used 80% of the total data as training data and remaining 20% as test data. Recognition was performed on every word of the sentence seperately as well as on the whole sentence as continuous speech. Experiments were carried out for 4 sets of noise conditions. First one is the clean speech and after that recognition experiment was done on speech corrupted with AWGN of SNR 30dB, 20dB & 10dB.

Word recognition results for all the noise conditions are given in Tables 7.2, 7.3, 7.4 & 7.5

Table 7.2: *Word recognition result for clean speech*

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Video reliability	52.33%	43.54%	44.04%	11.31%	21.89%	39.93%
Audio reliability	98.12%	95.74%	77.08%	72.10%	95.94%	99.94%
Video only result	50.33%	39.05%	40.55%	10.92%	25.24%	43.50%
Audio only result	96.99%	96.49%	79.89%	72.83%	96.65%	99.89%
Concatenated features result	82.98%	74.44%	57.62%	48.13%	71.19%	83.48%
DS fusion result	99.00%	96.63%	80.12%	74.34%	97.52%	99.89%

Whole sentence(Continuous speech) recognition results for various noise values are tabled in Table 7.6

Figure 7.7, 7.8, 7.9, 7.10, 7.11 & 7.12 shows graph of percent word recognition for different words for various noise conditions for audio only features, video only features, concatenated features and using DS fusion mechanism.

Table 7.3: *Word recognition result for speech corrupted with AWGN of SNR 30dB*

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Video reliability	52.33%	43.54%	44.04%	11.31%	21.89%	39.93%
Audio reliability	81.43%	89.50%	63.45%	59.61%	85.60%	88.64%
Video only result	50.33%	39.05%	40.55%	10.92%	25.24%	43.50%
Audio only result	81.94%	91.47%	64.16%	64.91%	86.36%	90.84%
Concatenated features result	80.42%	72.30%	54.41%	35.93%	58.49%	56.88%
DS fusion result	83.95%	91.97%	64.96%	65.97%	87.28%	92.67%

Table 7.4: *Word recognition result for speech corrupted with AWGN of SNR 20dB*

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Video reliability	52.33%	43.54%	44.04%	11.31%	21.89%	39.93%
Audio reliability	54.72%	78.34%	58.17%	35.91%	60.66%	68.92%
Video only result	50.33%	39.05%	40.55%	10.92%	25.24%	43.50%
Audio only result	53.35%	78.92%	57.00%	37.25%	67.90%	72.25%
Concatenated features result	56.67%	64.77%	48.97%	26.7%	45.38%	50.06%
DS fusion result	64.47%	79.80%	58.35%	42.67%	69.87%	75.91%

Table 7.5: *Word recognition result for speech corrupted with AWGN of SNR 10dB*

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Video reliability	52.33%	43.54%	44.04%	11.31%	21.89%	39.93%
Audio reliability	48.78%	64.12%	48.72%	18.92%	43.17%	49.70%
Video only result	50.33%	39.05%	40.55%	10.92%	25.24%	43.50%
Audio only result	48.95%	57.06%	51.51%	17.60%	47.84%	55.67%
Concatenated features result	62.91%	59.61%	44.45%	17.98%	36.39%	48.01%
DS fusion result	64.63%	60.38%	52.87%	21.86%	50.56%	59.45%

Table 7.6: *Continuous speech recognition results*

	Clean speech	SNR=30dB	SNR=20dB	SNR=10dB
Video	27.14%	27.14%	27.14%	27.14%
Audio	87.24%	78.20%	66.24%	49.32%
Concatenated	56.52%	48.71%	42.63%	35.16%

Figure 7.13, 7.14, 7.15 & 7.16 shows graph of percent word recognition against scale variable as described in Equation 5.1 for different values of noise conditions.

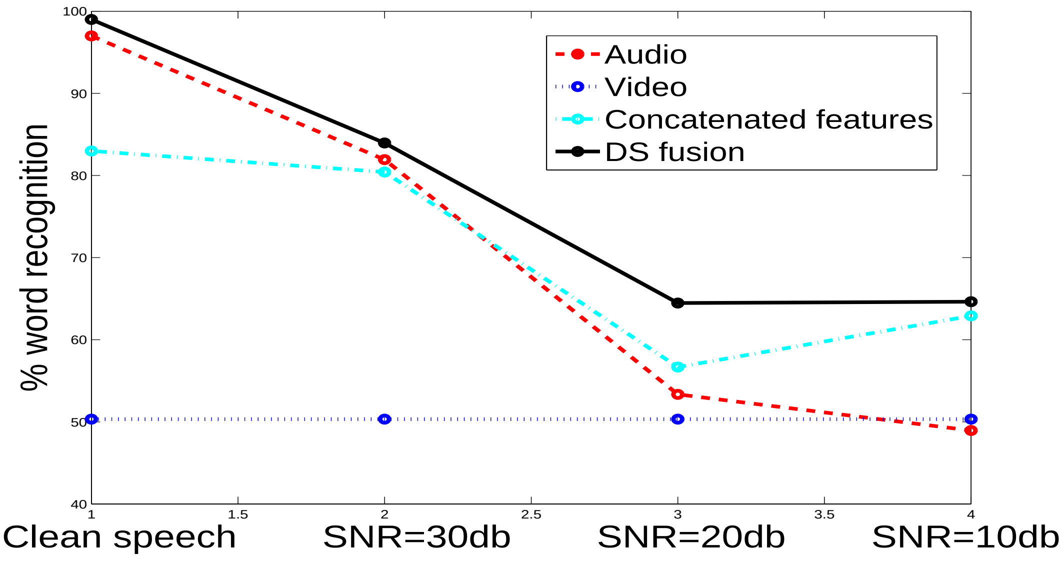

Figure 7.7: *Recognition results for word1="bin|lay|place|set"*

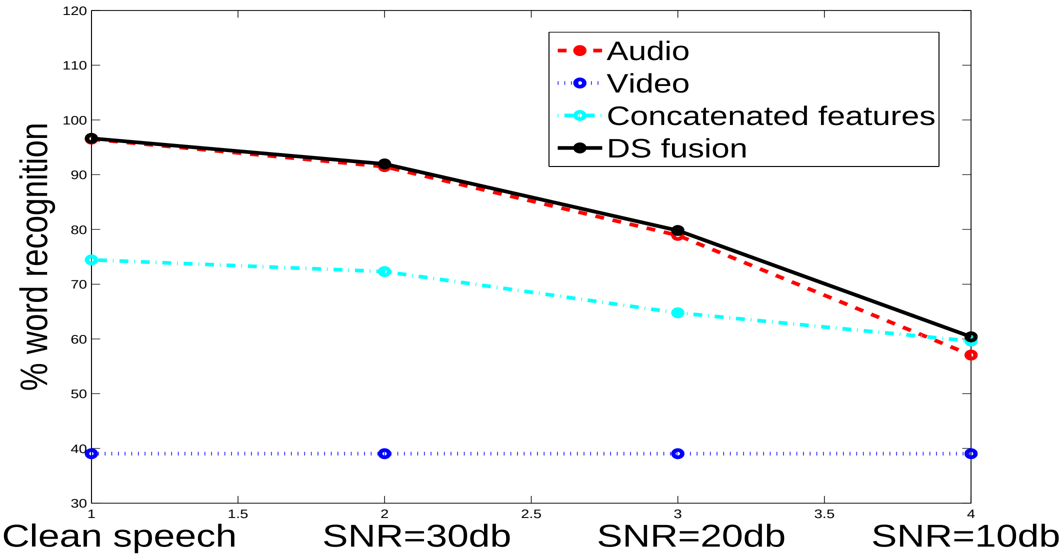

Figure 7.8: *Recognition results for word2="blue|green|red|white"*

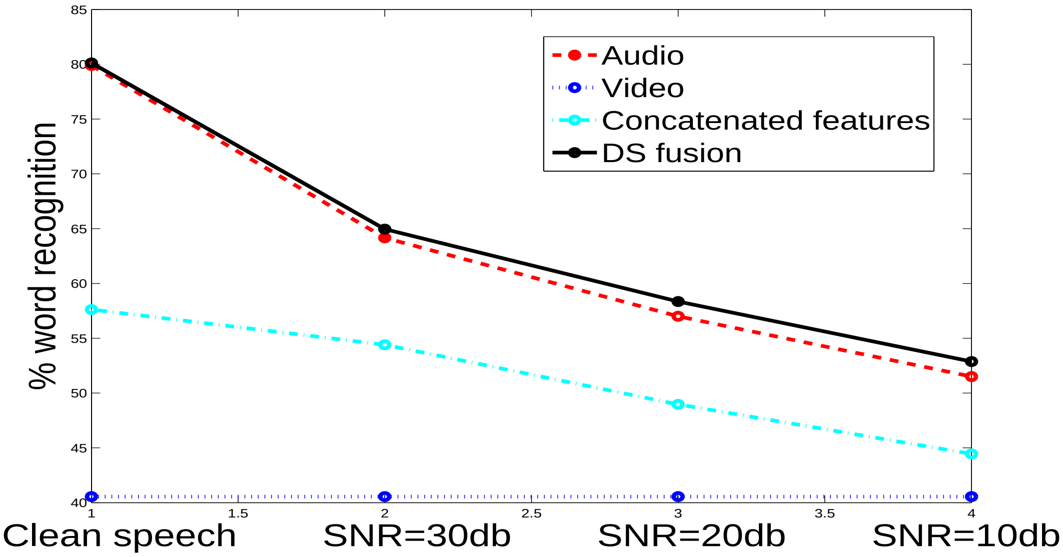

Figure 7.9: *Recognition results for word3="at|by|in|with"*

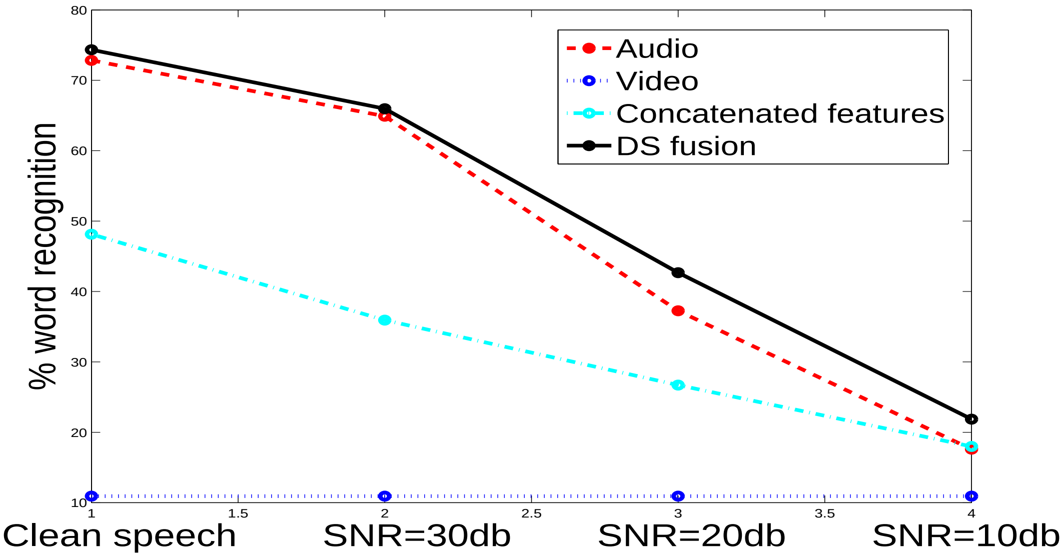

Figure 7.10: *Recognition results for word4="A-Z, except W"*

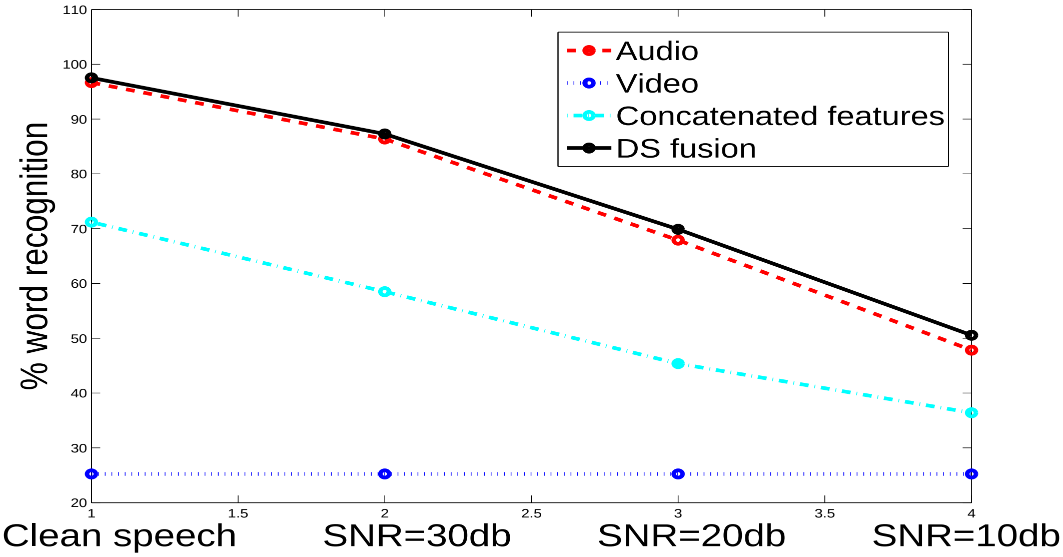

Figure 7.11: *Recognition results for word5="zero-nine"*

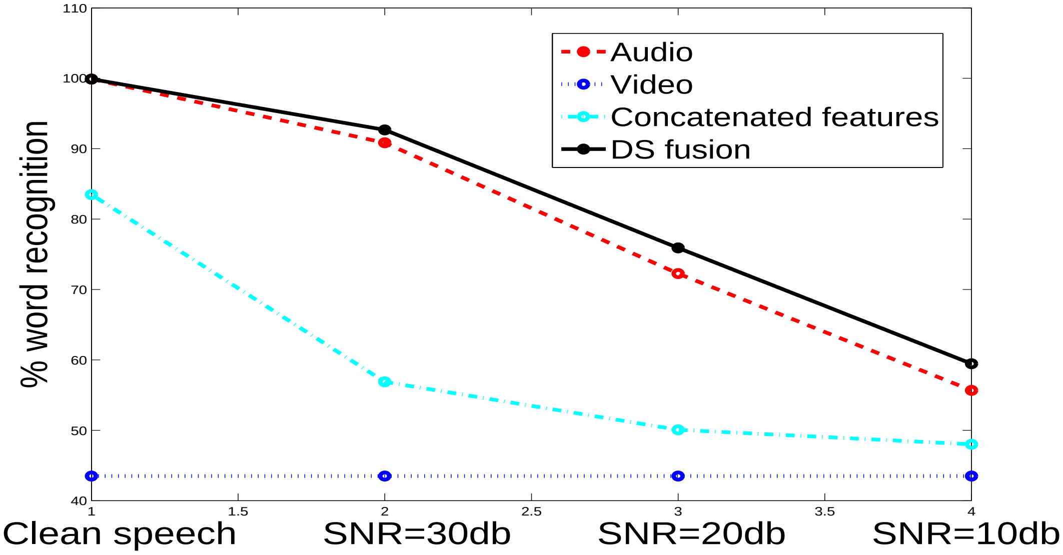

Figure 7.12: *Recognition results for word6="again|now|please|soon"*

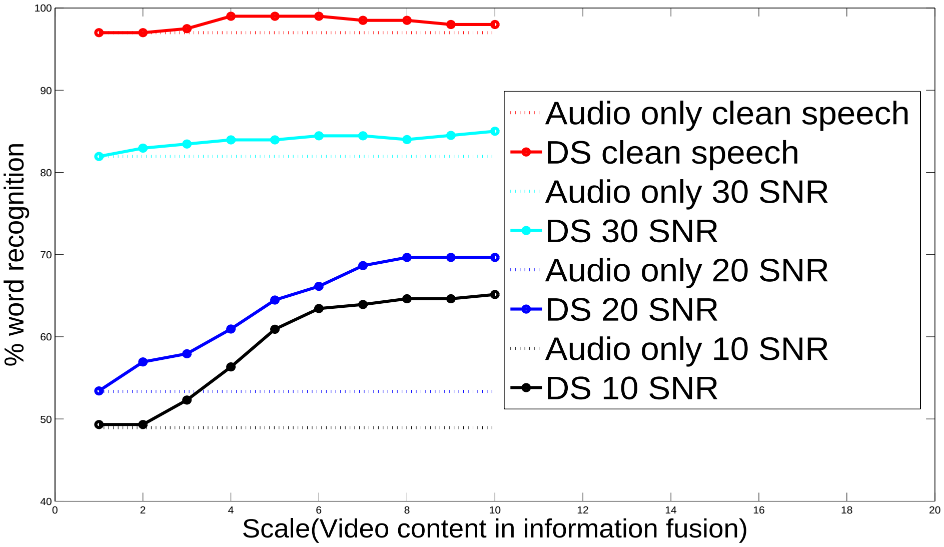

Figure 7.13: *Recognition results for word1="bin|lay|place|set" as a function of scale*

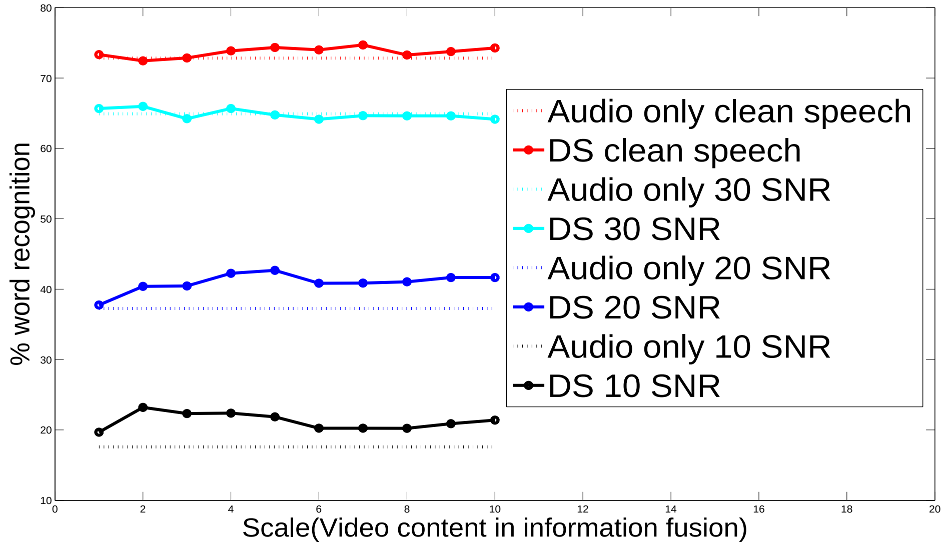

Figure 7.14: *Recognition results for word4="A-Z, except W" as a function of scale*

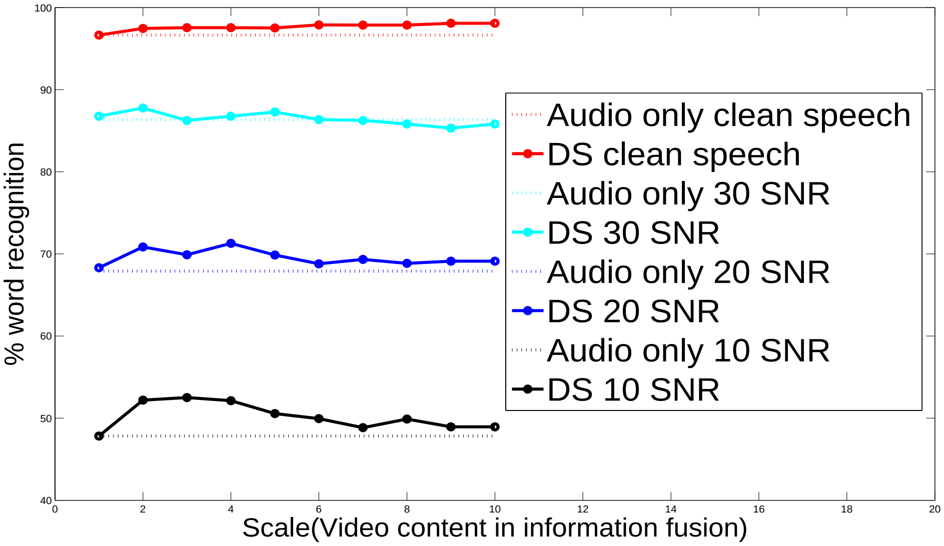

Figure 7.15: *Recognition results for word5="zero-nine" as a function of scale*

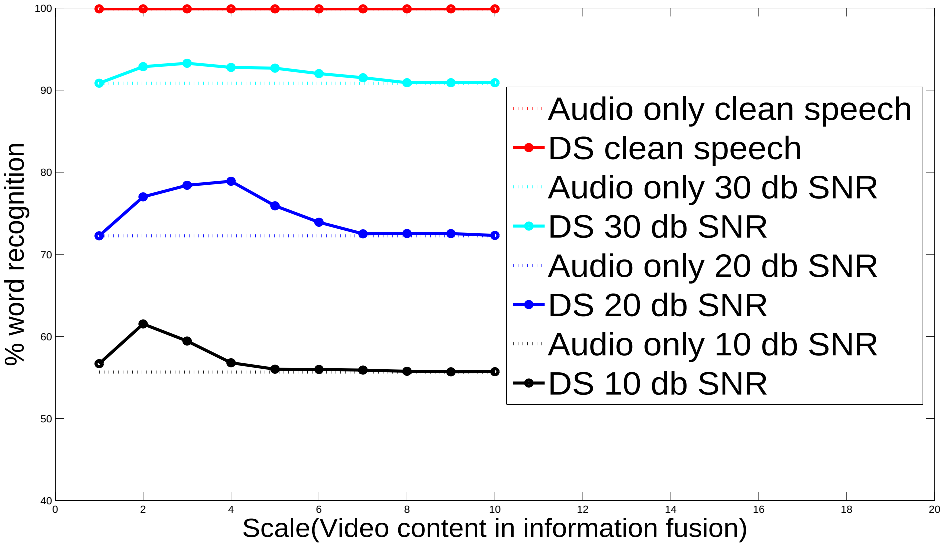

Figure 7.16: *Recognition results for word6="again|now|please|soon" as a function of scale*

8

Conclusion and future scope

A methodology to fuse information from multiple modalities using Dempster Shafer theory has been proposed here for speaker diarization and audio visual speech recognition. Performance of unimodal speech based systems reduces drastically when speech is corrupted with noise and cross talk. Hence the use of additional information from the video modality can help alleviate this problem. As the experimental results shows there is a significant improvement in the system performance due to the use of multiple modalities and subsequent fusion. The Dempster Shafer theory provides a framework for soft fusion when compared to the conventional probabilistic fusion framework used in speech applications.

8.1 Speaker diarization system performance

Performance of speaker diarization system increases considerably when video information is fused with audio information as compared to audio only based system.

A parameter of the system performance, Diarization Error Rate (DER), is considerably low for system based on DS fusion of audio and video information, as compared to audio only system. This result is being shown in Figure 7.2 for ami database as well as data collected in lab meeting room setup.

Table 7.1 shows that audio has been more reliable than video, which is quiet evident as there are certain sounds which can be produced without involving mouth movement (e.g. nasals). This fact also reflects from Figure 7.1.

8.2 Audio visual speech recognition system performance

Speech recognition problem is more challenging than speaker diarization problem because it is a multiple hypothesis problem. Moreover video information for speech recognition case is even less reliable as many type of sounds can be generated from one type of lip shape and different speakers have different speaking styles.

For clean speech, audio provides excellent recognition results but with increase in noise content, audio performance falls drastically. In high noise conditions even this less reliable video information can be quiet helpful in improving recognition results as is being shown in Table 7.2, 7.3, 7.4 & 7.5 and graphically being shown in Figure 7.7, 7.8, 7.9, 7.10, 7.11 & 7.12.

Here results based on audio visual information fusion based on DS theory have outperformed audio only recognition results and recognition results of concatenated audio-visual features.

The weight being given to video information changes the recognition results, as per it's reliability. For the example of word1 = "bin | lay | place | set", for which video reliability is good(52.33%), when video information is given higher weightage, it produce excellnet recognition results, especially for noisy conditions, as being shown in Figure 7.13. For other words also increasing video weightage improves recognition results, but as video content gets too high performence degrades, because of less reliability of video than audio.

8.3 Future scope

The work presented in this thesis has the potential to be extended in various applications and to incorporate newer methods to increase the efficiency. Audio visual speech recognition system can be employed in a car environment for assisted driving, which would take speech commands for it's functioning.

Newer information fusion methods can also be used to increase the efficiency and accuracy of the system.

Appendix A

HMM modeling of audio and video data

Audio and video features obtained for the whole sentence are separated for each word of the sentence using transcription files. These features are then trained to generate HMMs for each word. The tool used to generate these HMMs is Hidden Markov Model Toolkit(HTK). Before using HTK to train the models, feature vectors which are in ASCII format are first converted to HTK format.

A.1 Hidden Markov Model Toolkit(HTK)

HTK (Hidden Markov Model Toolkit) is software toolkit for handling HMMs. It is mainly intended for speech recognition, but has been used in many other pattern recognition applications that employ HMMs. HTK is used for building and manipulating hidden Markov models, being the core of most state-of-the-art speech recognition systems.

The main HTK functions that are used in this audio-visual speech recognition application are described below

A.1.1 HParse

HTK recognizer requires a word network to be defined using a low level notation called HTK Standard Lattice Format (SLF), in which each word instance and each word-to-word transition is listed explicitly. This word network can be created automatically from the grammar above using the HParse tool.

Grammar file that we have used is:

```
$command = bin | lay | place | set;
$color = blue | green | red | white;
$preposition = at | by | in | with;
$letter = a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v |
x | y | z;
$digit = zero | one | two | three | four | five | six | seven | eight | nine;
$adverb = again | now | please | soon;
($command $color $preposition $letter $digit $adverb)
```

Command to use HParse function is

```
HParse gram wdnet
```

This will create an equivalent word network in the file wdnet.

A.1.2 HCompV

The first step in HMM training is to define a prototype model. The parameters of this model are not important, its purpose is to define the model topology. We have used a 3-state left-right HMM with no skips such as the following:

```
o <VecSize> 13 <USER>
h “proto”
<BeginHMM>
<NumStates> 5
<State> 2
<Mean> 13
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> 13
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> 13
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> 13
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> 13
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> 13
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 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 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>
```

The proto file for video modeling has 16 instead of 13 features.

Config file used is as follows:

```
# Coding parameters
TARGETKIND = USER
TARGETRATE = 100000.0
SAVECOMPRESSED = T
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
```

```
ENORMALISE = F
```

The HTK tool HCompV will scan a set of data files, compute the global mean and variance and set all of the Gaussian in a given HMM to have the same mean and variance. Hence, assuming that a list of all the training files is stored in train.scp, the command

```
/usr/local/bin/HCompV -C config -f 0.01 -m -S train.scp -M hmm0 proto
```

will create a new version of proto in the directory hmm0 in which the zero means and unit variances above are been replaced by the global means and variances of all the feature vectors.

A.1.3 HERest

The flat start monophones stored in the directory hmm0 are re-estimated using the embedded re-estimation tool HERest invoked as follows:

```
/usr/local/bin/HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp
-H hmm0/macros -H hmm0/hmmdefs -M hmm1 monophones
```

The effect of this is to load all the models in hmm0 which are listed in the model list monophones. These are then re-estimated using the data listed in train.scp and the new model set is stored in the directory hmm1. This step is repeated 15 times. After this we get HMM models for all the words in the vocabulary.

A.1.4 HVite

This is the recognizer function of HTK and is used to recognize test data. Assuming that test.scp holds a list of the coded test files, then each test file will be recognized and its transcription is saved in an MLF called recout.mlf by executing the following:

```
/usr/local/bin/HVite -H hmm15/macros -H hmm15/hmmdefs -S test.scp -l ’*’ -i
```

```
recout.mlf -w wdnet -p 0.0 -s 5.0 dict tiedlist
```

HVite also has a option of generating lattices and N-best output, which give N best match options and there log probabilities.

A.1.5 HResults

Assuming that the MLF testref.mlf contains word level transcriptions for each test file, then the actual performance can be determined by running HResults as follows

```
/usr/local/bin/HResults -I testref.mlf -f tiedlist recout.mlf
```

References

[1] A. Adjoudani and C. Benoit. On the integration of auditory and visual parameters in an hmm-based asr. *Stork and Hennecke [91], pp. 461 - 472*, 2001. 3

[2] Jitendra Ajmera, Guillaume Lathoud, and Iain McCowan. Clustering and segmenting and their locations in meetings. *International Conference on Acoustics, Speech and Signal Processing, pp. 605 – 608*, 2004. 25

[3] S. Basu, N. Oliver, and A. Pentland. 3d modeling and tracking of human lip motions. *International Conference on Computer Vision, pp. 337*, 1998. 1

[4] C. Benoit. The intrinsic bi modality of speech communication and the synthesis of talking faces. *The Structure of Multi modal Dialogue II, pp. 315 – 325*, 2000. 2

[5] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing, pp. 195-207*, 1979. 2

[6] Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, Q2, 1998. 41

[7] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. *International Conference on Acoustics, Speech and Signal Processing, pp. 557-560*, 1993. 1

[8] C. Bregler and Y. Konig. 'eigenlips' for robust speech recognition. *International Conference on Acoustics, Speech and Signal Processing, pp. 669-672*, 1994. 2

[9] Martin Cookea, Jon Barker, Stuart Cunningham, and Xu Shao. An audio-visual corpus for speech perception and automatic speech recognition. *Acoustical Society of America, pp. 2421 – 2424*, 2006. 61

[10] Gerald Friedland, Hayley Hung, and Chuohao Yeo. Multi modal speaker diarization of real world meetings using compressed domain video features. *International Conference on Acoustics, Speech and Signal Processing, pp. 4069 – 4072*, 2009. 40

[11] Maurizio Gentilucci and Luigi Cattaneo. Automatic audiovisual integration in speech perception. *Springer-Verlag, pp. 66 – 75*, 2005. 1, 8

[12] Kentaro Ishizuka, Shoko Araki, Kazuhiro Otsuka, Tomohiro Nakatani, and Masakiyo Fujimoto. A speaker diarization method based on the probabilistic fusion of audio-visual location information. *International Conference on Multi modal Interfaces and Machine Learning for Multi modal Interaction, pp. 55 – 62*, 2009. 40

[13] Ludmila I. Kuncheva. *Combining pattern classifiers: Methods and algorithms.* 2004. 12

[14] Jong-Seok Lee and Cheol Hoon Park. *Adaptive Decision Fusion for Audio-Visual Speech Recognition.* 2008. 15, 55

[15] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. The ami meeting corpus. *The AMI Project Consortium*, 2005. 59

[16] Ara V. Nefian, Luhong Liang, Xiaobo Pi, Liu Xiaoxiang, Crusoe Mao, and Kevin Murphy. A coupled hmm for audio-visual speech recognition. *International Conference on Acoustics, Speech and Signal Processing, pp. 2013 - 2016*, 2002. 16

[17] Chalapathy Neti, Gerasimos Potamianos, Juergen Luettin, Iain Matthews, Herve Glotin, Dimitra Vergyri, June Sison, Azad Mashari, and Jie Zhou.

Audio-visual speech recognition. *Workshop 2000 Final Report*, 2000. 4, 14, 15

[18] Jose M. Pardo and Charles Wooters. Speaker diarization for multiple-distant-microphone meetings using several sources of information. *IEEE Transactions on Computers Volume 56, Issue 9, pp. 1212 – 1224*, 2007. 25

[19] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE, pp. 257 – 286*, 1989. 29

[20] Glenn shafer. *A Mathematical Theory of Evidence.* 1976. 17

[21] David G. Stork and Marcus E. Hennecke. *Speechreading by humans and machines: Models, systems and applications.* 1995. 7

[22] Jean-Philippe Thiran, Ferran Marques, and Herve bourlard. *Multi modal signal processing: Theory and applications for human-computer interaction.* 2010. 12

[23] Michiel Visser, Mannes Poel, and Anton Nijholt. Classifying visemes for automatic lipreading. *V. Matousek et al, pp. 843*, 1999. 9

Publications

[1] Dheeraj Kumar, R Malhotra, A Singh, and Rajesh M Hegde. Multimodal speaker diarization using a soft belief function. *7th International Conference on Natural Language Processing, ICON 2009, Hyderabad, India, pp. 376 – 381*, 2009.