Bimodal Speech Recognition

Dissertation

to obtain the academic degree of

Dr. rer. nat.

in the faculty of Mathematic,Informatics and Nature Science

University of Hamburg,Germany

presented by

Tian Gan (M.Sc.)

from Xi'an (China)

Juni 2012

To My Family

supervised by:

Prof. Dr. Wolfgang Menzel

reviewed by:

Prof. Dr. Wolfgang Menzel

Prof. Dr. Jianwei Zhang

Oral Examination on: 12th. Nov. 2012

Abstract

Automatic speech recognition (ASR) is difficult to improve further, if only acoustic evidence is considered. However, two lines of study have been proposed to improve the performance of ASR by using additional information. On the one hand, audio-visual speech recognition (AVSR) uses an extra channel of visual cues for compensating reduced signal quality, e.g., in a noisy environment, in order to improve speech recognition performance. On the other hand, articulatory information was introduced to model coarticulation effects based on insights form the speech production procedure. The goal of this thesis is to investigate possibilities and benefits of integrating articulatory information into AVSR systems.

As one of the research questions, the issue of feasibility is considered first. We defined four different types of approaches for using articulatory information. Except for the articulatory raw data approach, the design and implementation of other three ones are all discussed in this thesis:

1. The articulatory transcription approach uses an HMM/N-best decision framework, where an N-best decision schema is a method to optimally combine decisions from different articulatory channels.

2. The articulatory feature approach uses an ANN/HMM framework to extract abstract articulatory classes as articulatory features to complement or replace the low-level audio and visual features.

3. The articulatory modeling approach uses dynamic Bayesian networks (DBN) to build different training and decoding structures for integrating articulatory information.

Compared to the results of a low-level information-based AVSR, we found that all the results from the above mentioned systems indicate an improvement in recognition accuracy.

As a second research question, the modeling issue is emphasized in this thesis. Frame, sub-phone, phone and word are the four levels of phonetic observation to be considered. The level of articulatory information fusion and synchronization is analyzed in different approaches respectively. The articulatory modeling approach was found to be promising for integrating

loosely synchronized multi-channel information.

Zusammenfassung

Es ist schwer, automatische Spracherkennung (ASR) weiter zu verbessern, wenn nur das akustische Signal betrachtet wird. In de Literatur werden jedoch zwei alternative Ansätze verfolgt, um die Leistung der ASR unter Einbeziehung zusätzlicher Informationen zu verbessern. Zum einen audiovisuelle Spracherkennung (AVSR), die einen zusätzlichen Kanal visueller Merkmale nutzt, um reduzierte Signalqualität zu kompensieren, z. B. für Spracherkennung in Umgebungen mit viel Hintergrundlärm. Zum anderen wurden artikulatorischen Informationen Modell eingeführt, um Koartikulationseffekte und Erkenntnisse der Sprachproduktion mit in das Spracherkennungsverfahren einflies̈en zu lassen. Das Ziel dieser Arbeit ist es, die Möglichkeiten und Vorteile der Integration von artikulatorischen Informationen in AVSR-Systeme zu untersuchen.

Als eine der Fragestellungen wird zuerst die Frage der Machbarkeit untersucht. Wir haben vier verschiedene Ansätze für die Verwendung artikulatorischen Informationen definiert. Mit Ausnahme des Ansatzes zur Nutzung von artikulatorischen Rohdaten werden im Rahmen dieser Arbeit alle diese Ansätze diskutiert:

1. Der Ansatz der artikulatorischen Transkription nutzt ein HMM- / N-Beste-Framework als Entscheidungsgrundlage. Das N-beste Entscheidungsschema ist ein Verfahren zur optimalen Kombination von Entscheidungen aus verschiedenen artikulatorischen Kanälen.

2. Der Ansatz der artikulatorischen Merkmale nutzt ein ANN- / HMM-Framework, um abstrakte Klassen als artikulatorische Merkmale zu extrahieren und die Low-Level-Audio- und visuellen Merkmale durch diese Klassen zu ergänzen oder zu ersetzen.

3. Die artikulatorische Modellierung verwendet dynamische Bayessche Netze (DBN) zur Integration mit verschiedenen Strukturen für das Training und die Dekodierung von artikulatorischen Informationen.

Verglichen mit den Ergebnissen eines einfachen informationsbasierten AVSR fanden wir, dass alle Ergebnisse aus den oben genannten Systemen auf eine Verbesserung der Erkennungsgenauigkeit hindeuten.

Als zweite Forschungsfrage richtet sich diese Arbeit auf die Modellierung. Es werden Frame-, Sub-Phon, Phon und Wort als die vier Ebenen der phonetischen Modellierung betrachtet. Die Ebene der artikulatorischen Informationsfusion und der Synchronisation werden für die verschiedenen Ansätze analysiert. Die artikulatorische Modellierung wurde als besonders vielversprechend für die Integration von lose synchronisierten Multi-Channel-Informationen identifiziert.

Acknowledgement

First of all, thanks to my supervisor, Professor Wolfgang Menzel at the University of Hamburg. His efforts helped me to adapt to the research work with a scientific attitude. I benefitted greatly not only from his careful supervision and encouragement to my entire research work, but also from his continuous support to my daily life in Hamburg.

Furthermore, I am grateful to the members of the graduate school CINACS (Cross-modal Interaction in Natural and Artificial Cognitive Systems) for their discussions and cooperations which helped me to broaden my view of cross-modal interaction in different areas. I would like to single out Professor Jianwei Zhang, the coordinator of CINACS. His coordination between Tsinghua and Hamburg University provided us plenty of scientific events, such as summer schools, colloquia and workshops, and an excellent interdisciplinary research environment.

Part of this work was completed while I was in the Multimedia Group at the computer and science department of Tsinghua University, Beijing, China. Thanks go to Professor Shiqiang Yang who supported me greatly both financially and ideally. The CINACS students in Tsinghua also helped me a lot by providing advice and assistance in the beginning of my Beijing life.

Last but not least I would like to thank to my parents for their assistance throughout my education, Jianfang and Jingtian for giving me emotional support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Audio Visual Speech Recognition

After almost 40 years of research in Automated Speech Recognition (ASR), scientists have covered applications ranging from speaker dependent, isolated word recognition, to speaker independent, large vocabulary, continuous speech recognition (Petajan, 1984; Potamianos et al., 2004; Galatas et al., 2011). The technology has reached a level of performance which seems difficult to be improved further, if only acoustic evidence is considered. On the other hand, most of the currently available systems require proper acoustic conditions, including a quiet environment, good quality microphones, a suitable distance to the microphone, etc. There is a clear necessity to overcome these limitations by including additional speech-related information into the recognizer.

Visual speech is a natural candidate here, because it is independent of the acoustic environment. In general, visual speech refers to any speech related information observed by the visual system of humans or computers. Gestures and lip motions are two different representations of visual speech. Both of them can be used for helping to overcome the limitations of a purely acoustic recognizer under different multimodal environments. On the one hand, gestures, which are normally defined as any bodily motion, especially the motion of the face or the hand, can be observed directly by the cameras to interpret a sign language. In the area of human-robot interaction, gestures can be used with speech together to achieve certain speech recognition and understanding tasks (Gasteratos et al., 2008). On the other hand, if available, humans use both acoustic information and a speaker's lip movement to recognize speech. Evidence from human speech perception consistently shows that visual cues from lip motions might considerably contribute to speech comprehension.

In this study we focus only on the visual information from lip movements as a modality of the bimodal speech recognition system. Not surprisingly, since the first attempt by Petajan in 1984 (Petajan, 1984), a range of audio visual speech recognition (AVSR) systems has been developed, which confirmed the initial assumption that lip movements information is particularly helpful for recognizing noisy speech. Their results convincingly show recognition rate improvements in noisy environments, where the recognition rate of acoustic-only recognizers drops significantly. Although there are clear differences in how these systems process audio and visual information and combine them together, they all share a quite similar system architecture based on a state-of-the-art approach to word recognition using phones as a subword modeling unit. In a standard AVSR system, as shown in Figure 1.1, acoustic and visual features are extracted from the incoming audio and visual speech signals and passed to the modeling component for training, which estimates audio and visual subword unit probabilities either separately or in an integrated manner. These models are subsequently used in the lexical decoding process to recognize testing utterances.

1.2 Articulatory Feature Based AVSR

All of the audio visual speech recognition systems we introduced in the previous section are conventional AVSRs based on acoustic and visual information. That is, audio and visual speech features in such systems are based on acoustic information to describe phonetic units. Audio and visual speech are modeled as two channels of continuous units. Each of these units corresponds to a hidden phonetic state or a sequence of states. These units are called phones and visemes in such AVSR systems. Although some success has been made with combining acoustic and visual information in such systems, (Potamianos et al., 2004) insights from speech production inspired us to explore articulatory information as an extra source of information for AVSR systems.

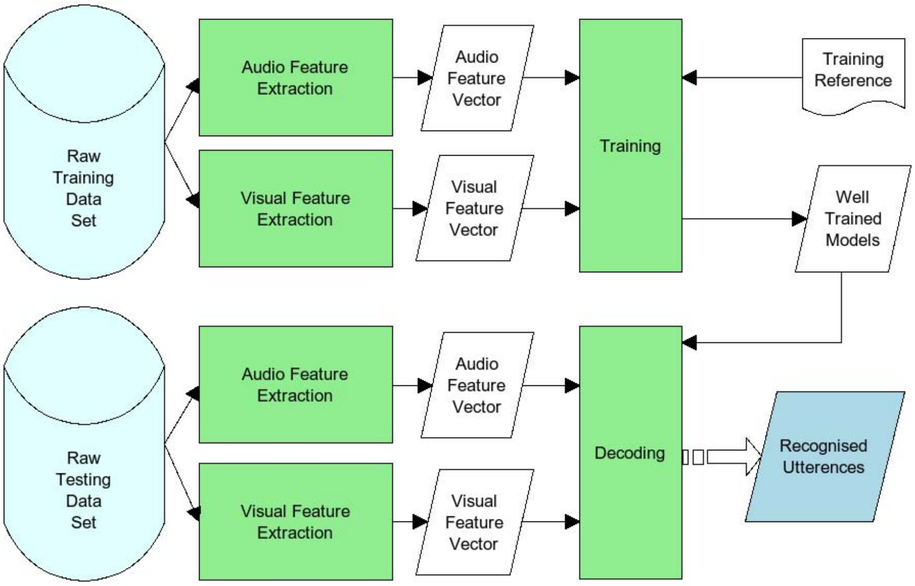


Figure 1.1: A general framework of audio visual speech recognition system

Articulation information based speech recognition systems have been proposed for several years and there is an increasing amount of work addressing such methods. Articulatory information can be either detailed numerical descriptions of the movements of articulators during speech production (Papcun et al., 1992; Thomas, 1994) or articulatory features which characterize essential aspects of articulation by using an intermediate abstract representation between signals and phonetic units (Deng and Sun, 1993; Erler and Freeman, 1996; Kirchhoff, 1998; Ghosh and Narayanan, 2011). The former one, articulatory raw data, is not considered in this thesis due to its complex recording equipment and methods. The latter one, together with other two types of articulatory information, named as articulatory transcription and articulatory modeling are applied to speech recognition in this thesis. Articulatory transcriptions make use of articulatory information as a symbolic representation of human speech in parallel channels. Articulatory modeling amounts to building model structures with several channels according to findings from articulatory phonology (Browman and Goldstein, 1993). A detailed explanation of articulatory information is given in the following chapters.

As mentioned, a number of ideas have been proposed for using articulatory information in conventional acoustic recognition systems. But as far as we know, only few of them (Gowdy et al., 2004; Saenko et al., 2004), have addressed the question of representing visual cues as parts of articulatory information. The fundamental idea of AVSR is to complement acoustic information with visual cues. Visual cues can be retrieved from various sources including, in this thesis, the movements of the lips. Because the lips are one of the important articulators, an AVSR system could also be categorized as an instance of articulatory information based ASR. In conventional AVSR systems, however, visual cues are the only information related to articulatory gestures. The audio channel still exploits traditional acoustic signal processing techniques. In contrast to the previously mentioned articulatory information based ASR systems i.e. (Papcun et al., 1992), conventional AVSRs measure the movements of articulatory gestures only implicitly. Despite this measurement being implicit, AVSRs have already shown an advantage for improving recognition accuracy especially in noisy environments. In this thesis we combine the idea of articulatory information and AVSR. For both, the audio and the visual channel, several types of representations based on articulatory information will be used instead of purely signal and image based parameters.

For using articulatory information in AVSR, there are three potential advantages that motivated this research. Firstly, based on speech production theory AVSR naturally uses partial articulatory information. There is an apparent correlation between some of the articulatory features and the visual shape of the lips during speaking, namely for labial consonants which are pronounced with closed lips and the roundedness feature which provides important cues to distinguish different kinds of vowels. Articulatory information might lend itself as an appropriate interface to integrate visual cues into the recognition procedure. In contrast to the audio-only articulatory feature based speech recognition system, the articulatory information can be better modeled using both the audio and visual cues. Secondly, in conventional AVSRs, the fusion of information from the audio and the visual channel is an important issue. Different sources of information can be integrated at different stages, i.e. during the feature extraction stage for feature fusion, during training for model fusion and at the decoding stage for decision fusion. Articulatory information provides us with an intermediate representation to study and compare these fusion techniques in audio-visual speech recognition. Thirdly, articulatory information is well suited to model the coarticulation phenomenon of speech production. While producing speech, humans move their articulators asynchronously rather than well aligned to the phonetic units. This asynchronicity can be better dealt with articulatory information based AVSR than in AVSR based solely on phonetic information.

Generally speaking, this thesis aims at increasing the robustness of acoustic-only recognizers by fusing articulatory information from both channels. According to the advantages of using articulatory information, several scientific goals are set in the framework of this thesis to achieve this ambition. Firstly, articulatory information should be clearly defined in the context of AVSR systems. Then, articulatory information based AVSR systems should be designed and implemented based on these different types of articulatory information. Thirdly, the information fusion and asynchronicity should be analyzed in these systems in order to find a proper way for applying articulatory information within an AVSR system.

1.3 Outline of Thesis

The remainder of this thesis is organized as follows:

In Chapter 2 we describe the basic idea of articulatory information based speech recognition which includes the motivation, definition and methodologies of using articulatory information. Three approaches to work with articulatory information, which are based on articulatory features, articulatory transcriptions and articulatory modelling, are introduced.

In Chapter 3 various feature extraction methods will be explained. Here ”feature” refers to a low level representation used for statistical training and testing. Feature extraction from both, the audio and the visual channel, is discussed.

In Chapter 4 three basic audio-visual classification methods are described. These methods have been used in speech recognition for a long time and provide the formal foundation of our articulatory information based AVSR systems.

Based on different articulatory information approaches introduced in Chapter 2, three AVSR systems are designed and implemented. In the subsequent Chapters 5 to 7 we describe these systems including their formal background, design and implementation. We provide experimental results and discuss advantages and disadvantages of the different approaches.

Chapter 5 presents the articulatory transcription based AVSR system which applies the HMM/N-best classification framework.

Chapter 6 describes the articulatory feature based AVSR system and its HMM/ANN architecture.

Chapter 7 introduces the articulatory modeling based AVSR system and its formal background Dynamic Bayesian networks.

Chapter 8 gives a summary, discussion, and suggestions for future work.

Chapter 2

Articulatory Information based Speech Recognition

In this chapter, we will first describe the general idea of articulatory information, including its theoretical motivation and some famous experimental investigations which point to the advantages of using articulatory information for AVSR systems. Secondly various methodologies for using articulatory information in speech recognition are reviewed and categorized. Finally, the concepts of information fusion and the problem of asynchronicity are introduced. These issues are the main points during the analysis of different approaches to articulatory information based AVSR in the following chapters.

2.1 Motivation

2.1.1 Speech Production Procedure

As we know from the study of speech production, speech is the result of a complex interaction of physical and emotional factors. All these factors, which influence the speech output, compose the human speech production procedure. As depicted in Figure 2.1, it can be categorized into three different steps.

The first step is the emergence of a motivation (a desire or need) to communicate, which is happening in the brain. The formulation of ideas and feelings is translated by the brain into language and motor programs that operate speech muscles. Nerve impulses then transmit the communication

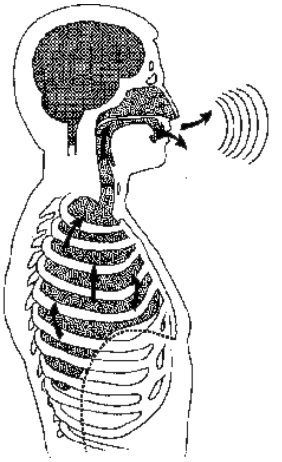


Figure 2.1: Human Vocal System. (PVCrp.com, 2010)

signals to muscles throughout the speech mechanism.

The respiratory system plays an important role in the second step. Muscles start to compress air in the lungs, then forcing it to flow upward through the trachea (windpipe) and larynx (voice box). This process supplies the power source for vocal folds vibration and speech sound generation. The voice-activated respiratory muscles then relax in order to let the breath enter the lungs for the next part of an utterance. The interaction between air and vocal folds can lead to vocal folds vibration. This vibration eventually creates sound waves which are the source of the speech and the voice.

In the third step, the sound wave passes through the upper vocal tract, including the throat and mouth. Depending on the shape of the throat and mouth cavities, certain frequencies of the sound wave will be amplified or suppressed according to the resonance phenomenon. At the same time, the speech articulators, such as the tongue, jaw and lips, move their positions and change their shapes to alter the sound wave while it passes through the mouth. As a result, speech sounds, that is, vowels and consonants, are produced as speech production results.

Acoustics based ASRs focuses only on the results of speech production. Knowledge of speech perception instead of speech production is used to extract acoustic cues and phonetic information. In contrast to acoustic in-

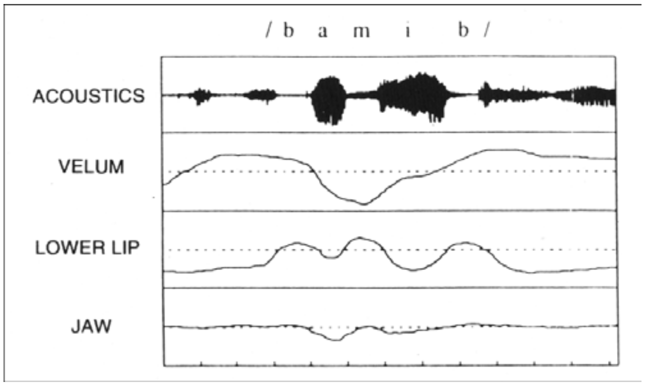


Figure 2.2: Acoustic waveform and measured articulatory trajectories for utterance of “Its a /bamib/ sid” From Krakow, 1987

formation, articulatory information refers to the position and movements of different articulators from the third step of the speech production process.

2.1.2 The Organization of Articulatory Movements

From previous section, we know that speech production can be described by the motion of various articulatory gestures. Articulatory gestures here can be defined as the actions necessary to produce language, such as hand movements for sign language and mouth movements for speech. Figure 2.2 shows the movements of some articulators and their corresponding acoustic correlate for an utterance.

On the top the acoustic wave form is depicted, below it the vertical movement of three articulators. The velum, also known as soft palate, is located at the roof of the mouth and it raises and lowers to open up the nasal cavity. From the curve we can observe that the velum is lowered to open up the nasal cavity just before a nasal phone /m/ is produced. The lower lip will also begin to raise slowly before the nasal phone /m/ starts. After /m/ is finished, the jaw will also be lowered. This result shows that different articulatory gestures are “loosely synchronized”, which means that they are synchronized to some degree but not always follow strict and obvious rules. The articulatory gestures are also semi-independent. For example the jaw’s movements always lead to the corresponding movements of the lower lip.

Wihtin the framework of acoustics based speech recognition, speech is normally described by a segmental representation which consists of a discrete sequence of acoustic models, such as phones. But in fact, when we generate speech, we produce a continuous sequence of articulatory gestures which are loosely synchronized and semi-independent of each other. This motivates the use of multi-stream articulatory information in speech recognition.

2.1.3 Reduced variability through Critical Articulators

Pancun et. al. (Papcun et al., 1992) put little pellets on some articulators of a subject, and irradiate the subjects with X-rays, to obtain the vertical movements of articulatory trajectories shown in Figure 2.3. Nine different subplots represent the vertical movements of different articulator trajectories. The top row represents the tongue dorsum which refers to the back of the tongue. The second row corresponds to the trajectories of the tongue tip. And the third row refers to the trajectories of the pellets located at the lower lip. Each figure shows different trajectories for different realizations of the sounds. For example, the first column refers to the consonant /p/ and /b/, which share the same place of articulation and can be identified by the articulator "lower lip" with a very low variation. A similar situation can be observed in the other two columns. The articulators which are most crucially involved in a consonant production are called critical articulators. In the figure we see that the motion of certain critical articulators has a low variability and is less susceptible to a specific group of consonants in contrast to other articulators. This property can contribute to the advantages of using articulatory information in speech recognition systems.

2.1.4 Coarticulation Problem

Another advantage of articulatory information is that it is well suited to model the coarticulation phenomenon of speech production. Coarticulation refers to the modification of a speech sound due to the adjacent ones. This effect happens normally between a consonant and its following vowel. From the perspective of speech production, coarticulation is caused by a set of asynchronous and therefore highly overlapping articulatory gestures. Figure

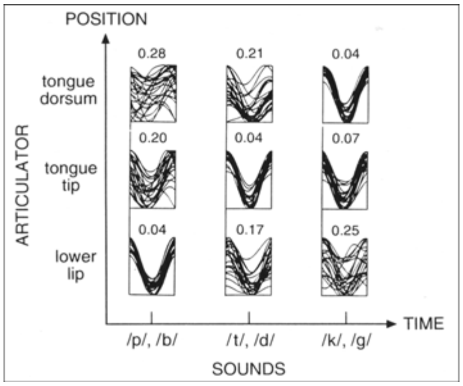


Figure 2.3: The vertical movements of articulatory trajectories from tongue dorsum, tongue tip and lower lip for labial, coronal, and velar sounds. From (Papcun et al., 1992)

2.4 shows a relative timing measurement of five articulatory gestures during the production of the English word “pan”. The results are retrieved from X-ray studies and the rectangular boxes represent the timing information of different articulators’ movements. It is clear to see a lack of synchrony between the movements of all articulators. In addition, since the movements of articulatory gestures have (depending on the phonetic context) different start and end points, some phonetic units with various pronunciations will be produced according to the coarticulation effect. For example, while the sound /n/ of English normally has an alveolar place of articulation, in the word “tenth” it is pronounced with a dental place of articulation because the following sound, /θ/, is dental.

The coarticulation effect is modeled in conventional speech recognition systems using context dependent acoustic models, such as biphones or triphones. This solution requires a large amount of models. Since training data is limited, some models will not be well trained. The use of articulatory information could help to overcome this deficiency because it directly addresses the coarticulation problem.

2.2 How to Use Articulatory Information

Using articulatory information in speech recognition systems may raise three major questions: 1) How to extract useful articulatory information? 2) How

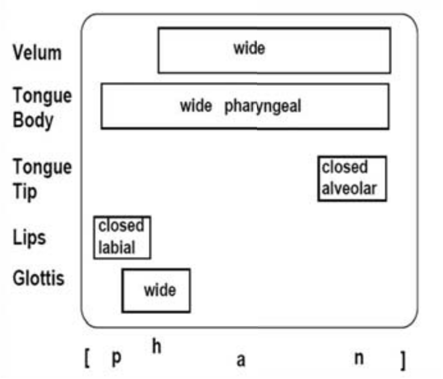


Figure 2.4: Relative timing of articulatory gestures for the production of the English word *pan*. (Browman and Goldstein, 1992)

to classify articulatory information in order to train articulatory gesture models? and 3) How to correctly map these articulatory gestures back to the phonetic units? In this section, we first describe the background of pattern classification, which is the basis of designing articulatory information based approaches. Then, different approaches to use articulatory information will be briefly presented. The details of implementation and evaluation will be given in Chapter 5 , Chapter 6 and Chapter 7.

2.2.1 Architectures for Articulatory Information based ASR

Based on the literature, we propose here a categorization of articulatory information based ASR. The approaches differ with respect to the stage where articulatory information is introduced into ASR. Thus, we distinguish four groups of systems as shown in Figure 2.5. 1) measuring articulatory information directly , 2) generating articulatory information from transcriptions, 3) extracting articulatory information by feature extraction and 4) encoding articulatory information with statistical models. In a typical ASR, signals (or raw data) are processed to extract feature vectors and to generate feature transcriptions by the feature extraction and transcription generation component respectively. The features and the transcriptions are further used to build statistical models during training. Articulatory information can be used in any one of these components to construct articulatory information

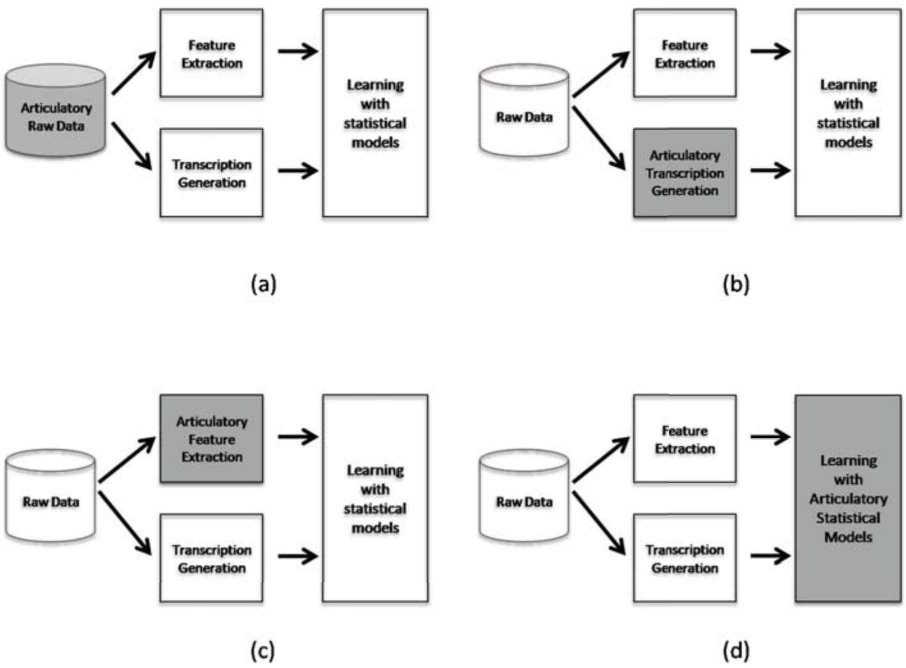

Articulatory Raw Data
Feature Extraction
Transcription Generation
Learning with statistical models
(a)
Raw Data
Feature Extraction
Articulatory Transcription Generation
Learning with statistical models
(b)
Raw Data
Articulatory Feature Extraction
Transcription Generation
Learning with statistical models
(c)
Raw Data
Feature Extraction
Transcription Generation
Learning with Articulatory Statistical Models
(d)

Figure 2.5: Four types of articulatory information based ASR systems. The components in dark point out where the articulatory information is used. (a) Articulatory raw data: True geometry of the articulators (b) Articulatory transcriptions: Decision making after several parallel phone recognizers (c) Articulatory features: Abstract representation (d) Articulatory models: Integrating relations among articulators

based ASRs.

2.2.2 Using Articulatory Raw Data Describing the True Geometry of the Articulators

Directly observing the movements of articulators is the most accurate way to obtain articulatory information (Figure 2.5(a)). The position of articulators is physically measured by the method of cineradiography, where metal pellets are attached to a subject's articulators (typically lips, tongue tip, tongue dorsum, and jaw), whose movements are then recorded by X-ray. (Papcun et al., 1992; Thomas, 1994), e.g., used X-ray microbeam data coupled with acoustic data to map acoustic parameters to articulatory trajectories using a neural network. Electromagnetic Articulography (EMA) and Electropalatography (EPG) data are other methods from which the current position of articulators can be calculated. Such data also has been used in speech recognition (Frankel and King, 2001). All these systems used early stage articulatory information. The advantage of using raw data is that it only contains the characteristics of articulators and is completely insensitive to distortions of the acoustic signal, such as added environmental noise. The disadvantage is obviously the complexity and inconvenience of the recording process, that makes it hard to use articulatory raw data in practical ASR systems. Therefore, this approach of using articulatory information is not used and discussed in the thesis.

2.2.3 Using Articulatory Transcriptions in an HMM/N-Best Decision Framework

The first system we developed using articulatory information is the articulatory transcription based AVSR. Similar to phonetic transcription, an articulatory transcription is also a symbolic representation of human speech. The difference between these two types of transcriptions consists in a phonetic transcription representing perceptually distinguishable phonetic units for each speech utterance, but an articulatory one describing different positions or states of each articulator. The relation between them can be naturally used to draw a mapping between phonetic and articulatory transcriptions. According to this mapping, we can easily change the phonetic transcriptions in acoustics based ASR into articulatory ones to build an articulatory information based ASR.

An articulatory transcription based ASR as shown in Fig.2.5(b) can be achieved using several parallel phone recognizers. These classifiers receive the same input feature vectors, namely descriptions of the acoustic and visual speech signals. However, the class labels used in the channels are different representing the position or mode of different articulators. During a separate training of all channels, classifiers are able to recognize the same segment of acoustic or visual signals into various articulatory representations. The recognized articulatory descriptions can be then integrated by a decision fusion component to find the final results. For example, an HMM/N-Best decision Framework can be used for this approach, where HMMs (Hidden Markov Models) are applied to model articulatory information in different channels and the N-Best decision schema is used to combine the results generated from HMM decoding. In the articulatory transcription approach, the reduced number of classes in each classifier has improved the recognition rate of the individual channels. However, due to the limitation of the decision fusion schema which is based on forces alignment, coarticulation cannot be modeled properly. The details of the design and implementation of an articulatory transcription based AVSR system are presented in Chapter 5.

2.2.4 Using Articulatory Features in an ANN/HMM Framework

As an alternative to the articulatory transcription, the use of articulatory features (AF) for ASR has been proposed (Kirchhoff, 1999; Amer and Berndsen, 2003; Papcun et al., 1992). Articulatory features are usually described as abstract classes, which capture relevant characteristics of the speech signal in terms of articulatory information. These classes can be used as an intermediate representation, leading to a two-stage classification procedure with a remarkable degree of robustness under noisy conditions (Kirchhoff, 1999). Moreover, compared to purely acoustic features (like Mel-Frequency Cepstral Coefficients, also MFCC), AFs can also be used to represent properties of the speech production process, such as lip rounding, tongue position, manner of articulation, etc.

The articulatory features used in Fig.2.5(c) provide an intermediate abstract representation lying between the acoustic signal preprocessing level and the subword unit probability estimation level. Most articulatory information based ASR systems in the literature belong to this type (King et al., 1998; Kanokphara and Berndsen, 2005; Kirchhoff et al., 2000). The experiments in Chapter 6 are also based on this idea. This approach applies multi-channels of ANNs (Artificial Neural Networks) to extract articulatory information, where each channel describes the status of a particular articulator. The results of these ANNs are combined to generate articulatory feature vectors, which are then used to train an HMM based speech recognizer. Several reasons make the articulatory feature based approach attractive for ASR. Firstly, it can provide a rather detailed description of coarticulation phenomena, since they are related to both the acoustic signal and the higher level of linguistic information. In particular, it is able to accommodate the kind of asynchronous transitions between subsequent segments that can be observed with articulatory movements. Secondly, compared to an acoustics based recognizer, the parallel independent articulatory feature based recognizer makes use of fewer classes, which therefore are better suited to be used in case of sparse training data. In contrast to articulatory transcription based ASR, this approach integrates articulatory information in the feature extraction component rather than in the transcription generation component. We provide more details of the articulatory feature based AVSRs in Chapter 6.

2.2.5 Encoding Articulatory Information By Means Of DBN Models

A DBN (Dynamic Bayesian Network) is a graphical model that represents sequences of variables. A time-series of symbols (like speech) is a typical sequence which can be modelled by means of a DBN. HMMs are a special case of DBNs. DBN assumes that particular dependencies are relevant. The hidden state is represented by a single discrete random variable. In the more general case of DBNs, however, the hidden state is represented by a set of random variables, each of which can be discrete or continuous. It helps us to model causality in a more natural way in order to build more flexible and meaningful classifiers for asynchronous movement of articulators.

In contrast to the previous approaches (using articulatory transcriptions and using articulatory features), encoding articulatory information during training lends itself to integrate articulatory information asynchronously. Audio and visual articulatory information are combined on different phonetic levels (e.g. Word, Phone, Subphone, etc.). In this thesis we use DBNs with different topologies to build a set of articulatory information based AVSR systems.

Here we briefly introduce some types of DBN based AVSR structures. These types differ from each other by their model topology. We use the term “channel” to denote a DBN structure with different phonetic levels. Firstly, a single channel DBN-based AVSR structure refers to a DBN structure with only one phonetic channel. This single channel is designed with four levels, namely word level, phone level, subphone level and observation level. Using this structure, audio and visual articulatory information can only be integrated at the observation level. That is, audio and visual features are combined at each state. Secondly, a two channel DBN-based AVSR with audio and visual channels uses an additional visual channel to model the visual information. The articulatory information can be combined on different levels. Thirdly, a three channel DBN-based AVSR with articulatory channels is designed based on pronunciation rules. Each channel in the structure models the movements of a particular articulator. The audio articulatory channels model the audio information and the visual articulatory channels use the visual information. All the information can be combined at the word level. The details of DBN and DBN based AVSRs are introduced in the Chapter 7.

2.3 Information Fusion and Synchronicity in Articulatory Information Based AVSR

In the area of multimodal interaction, modelling the temporal relationships between the input channels is an essential aspect (Luettin et al., 2001). Especially in the interaction between the acoustic and the visual modality plenty of research has been done which addresses the issue of synchrony and asynchrony. The necessity to synchronize parallel streams of information not only arises for the coupling of the acoustic and the visual modality, but also among parallel channels within one modality.

This section only introduces the concepts of information fusion and synchronicity for articulatory information based AVSR. The details of the different systems are described in Chapter 5-7 respectively.

2.3.1 Information Fusion for Speech Recognition

Information fusion is an instance of the general classifier combination problem. In articulation based AVSR information is described on two levels. On the one hand, the acoustic and the visual modality both provide information related to speech classes, such as phone and viseme units. The two channels of observation can either be used to train two independent single modality classifiers separately, or can be combined to train a bimodal speech classifier. On the other hand, the theory of speech production naturally separates the movements of different articulators into multiple parallel articulatory channels. Since each articulatory channel represents only some part of the speech characteristics, its observation can not be used to train a full speech classifier alone. Multiple channels of articulatory information must be properly fused in order to provide useful cues for a speech recognition system.

Table 2.1: Information Fusion in Articulatory Information based AVSR.

Fusion Methods	Fusion Procedure	Used articulatory information
Feature Fusion	in feature extraction	articulatory feature vectors
Model Fusion	during training	articulatory feature models
Decision Fusion	during decoding	articulatory feature transcription

Various information fusion techniques have been proposed in the literature on AVSR and articulatory information based ASR. These fusion algorithms differ in their basic designs. However, both audio-visual fusion and articulatory information fusion techniques can be roughly categorized into three groups as shown in Table 2.1. Feature fusion combines the multichannel information at the level of observation. It happens before training the recognition model. Feature vectors from different channels can be simply concatenated into a single observation. In contrast, model fusion happens within the training procedure itself. Multiple channels of information are combined at the subphone, phone or word level. In a single classifier, a fused conditional probability distribution is computed from all channels of observation at runtime. Finally, decision fusion employs multiple channels of single modality classifiers to train their independent speech classes. Their results are then combined into a final decision by means of a linear combination of scores or a more complex decision taking strategy.

2.3.2 Synchrony and Asynchrony

From the perspective of speech perception, it has been found that an auditory stimulus needs to be delayed to be perceptually aligned with a visual stimulus (Serences et al., 2009). The temporal offset arises because the acoustic transfer between the outer and inner ears is a relatively direct process, in contrast to the visual transfer in the retina which passes several cascading neurochemical stages (Alais et al., 2005). However, it is difficult to determine the timing for correlating audio and visual signals exactly. For technical solutions in AVSR, finding better possibilities than HMM to model the temporal differences between both channels is still a challenging task.

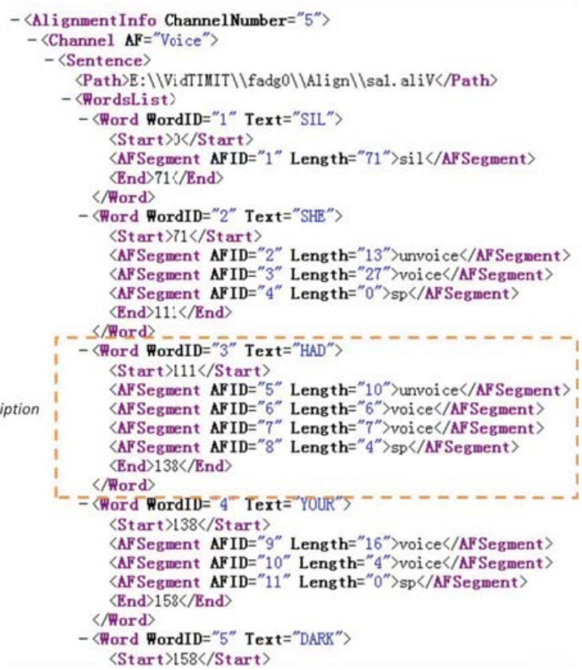
From the perspective of speech production, speech units are composed of movements of "loosely" aligned hidden articulators. Here "loosely" means that the movements of all articulators happen simultaneously, but their starting and ending time is not always the same. That is, although all articulators have to move in order to generate a word, the temporal variations among them may lead to different pronunciations for the same word.

According to findings from speech perception and speech production, we are motivated to investigate the synchronicity issues in articulation information based AVSRs. Two concepts are introduced here before we start to analyze the synchronicity issues in the following chapters. On the one hand, synchronizing information from multiple channels means that this information has the same starting and ending time for a particular speech unit in each information channel. On the other hand, asynchronous information from multiple channels means that the information does not have the same starting and ending time for a particular speech unit. A typical asynchronous case is the fusion of acoustic and visual speech signals. The visual signal for a phone, for example, happens to always precede the corresponding acoustic signal. The main question to investigate this issue is about where do we find synchronicity and asynchronicity. All the possible categories are various phonetic abstractions, such as, words, phones, subphones, etc. They represent the continuous reality of the speech signal.

2.3.3 Temporal Alignment of Articulatory Features

In the thesis we focus on the synchronicity issues of the articulatory information. Usually a word can be decomposed into several articulatory values for each feature. For example, the word "had" can be transcribed as [unvoiced voiced voiced] in the "Voicing" feature and as [nil flat nil] in the "Rounding" feature. Without temporal information attached to the values, however it is still difficult to analyze the relationships among articulatory features. We applied a forced alignment procedure to automatically labeling the boundaries between two adjacent articulatory values.

In speech recognition, search engine is given an exact transcription of spoken



```
-<AlignmentInfo ChannelNumber="5">
  -<Channel AF="Voice">
    -<Sentence>
       <Path>E:\\VidTIMIT\\fadg0\\Align\\sa1.aliV</Path>
      -<WordsList>
        -<Word WordID="1" Text="SIL">
           <Start>0</Start>
           <AFSegment AFID="1" Length="71">sil</AFSegment>
           <End>71</End>
         </Word>
        -<Word WordID="2" Text="SHE">
           <Start>71</Start>
           <AFSegment AFID="2" Length="13">unvoice</AFSegment>
           <AFSegment AFID="3" Length="27">voice</AFSegment>
           <AFSegment AFID="4" Length="0">sp</AFSegment>
           <End>111</End>
         </Word>
        -<Word WordID="3" Text="HAD">
           <Start>111</Start>
           <AFSegment AFID="5" Length="10">unvoice</AFSegment>
           <AFSegment AFID="6" Length="6">voice</AFSegment>
           <AFSegment AFID="7" Length="7">voice</AFSegment>
           <AFSegment AFID="8" Length="4">sp</AFSegment>
           <End>138</End>
         </Word>
        -<Word WordID="4" Text="YOUR">
           <Start>138</Start>
           <AFSegment AFID="9" Length="16">voice</AFSegment>
           <AFSegment AFID="10" Length="4">voice</AFSegment>
           <AFSegment AFID="11" Length="0">sp</AFSegment>
           <End>158</End>
         </Word>
        -<Word WordID="5" Text="DARK">
           <Start>158</Start>
```

The word "HAD" transcription of "Voicing" feature

Figure 2.6: Alignment of AF Transcriptions in XML.

utterance and a sequence of speech data. The ASR system then aligns the transcribed data with the speech data in order to find the time segments where the speech data best fit their corresponding spoken words. In our case, we train parallel classifiers using the acoustic signal and its articulatory transcription without time information using the forced alignment procedure. For decoding the same training data is used as testing data to force the classifiers to align the models and the transcriptions. The result of the forced alignment for an individual classifier is an articulatory transcription with time information. For example, we can represent this time information using the XML format as shown in Figure 2.6.

In the "Voicing" feature, the decomposed word "HAD" is annotated with its starting and ending time [10 6 7] and the length information of all articulatory values [unvoiced voice voice]. Together with time information from other features, the degree of synchronization can be measured.

A graphical user interface (GUI) as shown in Figure 2.7 has been developed in order to observe the asynchronous features. By combining all transcrip-

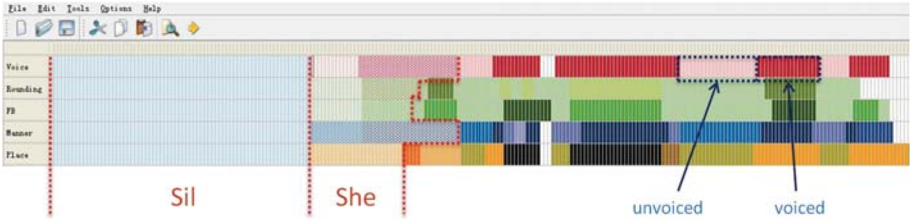


Figure 2.7: An example of AF transcription with time information.

tions obtained by forced alignment we present the time information in an $m \times n$ matrix (see Figure 2.7). Columns n correspond to time where each interval represents a frame. Rows m in the matrix represent the individual articulatory features. Differently colored segments in a row encode different values of an articulatory feature. For example, the two different red colors in the figure represent the “unvoiced” and the “voiced” values of “Voicing” feature. In the GUI a cell can be selected to indicate the value of an articulatory feature. Furthermore, by clicking on a cell, all articulatory features from the current word will be also selected and displayed in different color. As shown in the Figure 2.7, the word “SHE” is selected. All the articulatory features within the word “SHE” are displayed with a shadow to indicate the word boundary. In the later chapters we are going to analyze the details of synchronization with respect to the results for different architectures.

Chapter 3

Features of Audio Visual Speech

3.1 Audio Feature Extraction

The first step of designing an automatic speech recognition system is always to find a proper way to extract speech features from raw speech signals. The term of "(speech) features", a parametric description, refers to a machine-internal representation with certain characteristics of the speech signal. The selection of the best representation of acoustic data is an important task in the design of any speech recognition system. The audio feature extraction component transforms speech to a vector of features which are suitable for further processing. Obviously, a good system recognition performance relies greatly on an feature extraction procedure. Through more than 50 years of ASR research, many different speech feature representations have been suggested and evaluated. In particular, a set of audio processing techniques based on characteristics of the human auditory system has been designed for extracting acoustic features. For example, from the psychoacoustics point of view, which is the study of subjective human perception of sounds, the Mel scale or the Bark scale simulates the non-linear frequency resolution of the human hearing system. These filter banks are approximately logarithmic in frequency at the high-frequency end, but nearly linear at the low-frequency end. A cepstral transformation is normally used in order to avoid highly correlated filterbank amplitudes.

The aims of front-end processing are twofold. On the one hand, the parameters/features should capture the salient aspects of the speech signal. These features should be perceptually as relevant to the sounds as possible. In particular, they should comprise the spectral dynamics, i.e. the change of the spectra over time. On the other hand, the features should be robust in the sense that the recognition quality should not be affected by distortions that can appear at the input, due to e.g. environmental characteristics and/or the transmission medium. Also, a general ASR-application should be able to recognize speech from different persons.

In the following we introduce some acoustic features, which are most widely used in ASR systems. Both the mel-cepstral and the PLP analysis provide a feature representation which corresponds to a smoothed short-term spectrum that has been compressed and equalized similar to human hearing. Before we explain the differences of these approaches, their common basic techniques are described in this section.

3.1.1 Common Techniques

Windowing

Windowing refers to a process of multiplying a given signal by a window function. A window function is a function that is zero-valued outside of some chosen interval. Given a signal function and a window function, the product is also zero-valued outside the interval. The ”view” through the window, onto the values inside the interval represents the transferred signal affected by a certain window function. For speech processing, a typical length of a frame is 20-25 ms. Usually the frames are set to overlap so that their centers lie only 10 ms apart. The window function is chosen in a way that the values near the edges approach zero. This prevents discontinuities at the edges which would negatively affect the result of further processing. A popular function is the Hamming window which is represented by the equation 3.1, where N is the length of the window.

$$w(n) = 0.54 - 0.46cos(\pi n/N) \tag{3.1}$$

Fast Fourier Transform

The Fast Fourier Transform (FFT) transfers signals from time domain into frequency domain. It converts the signal information to a magnitude and phase component of each frequency. This information can be represented as a 2-dimensional vector or a complex number, or as magnitude and phase. Equation 3.2 represents the FFT of each time window for discrete time signal $x(n)$ with length N,

$$X(k) = \sum_{n=0}^{N-1} w(n)x(n)exp(-j2\pi kn/N) \tag{3.2}$$

for $k = 0, 1, ..., N-1$, where k corresponds to the frequency $f(k) = kf_s/N$, f_s is the sampling frequency in Hertz and $w(n)$ is the Hamming window in equation 3.1.

3.1.2 MFCC Features

Mel-Frequency Cepstral Coefficients (MFCCs) are short-term spectral features based on a linear cosine transform of a log power spectrum on a non-linear mel scale of frequency. MFCCs have been proposed in 1980 by (Davis and Mermelstein, 1980) and are still widely used until now in the area of automatic speech recognition, automatic speaker identification and music information retrieval. Figure 3.1 depicts the procedure of extracting MFCCs.

Firstly, audio signals are preemphasized to improve the overall signal-to-noise ratio by increasing the magnitude of some (usually higher) frequencies. Preemphasizing prevents some adverse effects, such as attenuation distortion, in the subsequent phases of MFCC processing. Furthermore, it approximates the unequal sensitivity of human hearing at different frequencies.

The procedure of dividing audio signals into frames is implemented in the windowing phase. Then, the windowed signals are subjected to a Fast Fourier Transform (FFT) for spectral analysis.

The power spectrum is warped according to the Mel scale in order to model the frequency resolution of the human ear. For that purpose the spectrum is segmented into a number of critical bands by means of a filter bank. Mel

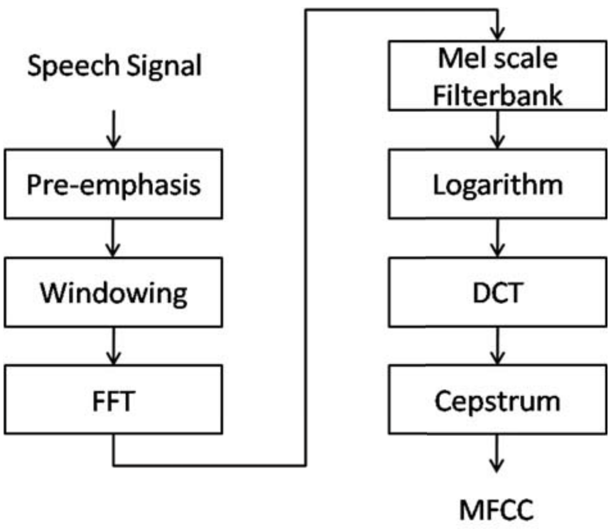


Figure 3.1: MFCC feature extraction front-end.

scale was developed on the basis of human auditory perception experiments (Stevens and Volkmann, 1940) and is approximately linear below 1 kHz and logarithmic above (Figure 3.2)

.

As shown in figure 3.3, the Mel filter bank is a collection of overlapping triangular filters. The Mel spectrum is computed by multiplying the Power Spectrum by each of these Mel Weighting filters and integrating the result. Furthermore, the spectral amplitudes are compressed by a logarithmic function.

$$X'(m) = ln\left(\sum_{k=0}^{N-1} \mid X(k) \mid \cdot H(k,m)\right) \tag{3.3}$$

Finally, the cepstrum is computed by means of the Discrete Cosine Transform (DCT):

$$c(l) = \sum_{m=1}^{M} X'(m)cos(l\frac{\pi}{M}(m-\frac{1}{2})) \tag{3.4}$$

for $l = 1, 2, \ldots, M$, where $c(l)$ is the lth MFCC.

On the one hand, the MFCCs are good at modeling the quasi-logarithmic frequency resolution of the human ear. But on the other hand, the logarithmic

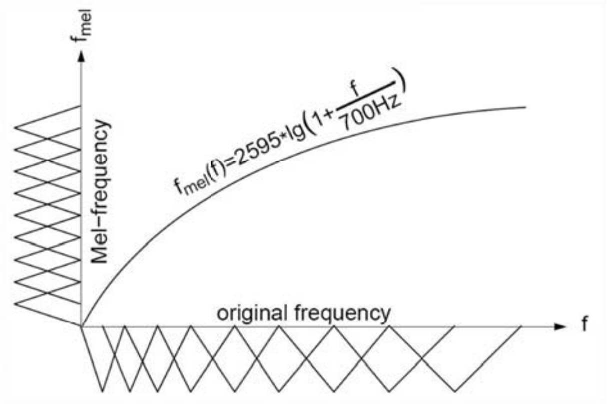


Figure 3.2: Mel frequency warping and the filterbank. The filters are either uniformly distributed at the Mel warped spectrum, or non uniformly at the original spectrum. In the latter case, they should be asymmetric as well.

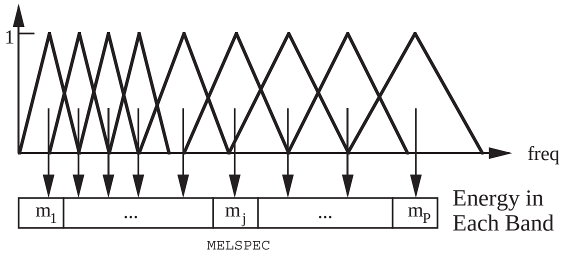


Figure 3.3: Mel frequency filterbank.

operation will also amplify even a small background noise level. Therefore, the MFCCs' sensitivity to noise negatively affects the performance of the entire ASR system.

3.1.3 RASTA-PLP Features

RASTA-PLP is an acronym for RelAtive SpecTrAl transform - Perceptual Linear Prediction, which was originally proposed in (Hermansky, 1990) and (Hermansky and Morgan, 1994). PLP warps the spectra in order to adapt the features to different speakers while keeping useful speech information. The design of PLP is more consistent with human hearing because it considers the nonequal sensitivity of human hearing at different frequencies. Speech loudness is assumed to be directly related to the speech quality. As a set of filtering approaches, RASTA methodology separates nonlinguistic spectral components (e.g. noise) from the spectral components generated by the movement of the vocal tract. The fact, that the rates of changes of these components are different, motivates us use RASTA to retrieve more robust speech features. The idea is to suppress the spectral components that change more slowly or quickly in order to focus on the speech information with a typical range of changes. The detailed methods of PLP and RASTA-PLP are introduced in this section.

PLP analysis is based on linear prediction (LP) analysis but additionally includes auditory properties through the computation of a compressed filter bank spectrum. As shown in Figure 3.4, the procedure of PLP analysis is similar to the MFCC analysis. In the component for spectral analysis, the input speech is segmented and weighted by the Hamming window as introduced in Equation 3.1. The same FFT as shown in Equation 3.2 is used here to transfer information from time domain to frequency domain.

In contrast to a Mel scale, the spectrum is warped along its frequency axis into the Bark frequency (Zwicker, 1961). The resulting warped power spectrum is then convolved with a simulated critical-band masking curve $H(\omega)$. Different to the triangular filters in MFCC, the filters in PLP are trapezoidal in shape (Hermansky, 1990). This particular shape of the critical-band is a crude approximation of the asymmetric properties of auditory filters. The discrete convolution of $H(\omega)$ with the power spectrum $P(\omega)$ yields samples of the critical-band power spectrum

$$S(\Omega_i) = \sum_{\Omega=-1.3}^{2.5} P(\Omega - \Omega_i)H(\Omega) \tag{3.5}$$

After critical-band spectral resolution, the pre-emphasis block is used to simulate an equal-loudness curve for the samples $S[\Omega(\omega)]$.

$$P[\Omega(\omega)] = E(\omega)S[\Omega(\omega)] \tag{3.6}$$

The function $E(\omega)$ is an approximation of the nonequal sensitivity of human hearing at different frequencies and simulates the sensitivity of hearing at a level of about 40-dB.

$$E(\omega) = [(\omega^2+56.8\times10^6)\omega^4]/[(\omega^2+6.3\times10^6)^2\times(\omega^2+0.38\times10^9)(\omega^6+9.58\times10^{26})] \tag{3.7}$$

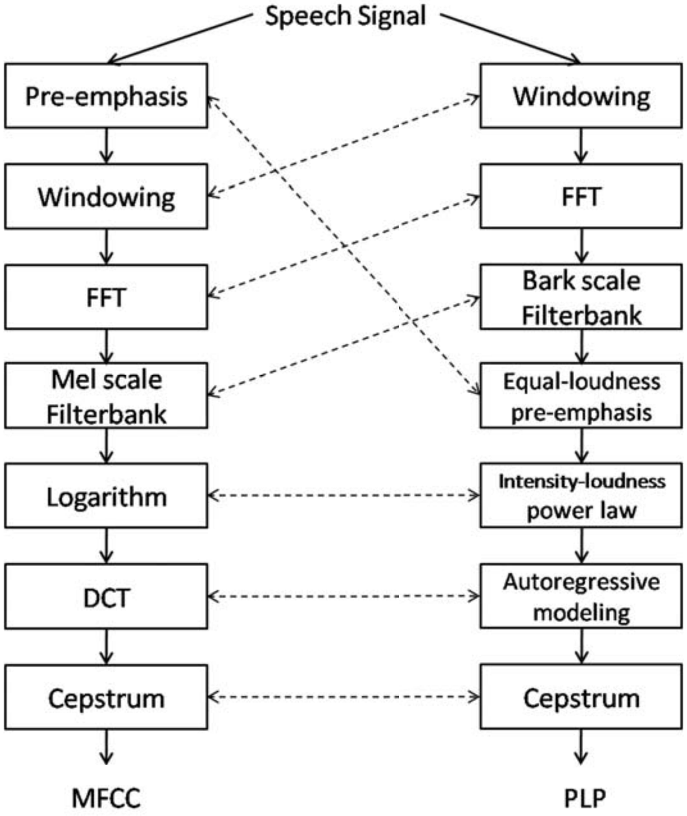


Figure 3.4: Audio feature extraction front-end comparison between MFCC and PLP. Similar processing steps between two methods are labeled by dashed lines between blocks.

According to the power law of hearing (Stevens, 1957), a cubic-root amplitude compression (equation 3.8) is then taken for simulating the nonlinear relationship between the intensity of sound and its perceived loudness.

$$I(\Omega) = P(\Omega)^{\frac{1}{3}} \tag{3.8}$$

In the final operation of PLP analysis, after inverse (discrete) Fourier transform (yielding autocorrelation coefficients), the autoregressive model is calculated to smooth out details from the auditory spectrum. The autoregressive coefficients are usually transformed into orthogonal parameters, such as cepstral coefficients.

PLP analysis was found to be vulnerable to linear spectral distortions. Previous research on human speech indicates that human listeners do not seem to be sensitive to slow changes in frequency. Furthermore, steady background noise does not impair human speech communication. These facts can be explained by the relative insensitivity of human hearing to slowly varying stimuli (Green, 1976). To alleviate this problem, PLP processing is often combined with a RASTA filtering method.

In order to make speech analysis less sensitive to the slowly changing nonlinguistic spectral components, (Hermansky and Morgan, 1994) proposed to replace the conventional critical-band short-term spectrum in PLP speech analysis with a special spectral estimate. In this spectral estimate, each frequency channel is defined by a filter which transforms the signal spectral at the zero frequency into spectral zero. Since this band-pass filter suppresses all the constant and slowly changing spectral components in each frequency channel, this spectral estimate yields the low sensitivity to slow variations in the short-term spectrum (Hermansky et al., 1991).

The steps of RASTA-PLP are shown as Figure 3.5. Raw speech signals are firstly analyzed to get the critical-band spectrum as in the conventional PLP. The logarithm of the critical-band is estimated by its temporal derivative using a regression line. A static nonlinear transformation is used to convert the environmental noise into additive components. Similar to the conventional PLP, RASTA-PLP adds the equal loudness curve and simulates the power law of hearing. Then this relative log spectrum is computed by the inverse

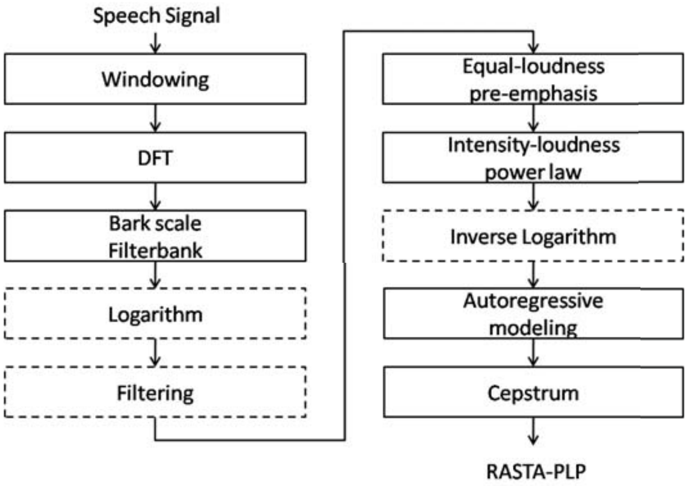


Figure 3.5: RASTA-PLP feature extraction block diagram. Dashed line blocks are RASTA processing steps.

logarithm (exponential function) to get a relative auditory spectrum. Finally, an all-pole model of this spectrum is computed as in the conventional PLP technique.

3.2 Visual Feature Extraction

In contrast to the auditory ones, visual features are obtained from speech related video sequences. The first main difficulty in the area of audio visual speech recognition is the visual front-end design. To obtain useful visual features, three different approaches are available: the appearance-based, the shape-based and the hybrid method. In general, the raw video data of the speaker are first preprocessed to detect and extract the region of interest (ROI), namely the mouth region. Then, different algorithms can be employed for converting the ROI into feature vectors for further processing. Figure 3.6 presents a general block diagram of visual feature extraction. In this section, we first describe the common techniques in the front end of visual speech processing. Then, the differences of three visual speech feature extraction methods are presented.

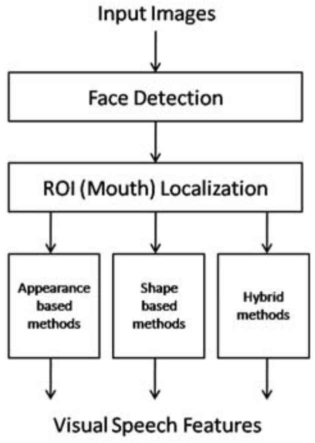


Figure 3.6: Visual Speech Extraction Front Ends

3.2.1 Common Techniques

Face Detection

Like other object-class detection methods, face detection aims at finding the locations and sizes of all face candidates in an image, where a face concept is already given. Face detection algorithms can be categorized into frontal human face detection and multi-view face detection. In the second case, the face objects are rotated along either an horizontal axis, or vertical one, or both of them. Considering the complexity of various processing techniques, face detection can be categorized into two groups. Some use traditional image processing techniques, such as color segmentation, edge detection, image thresholding, template matching, or motion information (Yang et al., 2002). Other methods use statistical modeling techniques, such as neural networks (Rowley et al., 1996), Gaussian mixture models (Poggio and Sung, 1995), or support vector machines (Osuna et al., 1997). In the following, we describe the frontal face detection algorithm proposed by Viola and Jones (Viola and Jones, 2001) as a face detector example.

The Viola-Jones detector is a strong, binary classifier built out of several weak detectors. Each weak detector is a simple binary classifier. During the learning stage, a cascade of weak detectors is trained to gain the desired accuracy rate using Adaboost (Freund and Schapire, 1997). Here the term cascade implies that each successive classifier is trained only on those

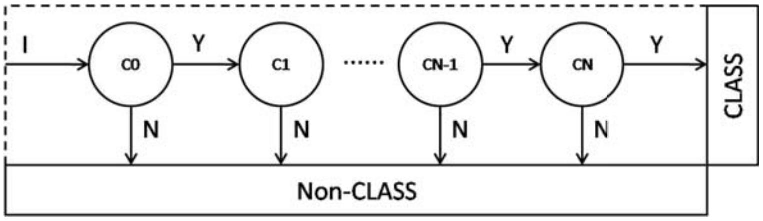


Figure 3.7: Cascade Classifiers. A series of classifiers are applied to every rectangular patch. A positive result from the first classifier triggers the evaluation of a second classifier which has been adjusted to achieve better detection rates and requires additional computation. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the example.

examples which pass through the preceding classifiers. Such a cascaded architecture (as shown in Figure 3.7) makes the viola Jones detector fast enough to run in real-time. To detect the face or other objects, the original image is partitioned into several rectangular patches, each of which is submitted to the cascade. If a certain rectangular patch passes through all of the cascade stages, then it is classified as “positiv”, which means the patch is the a valid face candidate. For being detected in real images, a face can have an arbitrary size compared to the patch which is always of a fixed size. An image pyramid is calculated in order to detect faces at multiple scales. A fixed size patch is moved through each image in the pyramid. Given a pyramid of images as inputs, like most object detection systems, the Viola-Jones detector starts the classification process at a base scale and scans the inputs at many scales.

The basic weak classifiers are based on very simple visual features, also named as haar-like features. As shown in figure 3.8, the Viola-Jones detector uses three types of features (two-rectangle feature, three-rectangle feature and four-rectangle feature). A two-rectangle feature describes the difference between the sum of the pixels within two adjacent rectangular regions. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. The value of a four-rectangle feature is the difference between diagonal pairs of rectangles. The differences between different features in a particular type are the location and size of the white and black area.

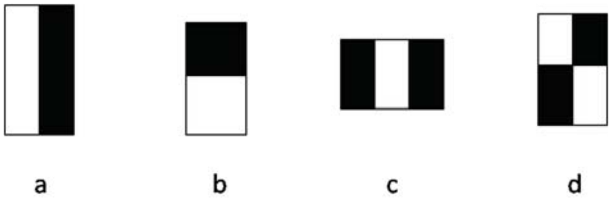


Figure 3.8: Example rectangle haar-like features. (a) and (b) are two-rectangle features, (c) is a three rectangle feature and (d) is a four-rectangle feature. The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the black rectangles.

To reduce the effort of computation, the Viola-Jones algorithm uses an "integral image" as an intermediate representation. An integral image $ii(x, y)$ at location (x, y) contains the sum of the pixels above and to the left of (x, y),

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{3.9}$$

As shown in Figure 3.9(a), any rectangular sum can be computed by means of four reference points from the integral image. For example, in order to calculate the sum of the pixels within rectangle D, we need values from four reference points. Assuming that the value of the integral image at location 1 is the sum of the pixels in rectangle A, the value at location 2 is $A + B$, at location 3 is $A + C$, at location 4 is $A + B + C + D$, then we can compute the sum within D as,

$$v_4 + v_1 - (v_2 + v_3) \tag{3.10}$$

where v_n represents the value of the integral image at the location n. For computing the values for the different types of Haar-like features in Figure 3.8, we need to calculate the difference between the number of pixels in the black and white rectangles. With the idea of integral image, the two-rectangle features need six reference points, eight points in the case of the three-rectangle features, and the four-rectangle features need nine reference points.

In the Viola-Jones algorithm, the weak classifiers are designed to select the single rectangle feature which best separates the positive and negative image examples. For each feature, the weak classifier determines the optimal

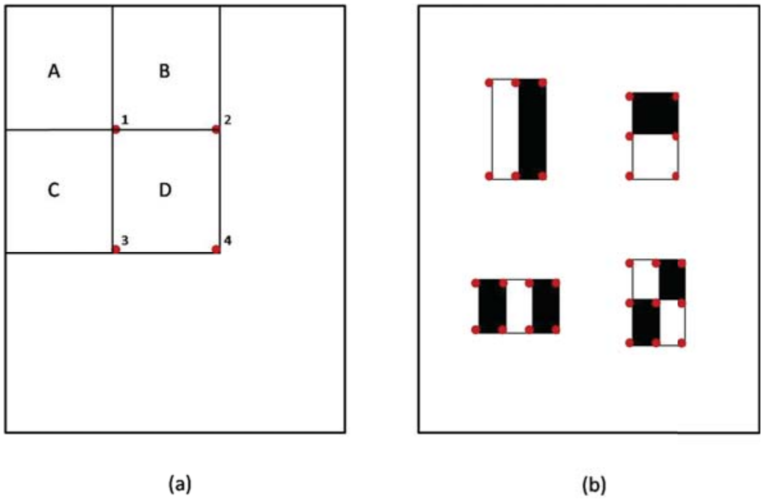


Figure 3.9: Integral Image. Red points refer to the reference points of particular rectangles.

threshold classification function, such that the number of misclassified image examples is minimal. A weak classifier $h_j(x)$ thus consists of a feature f_j, a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 & \textbf{if} p_j f_j(x) < p_j \theta_j \\ 0 & \textbf{otherwise} \end{cases} \tag{3.11}$$

ROI Localization

Since Viola and Jones describe a general object-class detection algorithm, it is in principle suited for finding any ROI according to the requirements of the application. In AVSR, the ROI typically is a rectangle containing the mouth, and possibly including larger parts of the lower face, such as the jaw, or the entire face. After finding a correct face candidate from an image, a similar approach can be used to detect the lip region. As an alternative to the Viola-Jones algorithm, many techniques of varying complexity can be used to locate these ROIs. For example, (Senior, 1999) used the algorithm of Fisher discriminant and Distance From Feature Space (DFFS) to locate features in the face. The features refer to various points on the face which indicate the location of different ROIs, such as, eyes, nose, eyebrow and mouth.

Principle Components Analysis

Principal Component Analysis (PCA), also named as discrete Karhunen Loeve transform (KLT), is a mathematical method to transform a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. Each principal component is a linear combination of the original variables. All the principal components are orthogonal to each other, so that there is no redundant information. In various visual feature extraction methods, PCA is always applied to reduce the raw visual information with an extremely high dimensionality into a feature representation with a lower dimensionality.

3.2.2 Appearance Based Visual Features

Appearance-based methods make use of all pixel level intensity and color values within the ROI as useful lipreading information. In order to capture also dynamic speech information, adjacent frame ROIs are typically considered. Finally, transformations, like DCT, PCA, etc., can be applied to reduce the extremely high dimensionality of ROI information.

For example, as shown in Figure 3.10, the final visual feature vectors can be extracted by means of these processing. For every video frame $V_t(m,n)$ at time t, pixel values from a $M \times N$-rectangular ROI are placed in the vector

$$x_t^{(ROI)} \leftarrow \{V_t(m,n) : m_t - \lfloor M/2 \rfloor \leq m < m_t + \lceil M/2 \rceil, n_t - \lfloor N/2 \rfloor \leq n < n_t + \lceil N/2 \rceil\} \tag{3.12}$$

where (m_t, n_t) refers to the speaker's mouth center. The vector length is $d^{(ROI)} = MN$ which is normally too large for the subsequent classification steps. Therefore the architecture in 3.10 uses three matrices, $\mathbf{P^{(DCT)}}$, $\mathbf{P^{(LDA)}}$, and $\mathbf{P^{(MLLT)}}$ to obtain a compact visual feature vector with a dimension of $D << d^{(ROI)}$. Similar to the PCA, the matrix $\mathbf{P^{(DCT)}}$ applies a *discrete cosine transform* (DCT) for linearly transforming the image data in order to reduce the dimensionality. The *linear discriminant analysis* matrix $\mathbf{P^{(LDA)}}$ is then used for ROI classification into the set of speech classes of interest. Finally, a *maximum likelihood linear transform* matrix $\mathbf{P^{(MLLT)}}$ is applied to maximize the observation likelihood in the original feature space.

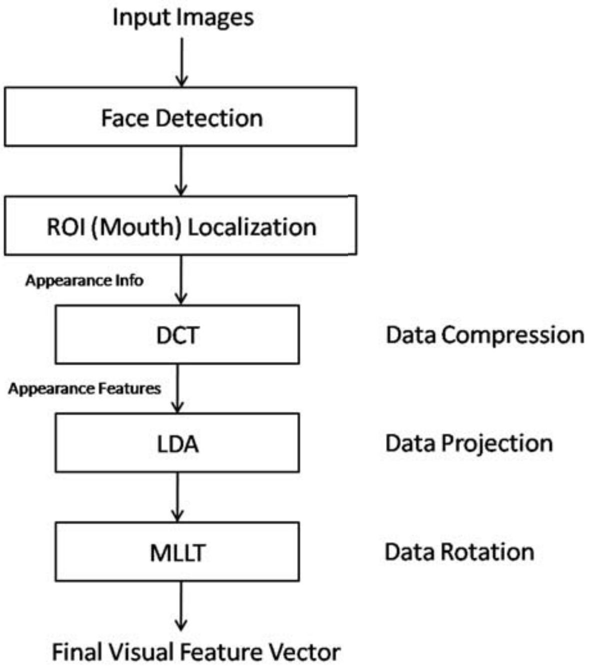


Figure 3.10: Appearance based method of visual front ends from (Potamianos and Neti, 2001)

3.2.3 Shape Based Visual Features

Compared to appearance-based methods, shape-based methods assume that most visual speech information can be expressed by the shape of the speaker's lips, or more generally, by the face contours, which includes also jaw and cheek shapes. Features extracted with shape based methods can be categorized into two groups: lip geometric features, and lip model features.

Lip geometric features

Lip geometric features represent a number of high level features of the lip contour, such as the contour height, width, perimeter, as well as the area contained within the contour. Figure 3.11 shows three types of lip geometric features. A large number of lipreading systems makes use of this feature type either alone or in combination with other features (Luettin et al., 1996) and (Dupont and Luettin, 2000).

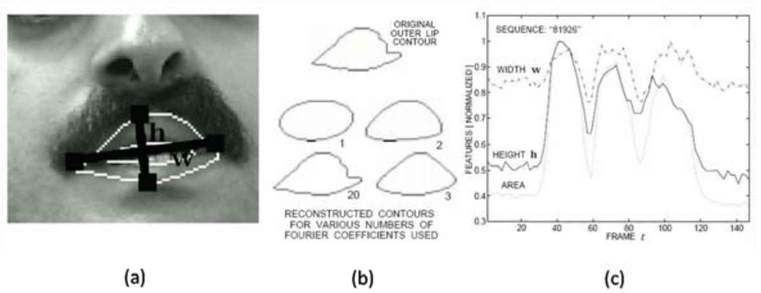


Figure 3.11: Lip geometric features. (a) illustrates the width (w) and height (h) of an outer lip. The upper part of (b) shows an original outer lip contour, and the others are reconstructions of an estimated outer lip contour from different sets of its Fourier coefficients. (c) presents three geometric visual features, tracked over a spoken utterance 81926. Lip contours are estimated as in (Potamianos et al., 1998). This figure is reprinted from (Graf et al., 1997)

Lip model features

In addition to lip geometric features, lip model features have also been widely used (Basu et al., 1998), (Chiou and Hwang, 1997) and (Cootes et al., 1995). The idea is an extension of the lip contour tracking process. Lip contour tracking applications need to obtain the shape of the mouth robustly and efficiently. The visual features are chosen from the parameters of lip- or face-contour models. However, the high variability of the mouth shapes during speech and other expressions, as well as the variability of the lip color and skin color between people makes the lip contour tracking task difficult and complex. Hence, a number of algorithms have been proposed to retrieve lip contour information from a given mouth region image. One of the most common methods for contour extraction is to use active contours, also known as snakes (Chiou and Hwang, 1997), which can provide high deformability and produce a good solution for object contour extraction.

A snake is a curve of the object contour represented by a set of control points. The algorithm attempts to minimize the energy associated to the current contour as a sum of an internal and external energy. The internal energy is formed by the snake configuration. It is assumed to be minimal when the snake has a relevant shape to the shape of the target object. The external energy is formed by external forces affecting the snake. It is assumed to be minimal when the snake is at the object boundary position. By iteratively updating the control point coordinates, the output of the energy function gradually converges, and an object contour is then retrieved. This model is highly popular in computer vision, and led to several developments in 2D and 3D.

Another example for lip contour tracking is the Active Shape Model (ASM) (Cootes et al., 1995), which is an iterative fitting algorithm using a statistical shape model which is also known as point distribution model (PDM). ASM provides not only a way to obtain lip contours but also a method for converting lips information into feature vectors. It represents a discrete version of the snake approach taking advantage of the PDM to restrict the shape range to an explicit domain learned from a training set. PDM is obtained from the statistics of hand labeled training data. The distribution of sets of "landmark" points represents significant positions of an arbitrary object. In most lipreading applications, the PDM describes a reduced space of valid lip shapes. The landmark points in this space are general representations of a particular lip shape which can be directly used as visual speech features for lipreading.

Landmark points in the training images can be either hand labeled or automatically generated (Hill et al., 1992). These points are used as input for calculating a point distribution model. Given all training images, each PDM is represented by the coordinates of its own labeled landmark points. In Figure 3.12, a training image is labeled with 44 points to describe the inner and outer lip contour (24 points on the outer and 20 on the inner contour) (Matthews et al., 2002).

If an ASM is labeled by a number of K contour points, then it can be described as a $2K$ dimensional "shape" vector.

$$\mathbf{x_S} = (x_1, y_1, x_2, y_2, ..., x_K, y_K)^T \tag{3.13}$$

Given a set of such vectors as obtained from the training data, the mean shape $\mathbf{X_s}$ can be calculated and the optimal orthogonal linear transform P_{PCA} can be determined by using PCA. Two similar shapes $\mathbf{x_1}$ and $\mathbf{x_2}$ can

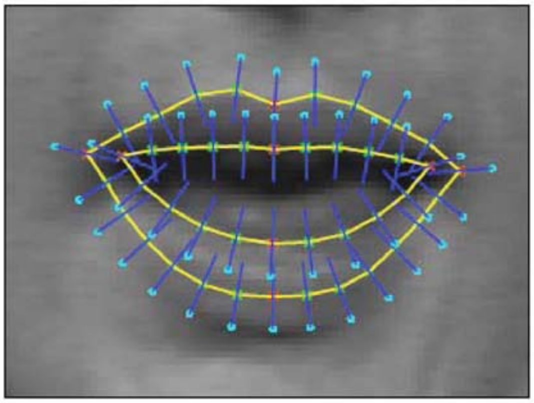


Figure 3.12: Inner and outer lip contour model (Matthews et al., 2002).

be aligned by minimizing,

$$E = (\mathbf{x_1} - M(s,\theta)[\mathbf{x_2}] - \mathbf{t})^T \mathbf{W} (\mathbf{x_1} - M(s,\theta)[\mathbf{x_2}] - \mathbf{t}) \tag{3.14}$$

where $M(s,\theta)$ is the pose transform for scale s and rotation θ. $\mathbf{t}$ represents the translation between two shapes. $\mathbf{W}$ is a diagonal weight matrix for each point with weights that are inversely proportional to the variance of each point. An iterative algorithm (Cootes et al., 1995) is used to compute the optimal alignment.

Given a tracked lip contour, the extracted visual features will be $\mathbf{y} = \mathbf{P^{(PCA)}x}$. This allows valid lip shapes to be represented in a compact, statistically derived shape space. Also, the dimensionality of landmark points which are highly correlated, is reduced.

3.2.4 Hybrid Visual Features

Hybrid methods combine appearance and shape-based approaches. Features extracted by shape-based and appearance-based methods are typically concatenated. For example, (Chen, 2001) used a combination of geometric lip features and the PCA projection of a subset of pixels contained within the mouth to describe the visual speech information. (Luettin et al., 1996) and (Dupont and Luettin, 2000) combine ASM features with the PCA based features which are extracted from ROI images containing the lip contour. A more elaborate approach is used by (Cootes et al., 2001), who creates the Active Appearance Model (AAM) which is a single model of face shape and appearance. An AAM maps a statistical model of object shape and appearance to a new image, which can be used for matching and tracking faces or for medical image interpretation. In this section, we briefly describe the idea of AAM.

As shown in Figure 3.13, three applications of PCA are used by (Cootes et al., 2001). In the appearance eigenspace calculation a PCA matrix P_A is obtained to model appearance changes. An $M \times N$ pixel ROI with color values can be represented as

$$\mathbf{x_A} = (r_1, g_1, b_1, r_2, g_2, b_2, ..., r_{MN}, g_{MN}, b_{MN})^T \tag{3.15}$$

In the case of shape based calculation, feature vectors like the one in equation 3.13 are used to model shape deformations. The resulting PCA matrix is P_S, and the combination of appearance and shape features is computed as follows,

$$\mathbf{x_{A,S}} = (\mathbf{x}_A W {P_A}^T, \mathbf{x}_S {P_S}^T)^T. \tag{3.16}$$

The final PCA matrix $P_{A,S}$ is calculated in order to compress the data, and reduce the redundancy of the appearance and shape correlation. The result of this component $\mathbf{y_{AAM}} = P_{A,S}\mathbf{x_{A,S}}$ are the AAM feature vectors. By applying data rotation and data projection as used for the appearance based methods, the final visual speech features can be obtained.

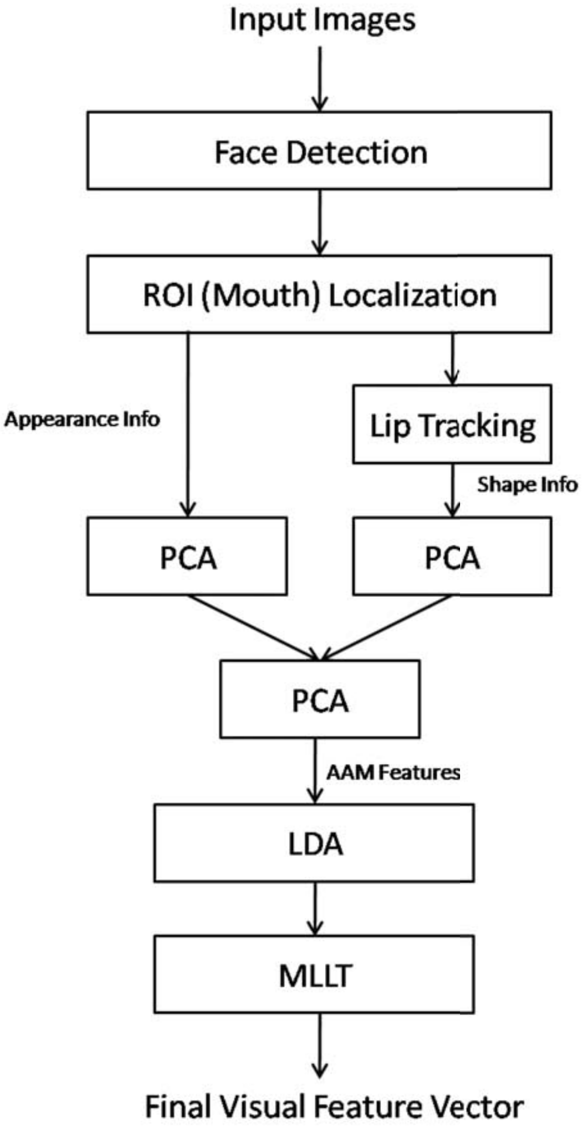


Figure 3.13: Active appearance based method of visual front ends from (Cootes et al., 2001)

Chapter 4

Audio-Visual Classification

Before discussing the methods of information fusion, it is necessary to introduce the idea of classification, which is an important issue behind the most of information fusion techniques. In this chapter we start with a brief description of the pattern classification background. The Bayesian theorem and the idea of informative and discriminative classification are described in the first section. Typical examples for informative and discriminative classifiers, namely Hidden Markov Models (HMM), Artificial Neural Networks (ANN) and Dynamic Bayesian Networks (DBN) are presented in detail in the following sections.

4.1 Pattern Classification Background

Statistical classification is a procedure of grouping items with similar characteristics together based on a given training information which includes input features and a corresponding transcription. Classification can be considered as the problem of estimating density functions for training data in a high-dimensional space and dividing the space into regions or classes.

The Bayesian theorem is the basic solution of most pattern recognition problems, since it tries to minimize the probability of a classification error. As an example, according to Bayesian theorem:

$$P(W \mid O) = \frac{P(O \mid W)P(W)}{P(O)} \tag{4.1}$$

the ASR problem can be stated as the problem of finding the sequence of words W which maximizes $P(W \mid O)$, where O refers to a sequence of observations, i.e. feature vectors. $P(W \mid O)$, $P(W)$ and $P(O \mid W)$ are called posteriori probability, prior probability and likelihood respectively.

To compute the posterior probability, we can segment all classifiers into two groups, informative and discriminative classifiers.

Informative classifiers model the class densities directly. Classification is done by examining the likelihood of each class producing the observed feature values and assigning to the most likely class to them. Examples include Fisher Discriminant Analysis, Hidden Markov Models(HMM), and Dynamic Bayesian Networks(DBN). In ASR, for example, the effort of maximizing $P(W \mid O)$ can be converted to a search for the sequence W which maximizes $P(O \mid W)P(W)$. $P(W)$ is known as the language model (LM) capturing high-level constraints and linguistic knowledge. $P(O \mid W)$ refers to the acoustic model, which describes the statistics of sequences of parameterized acoustic observations in the feature space given the corresponding uttered words (e.g. phone sequences). The most popular stochastic approach for acoustic modeling is HMM, where states of the hidden part represent phones (or sub-phonetic units), whereas the observable part accounts for the probabilities of the corresponding acoustic events.

In contrast to informative classification, the discriminative classification approach makes no attempt to model the underlying class feature densities. The focus is on modeling the class boundaries or the class membership probabilities directly. Examples include Logistic Regression, Artificial Neural Networks (ANN), and Support Vector Machines (SVM). With large enough training data and sufficient network size, ANNs are effective at modeling unknown distributions, i.e. learning the posterior probability of a class given an observation, $P(W \mid O)$.

The phone based AVSR and the articulatory information based AVSR could be designed by using either of these two groups of classifiers or combined classifiers. In the following sections, particular emphasis is placed on the classifiers used in our articulatory information based AVSR.

4.2 Hidden Markov Models

Hidden Markov Models (HMM) is a popular mathematical method for obtaining stochastic models of temporal or spatial observations. They can be used to model any time series, especially in the area of speech recognition.

4.2.1 From Observable Markov Models to Hidden Markov Models

In a discrete Markov process, a Markov chain is made of a finite number of states and selected state transitions among them. Along a regularly spaced discrete time, the system runs through a sequence of states according to a set of probabilities associated with the state transitions. To precisely calculate the probability of the next state, a full probabilistic description needs to take into account not only the current state but also all the predecessor states. If we consider a discrete, first order Markov chain as a special case, a probabilistic description to this model would be as follows,

$$P[q_t = S_j \mid q_{t-1} = S_i, q_{t-2} = S_k, \ldots] = P[q_t = S_j \mid q_{t-1} = S_i] \tag{4.2}$$

where q_t refers to the actual state at time t. S_i, S_j and S_k are possible state instances within system, which may denote the same state. As an approximation, equation 4.2 only considers the current and the predecessor state. Here we name the right-hand side of the equation 4.2 as transition probability a_{ij}, and it has the following properties,

$$a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i]$$

$$a_{ij} \geq 0;$$

and

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

In the above Markov model, the output of the process is a series of states to which corresponding observable events exist. Therefore, the above model could be called an observable Markov model. The state is directly visible to the observer, and the state transition probabilities are the only variables to consider in the system.

Since in the observable Markov models each state must correspond to an observable event, this model is too restrictive for many real world problems. (Baum and Petrie, 1966) proposed a hidden Markov model which assumes that the observation is a probabilistic function of the state. The underlying stochastic process of HMM is not directly observable, but can only be observed through another stochastic process that produces the sequence of observations. In other words, HMMs describe a set of observations, assuming they came from some unknown or hidden Markov process whose internal states are not directly observable. There are, however, visible probabilities which are influenced by the state. 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.

L.R.Rabiner has explained well the basic ideas of HMM in (Rabiner, 1989). A set of variables are formally defined as follows. By computing the values of these variables, an HMM can be used as a generator for an observation sequence.

- *Model states* are denoted as $S = \{S_1, S_2, \ldots, S_N\}$, where N is the number of states in the model. Since in HMM the states are hidden, the choice of N usually depends on the application. Generally the states are interconnected, that is, each state should be connected with another one. The state at time t is written as q_t.

- *Observation symbols* are denoted as $V = \{V_1, V_2, \ldots, V_M\}$, where M is the number of distinct observation symbols per state.

- *The transition probability distribution* can be written as $A = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i], \quad 1 \leq i, j \leq N$$

.

- *The emission probability distribution* in state j is $B = \{b_j(k)\}$, where
$$b_j(k) = P[v_k \; at \; t \mid q_t = S_j], \quad 1 \leq j \leq N, 1 \leq k \leq M$$
.
- *The initial state distribution* can be written as $\pi = \{\pi_i\}$, where
$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N$$
.

The above definitions describe a complete specification of an HMM which contains two model parameters N and M, and three probability distributions A, B and π.

4.2.2 Three Problems and Their Solutions

To use HMMs in real-world applications, three basic problems need to be solved, namely the evaluation problem, the decoding problem and the learning problem.

Evaluation Problem

Given the observation sequence $O = O_1 O_2 ... O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O \mid \lambda)$, the probability of the observation sequence, given the model? This problem can also be interpreted as an attempt to determine how well a given model matches a given observation sequence.

To calculate $P(O \mid \lambda)$, we can use simple probabilistic techniques. For example, the probability of O (given the model) can be obtained by summing the joint probabilities over all possible state sequences. But this calculation requires number of operations in the order of $2TN^T$. This is very large even if the length of the sequence T is moderate. Therefore other algorithms are needed to reduce the computation complexity. The forward-backward algorithm (Rabiner, 1989) is such an algorithm for computing the probability of a particular observation sequence with a considerably lower complexity.

In the forward-backward algorithm, we first define a forward variable $\alpha_t(i)$ as
$$\alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_i \mid \lambda) \tag{4.3}$$

Here, $\alpha_t(i)$ is the probability of a partial observation sequence, $O_1O_2\dots O_t$, and state S_i at time t, given the model λ. It can be computed inductively in three steps. In the initialization step, the forward probability set to the joint probability of state S_i and the initial observation O_1, as $\alpha_1(i) = \pi_i b_i(O_1)$. In the induction step, at time $t+1$, state S_j can be reached from N possible predecessor states, S_i, at time t. As shown in 4.4, the forward variable at time $t{+}1$ is computed from two parts, one is the sum of the products between all forward variables at time t and their transition probabilities to state j, and another one is the emission probability for state j at time $t+1$. This is iteratively computed for all states j at time t, where $t = 1, 2, \dots, T-1$.

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(O_{t+1}) \tag{4.4}$$

Finally in the termination step, as shown in 4.5, the observation sequence probability $P(O \mid \lambda)$ is calculated by the sum of the terminal forward variables $\alpha_T(i)$ for all states i, according to the definition of the forward variable as given in equation 4.3.

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{4.5}$$

The calculation of forward variables mainly depends on the number N of predecessor states in the HMM, no matter how long the observation sequence. Comparing to the direct calculation, this effectively reduces the calculation efforts from $2TN^T$ to N^2T.

Similar to the forward variable, we can define a backward variable $\beta_t(i)$ as the probability of the partial observation sequence $O_{t+1}O_{t+2}\dots O_T$, given that the current state is i.

$$\beta_t(i) = P(O_{t+1}O_{t+2}\dots O_T \mid q_t = S_i, \lambda) \tag{4.6}$$

By assigning an initial probability 1 to all terminal backward variables $\beta_T(i) = 1$, $\beta_t(i)$ can be recursively computed as follows,

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1})\beta_{t+1}(j). \tag{4.7}$$

By combining forward variable and backward variables, we obtain,

$$\alpha_t(i)\beta_t(i) = P(O, q_t = S_i \mid \lambda). \tag{4.8}$$

This provides us with another way to calculate $P(O \mid \lambda)$:

$$P(O \mid \lambda) = \sum_{i=1}^{N} P(O, q_t = S_i \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \tag{4.9}$$

Decoding Problem

Given the observation sequence $O = O_1O_2...O_t$, and the model λ, how do we choose a corresponding state sequence $Q = q_1q_2...q_t$ which is able to best explain the observation sequence? In other words, the task consists in finding the most likely sequence of hidden states that results in the observed sequence of events.

The solution to this problem depends upon the way "most likely state sequence" is defined. A simple approach would be to find the most likely state q_t at time t and to concatenate all such individual states. But this method only determines the most likely state at every point in time, and does not necessarily produce a physically meaningful state sequence. Instead, it is necessary to find a single state sequence which maximizes $P(Q \mid O, \lambda)$ or $P(Q, O \mid \lambda)$. The Viterbi algorithm solves this problem and finds the whole state sequence with the maximum likelihood instead of a combination of individual states. In order to facilitate the computation we first define an auxiliary variable,

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} P(q_1q_2 \dots q_t = i, O_1, O_2 \dots O_t \mid \lambda) \tag{4.10}$$

which refers to the highest probability along a single state sequence and a partial observation sequence up to time t, while the current state is i. By induction we have the following recursive relationship,

$$\delta_{t+1}(i) = [\max_{1\leq i\leq N} \delta_t(i)a_{ij}]b_j(O_{t+1}) \tag{4.11}$$

where the score is initialized to $\delta_1(i) = \pi_i b_i(O_1)$. Based on the dynamic programming algorithm, the procedure for finding the most likely state sequence starts by calculating $\lambda_T(i)$ using the recursion in 4.11, while always keeping a pointer to the "winning state" of the maximum detection. Finally the state $j*$ is found where

$$j* = \arg\max_{1\leq i\leq N}[\delta_T(i)]. \tag{4.12}$$

Starting from $j*$, the sequence of states is back-tracked as the pointer in each state indicates. This gives the required set of states.

Learning Problem

Generally, the learning problem consists in adjusting the HMM model parameters $\lambda = (A, B, \pi)$, so that the given set of observations O (called the training set) is represented by the model best. That means, to maximize $P(O \mid \lambda)$ for the intended application.

According to different optimization criteria for learning, there are various solutions for estimating the model parameters. The decision for a suitable one depends on the application itself. Maximum Likelihood (ML) estimates λ by finding the values for λ that maximize $L(\lambda)$, where $L(\lambda)$ is the likelihood estimate for the parameter of λ for a fixed observation $O = O_1, O_2, \ldots, O_T$. It is impossible, however, to analytically determine the model λ, which maximizes $L(\lambda)$. Instead, we can choose the model parameters in a way that the local likelihood is maximized by using an iterative procedure, like the Baum-Welch method, which is described below.

The Baum-Welch algorithm is an extension of the previously discussed Forward-Backward algorithm. In addition to the forward and backward coefficients defined in a previous section, we need to define two more auxiliary variables $\xi_t(i, j)$ and $\gamma_t(i)$. These variables can be expressed in terms of the forward and backward variables.

Firstly, $\xi_t(i, j)$ is defined for the probability of being in state S_i at time t and state S_j at time $t + 1$, given the observation sequence and the model

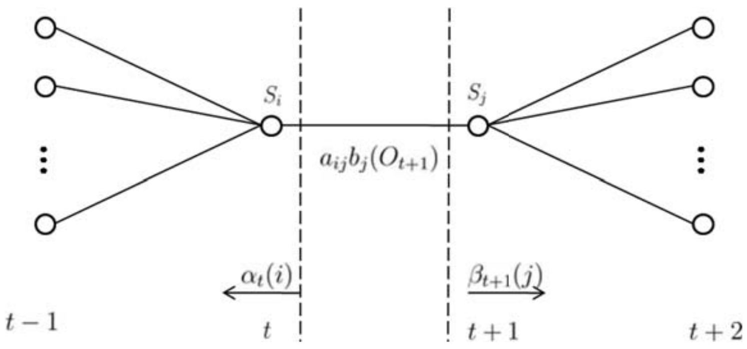


Figure 4.1: Illustration of the computation of the variables in the Baum-Welch algorithm. ()

parameters.

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) \tag{4.13}$$

As shown in Figure 4.1, it is easy to obtain the following formulas by the definition of forward and backward variables.

$$\xi_t(i,j) = \frac{P(q_t = S_i, q_{t+1} = S_j, O \mid \lambda)}{P(O \mid \lambda)} \tag{4.14}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{4.15}$$

The second variable $\gamma_t(i)$ is defined as the probability of being in state S_i at time t, given the observation sequence and the model parameters. It can be expressed in terms of the forward and backward variables as follows,

$$\gamma_t(i) = P(q_t = S_i \mid O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \tag{4.16}$$

Combining equations 4.15 and 4.16, we get the following relationship between $\xi_t(i,j)$ and $\gamma_t(i)$,

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j). \tag{4.17}$$

According to the above formulas and the concept of “occurrence counting”, an updating model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ can be interpreted as follows.

$\overline{\pi}_i$ represents the expected frequency of being in state S_i at time $t = 1$,

$$\overline{\pi}_i = \gamma_1(i). \tag{4.18}$$

$\overline{\alpha}_{ij}$ is the result of dividing the expected number of transitions from state S_i to S_j by the expected number of transitions from state S_i,

$$\overline{\alpha}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t}^{T-1} \gamma_t(i)}. \tag{4.19}$$

$\overline{\beta}_j(k)$ is the expected frequency of being in state j and observing symbol v_k divided by the expected frequency of being in state j,

$$\overline{\beta}_j(k) = \frac{\sum_{t=1,O_t=v_k}^{T} \gamma_t(j)}{\sum_{t}^{T} \gamma_t(j)}. \tag{4.20}$$

These re-estimation functions (4.18-4.20) can be calculated directly by maximizing the following auxiliary function,

$$Q(\lambda, \overline{\lambda}) = \sum_{Q} P(Q \mid O, \lambda) \log[P(O, Q \mid \overline{\lambda})]. \tag{4.21}$$

4.2.3 HMM in Speech Recognition

Hidden Markov models have been successfully applied in many areas, such as temporal pattern recognition, part-of-speech tagging, musical score following, partial discharges and bio-informatics. In automatic speech recognition, HMMs are used to draw a mapping between sequences of speech vectors and the desired underlying symbol sequences. There are two problems which make this mapping difficult. Firstly, different symbols can be mapped onto similar or even the same sounds. On the other hand, the same symbol can be improperly recognized differently because of large variations of speech waveforms in the real world, caused by speaker variability, environment, etc. Therefore, the use of HMMs in speech recognition is an N-to-N mapping instead of a one-to-one mapping between a speech waveform and a static pattern. Secondly, the boundaries among speech symbols are hard to be determined from the speech waveform alone. The boundaries can be located between words, phones, or other speech units according to the types of speech recognizers.

As an example, we consider a simple isolated word recognizer. For each word from a vocabulary of size W, we want to design a separate N-state HMM. After processing the speech signal of a given word into a time sequence of feature vectors, the first task is to build individual word models. The task of an isolated word recognizer is to solve the optimization problem $\arg\max_i P(W_i \mid O)$, where W_i is the ith vocabulary word. According to the Bayesian theorem in 4.1, $P(O \mid W_i)$ is computed by means of the acoustic model which are trained by using the Baum-Welch algorithm to estimate the model parameters for each word model. Once the set of words has been trained, the recognition of an unknown word is performed using the Forward-backward procedure to score each word model based upon the given test observation sequence, and selecting the word whose model score is highest.

If we are interested in the physical meaning of the model states, we can use the Viterbi algorithm in an additional step to align the elements of the training sequence to the states of the model, and then study the properties of the feature vectors that describe the observations occurring in each state.

In contrast to isolated word recognizers, continuous speech recognizers allow users to speak almost naturally, while the computer determines the spoken words. Recognizers with continuous speech capabilities are more difficult because of the “coarticulation” effect, varying speech rates and implicit utterance boundaries. Such systems use sub-words such as phones as modelling units. Phone HMMs are connected together to form a sequence using two more non-emitting entry and exit states for each model as glues. During training of a continuous speech recognizer, the boundaries dividing the segments of speech corresponding to each underlying sub-word model will not be known. That means that the boundary information cannot be obtained from the transcription alone especially for a large amount of data. However, there are some solutions to solve the utterance boundary problem. For example, embedded training uses the same Baum-Welch procedure as for the isolated case but rather than training each model individually all models are trained in parallel. HMM parameters are repeatedly updated until the required convergence is achieved. Although the location of symbol boundaries in the training data is not required, for such a procedure the symbolic transcription of each training utterance is still needed.

4.3 Artificial Neural Networks

In the previous section an informative classification model, namely the HMM, has been described. This section introduces another type of model based on Artificial Neural Networks (ANNs). ANNs are loosely inspired from principles of data processing in the brain and are usually expressed in network diagrams, where the number of inputs is dictated by the dimensionality of the input observation vector, and the number of outputs is dictated by the number of classes. ANNs may be used as nonparametric discriminant functions in pattern classification, as multiple regression models, or as "universal" function approximators. According to these properties of ANNs, they are suitable to be applied in a variety of ASR tasks, either as a stand-alone model or in combination with other models (Dahl et al., 2012).

There are many types of ANNs for dealing with different problem classes. The first and most simple type of ANNs is the feedforward neural network. The information in this network moves only forwards. There are no cycles or loops in the network. The simplest example of the network is the perceptron. The perceptron is a binary classifier which maps its input to an output by a linear condition. By simply adding layers to the architecture of Perceptrons, we can get the most common types of ANN, the Multilayer Perceptrons (MLP) (Cybenko, 1989). An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Since MLPs are widely used in many ASR systems, including the ones in this thesis, we will introduce the theory of MLPs and the backpropagation algorithm for training them later in this section.

Besides the feedforward neural network, there are many other types of ANNs such as, the self-organizing map (SOM) (Kohonen et al., 2001). It is constructed from a set of neurons to map points from an input space to coordinates in an output space. Being unsupervised, the training algorithm is an instance of the maximum-likelihood estimation for mixtures of Gaussian components, exactly like in HMMs. Furthermore, in contrast to feedforward networks recurrent neural networks (RNNs) (Jain and Medsker, 1999) are models with bi-directional data flow. While a feedforward network propagates data in one direction from input to output, RNNs also propagate data from later processing stages to earlier stages. RNNs can be used to deal with time sequences. Although various types of ANNs can be employed in ASR applications, we focus mainly on MLPs and their backpropagation algorithm, since they provide a good basis for the combination of ANNs and HMMs.

An MLP combines simple Perceptrons into a complex network consisting of several layers including hidden ones (4.2). Except for the input nodes, each node is a neuron with an activation function. The activation function, which defines the output of a node given set of inputs, associated to hidden and output units can be linear or non-linear. In contrast to input and output layers, the hidden layers deal with internal and intermediate representations. In a feedforward manner, input data are passed from one layer to another until the output data is generated. MLP utilizes a supervised learning technique called backpropagation for training the network. The training algorithm estimates a set of weights to be assigned to the links between each pair of units from adjacent layers.

The supervised training starts by initializing the input weights for all neurons to some random numbers between 0 and 1. Based on inputs of the network, we calculate the output. The resulting output is then compared with the desired output to get the error. According to this error, we modify the weights and threshold for all neurons. This process will be repeated until the error reaches an acceptable value, which means that the network has been trained successfully. Another case for stopping the process is that a maximum count of iteration has been reached, which normally stands for an unsuccessful training.

The challenge is to find a good algorithm for updating the weights and thresholds in each iteration to minimize the error. Given a multi-layer network with input units S_I, hidden units S_H, and output units S_O, we choose the gradient descent technique to minimize the cost function,

$$C = \frac{1}{2} \sum_{n \in S_O} (y_n - \overline{y_n})^2. \tag{4.22}$$

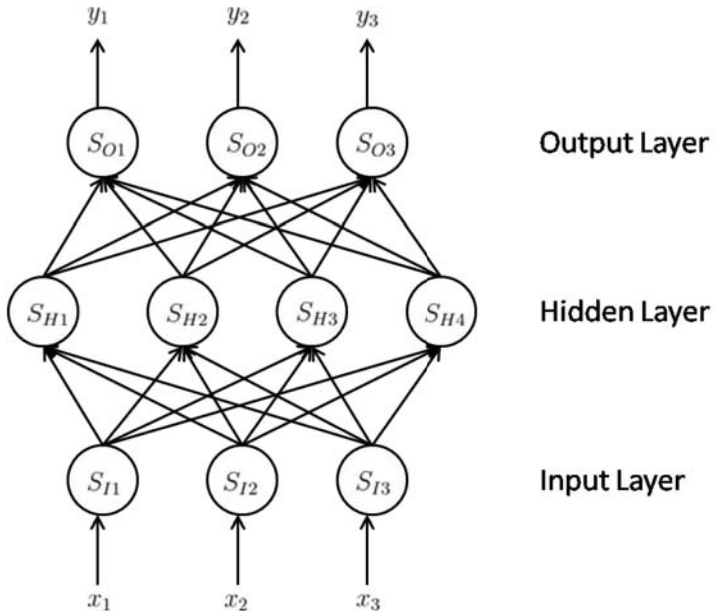


Figure 4.2: Multilayer Perceptrons.

where y_n is the nth target and $\overline{y_n}$ is the result generated from the output unit given the nth input. This cost function is used to correct the weights of the nodes. The weight change Δw_{ij} of the connection between the jth hidden unit and the ith output unit is computed as follows:

$$\Delta w_{ij} = -\eta \frac{\partial C}{\partial w_{ij}} = \eta (y_i - \overline{y_i}) f_i{}'(a_i)\overline{y_j} = \eta \delta_i \overline{y_j} \tag{4.23}$$

which is known as the *Delta Rule*, where $f_i{}'(a_i)$ denotes the derivative of the activation function computed over the current input a_i to the ith unit. Furthermore, let us consider a unit j of the hidden layer. The weight change Δw_{jk} associated to a connection between the kth input unit and the jth hidden unit,

$$\Delta w_{jk} = -\eta \frac{\partial C}{\partial w_{jk}} = \eta (\sum_{n \in S_O} w_{nj}\delta_n) f_j{}'(a_j) x_k \tag{4.24}$$

where δ_n is defined as in equation 4.23 for each output unit. For a generic unit j in the hidden layer we can similarly define,

$$\delta_j = (\sum_{n \in S_O} w_{nj}\delta_n) f_j{}'(a_j). \tag{4.25}$$

In 4.25 we see that the deltas for hidden units δ_j can not be computed as a direct function of the difference between the desired target output and the actual generated output. The way to solve this problem is to firstly compute the weight changes of the output layer according to the *Delta Rule*. Then propagate backward the deltas to the hidden layer. This is also known as *Back Propagation* algorithm (Rumelhart et al., 1986).

As a universal approximator, a trained MLP can be thought of as an "expert" in the category of information it has been given to analyze. The "expert" can then be used to project the new given inputs to different output categories. In particular, in the limit of an infinite number of training observations, the output of a trained MLP will approximate the true a posteriori probability for a given observation. MLPs are particularly useful if little knowledge about the form of the problem or nature of the training observations is available. ANNs have been successfully used for problems as diverse as automatic face detection (Rowley et al., 1996), and speech recognition with some success.

A major disadvantage of discriminant classifiers is their inter-class dependence. The formation of the decision boundaries requires clear knowledge of all other classes in the problem domain. Discriminant classifiers have found their application area in problems where all classes are static and well defined. This limits their application in a domain with complex classes like phone sequences for speech recognition. In Chapter 6 we try to overcome this disadvantage by combining informative and discriminant classifiers.

4.4 Dynamic Bayesian Networks

In the previous sections we discussed HMM based speech recognition, where hidden states represent a sequence of linguistic units and then relate these linguistic units to speech features. Two fundamental limitations, namely the interpretation problem and the factorization problem make it difficult to further improve HMMs for better modeling speech. On the one hand, in practical ASR systems, the physical meaning of the states is unclear and predefined differently according to developers' experiences and systems' requirements. For instance in a digit recognition task, the number of states is usually set to the same for all word models corresponding roughly to the average number of phones within words. Models with 2 to 10 states are considered to be appropriate for speech recognition tasks (Rabiner, 1989). This freely chosen number of states makes it difficult to assign a physical meaning to the states of an HMM, especially after training. Moreover, conventional HMMs are fundamentally unfactored. When a recognition task contains a combination of various information sources, HMMs cannot represent them precisely. This limitation is a disadvantage of HMMs for some special speech recognition tasks like audio-visual speech recognition. However, some efforts have been made to address this issue. (Potamianos et al., 2004) proposed phone-synchronous (state-asynchronous) multi-stream HMM and product (composite) HMM for audio visual speech recognition, and (Deng and Sun, 1993) designed their HMM framework for articulatory information accordingly. In this section, we describe Dynamic Bayesian networks (DBN), a generalization of HMM, which provides a compact representation of a joint probability distribution and can help to overcome the above mentioned limitations.

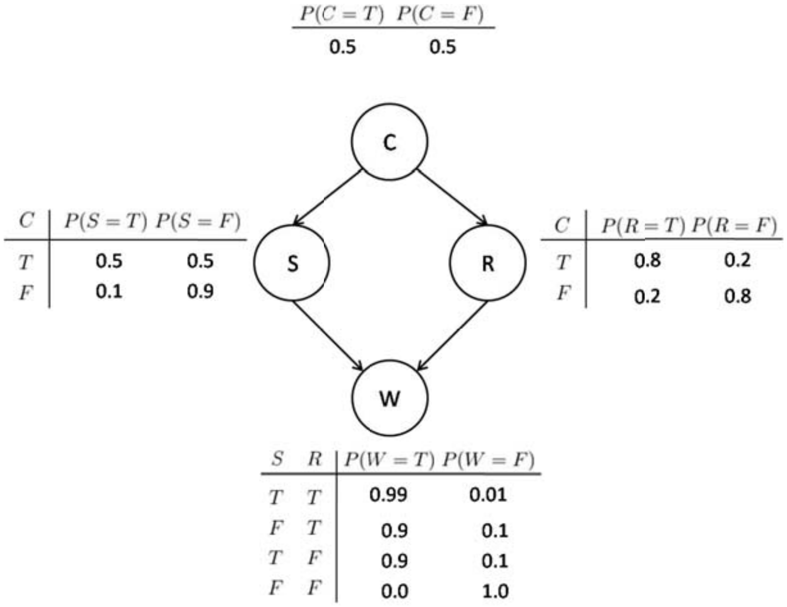
4.4.1 Bayesian Networks and Dynamic Bayesian Networks

Bayesian networks have been proposed as a specific graphical model (Frydenberg, 1990) by (Pearl, 1988) to represent a joint probability distribution of a set of random variables $X = \{x_1, x_2, \ldots, x_n\}$. If we assume that any x_i in X is conditional independent to all other lower-indexed variables, according to the chain rule the joint probability of X can be computed as:

$$P(x_1, \ldots, x_n) = \prod_i P(x_i \mid Parents(x_i)) \tag{4.26}$$

where $Parents(x_i)$ refers to a subset of variables $x_1, x_2, \ldots, x_{i-1}$. That is, the joint probability of all variables is the product of the probabilities of each variable given its parents' values.

Figure 4.3 illustrates an example of a Bayesian network. Assume that two events could cause grass to be wet, either the sprinkler is on or it's raining. Also, cloudy weather could either affect the sprinkler's working status or a change to rainfall. A probabilistic graphical model consists of a set of nodes and arcs depicted as a Directed Acyclic Graph (DAG) (Murphy, 2001), where nodes represent random variables and the (lack of) arcs represent conditional



$P(C=T)$	$P(C=F)$
0.5	0.5

C	$P(S=T)$	$P(S=F)$
T	0.5	0.5
F	0.1	0.9

C	$P(R=T)$	$P(R=F)$
T	0.8	0.2
F	0.2	0.8

S	R	$P(W=T)$	$P(W=F)$
T	T	0.99	0.01
F	T	0.9	0.1
T	F	0.9	0.1
F	F	0.0	1.0

Figure 4.3: An example of a Bayesian network (Murphy, 2001). Nodes represent random variables defined according to the scenario. Arcs represent conditional dependencies between variables. Tables, also called CPTs, are the parameters to be learned in the network.

independence assumptions. In general, an arc between two variables represents a direct dependency which can be interpreted as a causal relationship. In addition to the graph structure, the parameters of the model are specified as conditional probability distribution at each node. In case of discrete random variables, these parameters are stored in tables named Conditional Probability Table (CPT). A CPT provides a probability distribution for all the possible states of the child node, for each combination of possible states of the parent nodes. As shown in Figure 4.3, arcs represent the causalities and nodes are referred to events (C = It's cloudy, G = Grass is wet, S = Sprinkler is on, and R = It's raining). All three variables have two possible values T (for true) and F (for false).

This scenario is then modeled with a Bayesian network. Using the chain rule of probability and conditional independence assumptions, the joint probability of all the nodes in Figure 4.3 is,

$$\begin{aligned} P(C,S,R,W) &= P(C)P(S \mid C)P(R \mid C,S)P(W \mid C,S,R) \\ &= P(C)P(S \mid C)P(R \mid C)P(W \mid S,R) \end{aligned} \tag{4.27}$$

where the third term and the last term are simplified because of conditional independence relationships. The model can solve conditional probability questions like "How likely it was raining or sprinkling, given the grass is wet?" by calculating $P(R = T \mid W = T)$.

Dynamic Bayesian networks are a dynamic version of conventional Bayesian networks. The term "dynamic" refers to the system that we are modeling being a dynamic one. At each time slice t, a set of variables $X^t = x_1^t, \ldots, x_n^t$ is of interest, where a variable x_i^t represents the value of the ith parameter at time t. Except for the initial and final time slice, the topology of other frames in the network is a repeating structure. All the CPTs are also the same across the time slices. The joint probability distribution is then represented as,

$$P(x_1, \ldots, x_n) = \prod_i P(x_i^t \mid Parents(x_i^t)). \tag{4.28}$$

Assuming that networks obey the first-order Markov rule (Young et al., 2006), we know that the parents of a variable in time slice t must occur in either slice t or $t-1$. The joint probability distribution for all time frames can

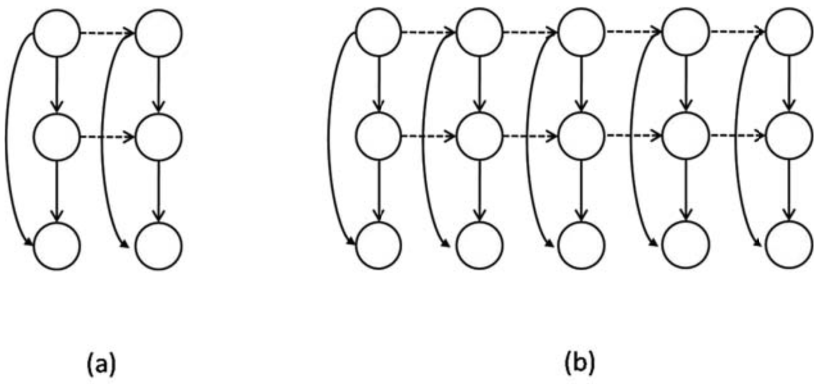


Figure 4.4: A "rolled" DBN template (a) and an unrolled DBN (b).

then be simply represented using a "rolled" template, where the conditional distributions within and between slices are repeated. A "rolled" template is well suited to be explained and understood during the design phase, and an "unrolled" DBN topology is normally used during the learning and decoding phase to deal with a suitably sized network for a given observation. As an example, Figure 4.4 (a) illustrates a rolled template with intra- and inter-slice arcs. Figure 4.4 (b) is then unrolled to show five time steps.

A DBN is a Bayesian network used for modeling time series data. It is suited for modeling temporal processes according to following reasons. Firstly, it is easy to represent arbitrary nonlinear properties by a tabular representation of conditional probabilities in a DBN. Secondly, each concept in a task can be explicitly modeled by a variable node in the DBN graph. Finally, the joint distribution can be factorized freely according to the requirements of the given recognition task.

4.4.2 Inference and Learning

Inference in Tree Structured Graphs

Inference in a probabilistic graphical model refers to obtaining conditional probabilities of any subset of variables given any other subset. As the simplest case, we start with an inference algorithm for a tree-structured graph.

Assuming that $B = \{B_1, B2, \ldots, B_n\}$ represents all variables in a graph, the posterior probability of B_i given a certain evidence E can be calculated by the Bayesian theorem as,

$$P(B_i \mid E) = P(B_i)P(E \mid B_i)/P(E), \tag{4.29}$$

where evidence E refers to the instantiation of some variables. As shown in Figure 4.5, any node of B can separate E in two parts. E^- is the set of observed values for the evidence variables in the subtrees rooted in B_i's children. E^- is the set of observed values for all other evidence variables. Furthermore since E^-,E^+ are conditionally independent given B, then Equation 4.29 can be written as,

$$\begin{aligned} P(B_i \mid E) &= P(B_i)P(E^-, E^+ \mid B_i)/P(E) \\ &= \alpha P(B_i \mid E^+)P(E^- \mid B_i) \end{aligned} \tag{4.30}$$

where α is a normalizing constant. Now we define two important quantities λ and π according to 4.30.

$$\lambda(B_i) = P(E^- \mid B_i),$$

$$\pi(B_i) = P(B_i \mid E^+)$$

so that the posterior probability can be written as $P(B_i \mid E) = \alpha\pi(B_i)\lambda(B_i)$.

During inference, each node in the graph stores the vectors, π, λ, and the conditional probability matrix p as parameters. Probability propagation is done through a message passing mechanism (Pearl, 1988) in which each node sends messages to its parents and children. That is, the λ probabilities are calculated in a bottom up way through the tree. The π probabilities are calculated based on λs in a top down manner.

Inference in Junction Trees

In the more general case Bayesian networks are not restricted to a tree structure. Inference is often done by clustering groups of variables from the original graph into "cliques". A clique C is a set of nodes in a graph such that all nodes in C are pairwise connected, and the set is maximal with respect to this property, i.e. not contained within another clique. By using cliques

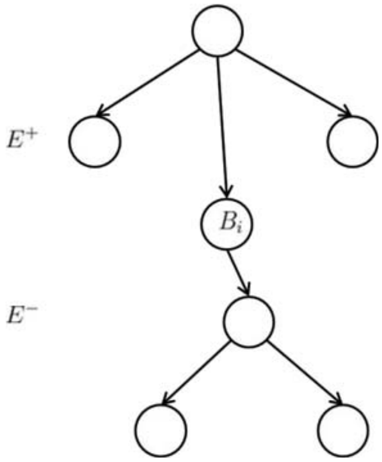


Figure 4.5: A tree structured Bayesian network. Evidence E is divided into two parts E^+ and E^- by a random variable B_i.

as nodes and drawing edges between these nodes, we can construct a new graph with tree structure. This new graph is named a clique tree or junction tree. The idea of a clique tree is to represent the same joint probability distribution as the old graph by means of a simple tree structure.

Figure 4.6 depicts a junction tree derived from a non-tree structured graph. The first step in junction tree construction consists of connecting nodes in the Bayesian network that have a common child and making all edges in the graph undirected, resulting in the so-called moral graph. This procedure, known as moralization, preserves the conditional independencies of the original network. In the second step, undirected edges are added between each parent resulting an undirected graph. The third step is the triangulation. By triangulation, undirected edges are added to derive a decomposable graph where junction tree exists and the message passing algorithm can be applied. Finally, the junction tree can be constructed as shown in Figure 4.6(d). The message passing algorithm can be applied onto this tree structured graph.

Inference in DBNs

In case of DBN, inference procedures are similar to constructing a “dynamic” junction tree (Figure 4.7). Firstly, moralization is done within each time slice. In the second step, we define a DBN partition at the boundary between each slice pair. This is done by adding the adjacent nodes from the previous

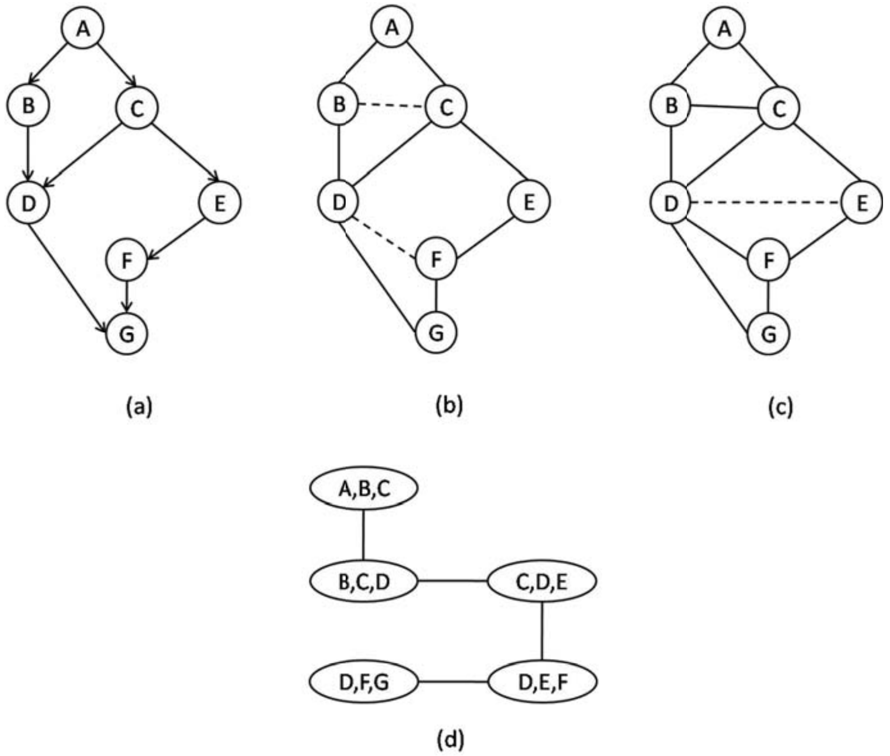


Figure 4.6: Junction tree construction. (a) Original Bayesian network. (b) Moral graph. Dashed lines are added during moralization. (c) The triangulated moral graph. The dashed line is added during triangulation. (d) A junction tree. (Zweig and Russell, 1998)

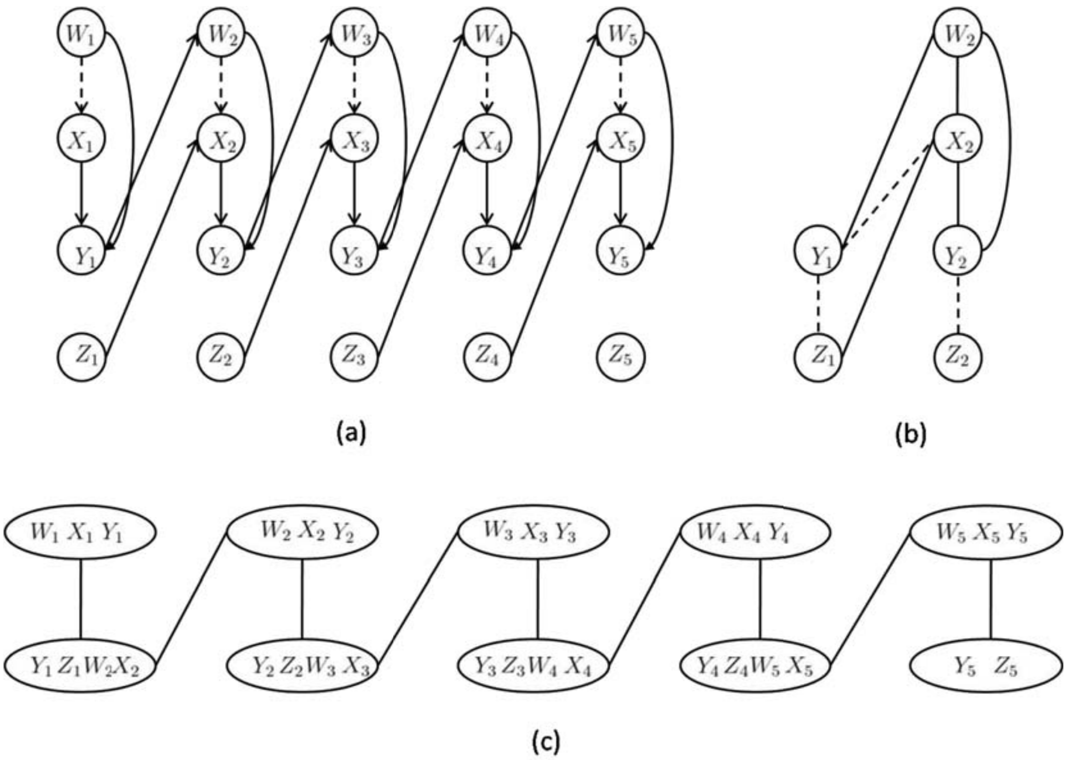


Figure 4.7: Dynamic junction tree construction. (a) Moralization within slices. Dashed arcs are added during moralization. (b) DBN partitions: triangulation between adjacent slices. Dashed lines are added during triangulation. (c) DBN junction tree.

time slice into the current time slice graph. The DBN partition is then triangulated and connected with its adjacent DBN partitions. Finally, a junction tree can be constructed from the triangulated DBN.

Learning

The learning problem in Bayesian networks requires two different techniques, determining the structure of the model and estimating the parameters (conditional probability distributions) for a given model structure.

Learning the structure is much harder than learning the parameters. For some complex issues, the network structure and the parameters of the local distributions must be learned from data. A detailed explanation of structure learning can be found in (Zweig and Russell, 1998). We will not discuss this issue here. In speech recognition, a Bayesian network is specified manually by the developer and is then used to perform inference. The DBN structures described in the thesis are all manually defined.

The learning techniques for parameters are analogous to the learning techniques for HMMs. Usually the conditional distributions are unknown and have to be estimated from data using the maximum likelihood (ML) approach. If there are hidden variables in a graph, it is impossible to directly calculate the maximum likelihood (or the posterior probability). Therefore, the expectation-maximization (EM) algorithm is applied to maximize the probability of observed data given the hidden variables' conditional probabilities. It has to be noted, that the EM algorithm is only applicable if the conditional probabilities can be represented by distributions in the exponential family (Zweig and Russell, 1998). If the derivative of the data likelihood with respect to the conditional probabilities can be computed, gradient descent (Zweig and Russell, 1998) is applicable.

Chapter 5

Using Articulatory Transcriptions in An HMM/N-Best Decision Framework

In this chapter we present a pilot experiment which uses articulatory transcriptions in AVSR. Firstly, we describe the design and implementation issues of this system. Then, we compare the results of this system with a baseline system which is a phone-based system. Finally, the advantages and disadvantages of this system are discussed. Its characteristics motivate us to design other systems in subsequent chapters.

5.1 Corpus and Baseline System

5.1.1 Corpus

Collecting corpus data for AVSR is expensive. Therefore, compared to ASR datasets, only a small number of audio visual data corpora exist. These audio visual data corpora can be further divided into two groups. For the audio visual speech recognition tasks, the corpora TULIPS1, AVletters, AVOZES, CUAVE, GRID, VidTIMIT, DAVID, IBM LVCSR and DUTAVSC are most frequently used. In speaker detection or identity verification applications, corpora like VALID, M2VTS, XM2VTS, VidTIMIT and DAVID are usually applied.

For the design of our audio visual speech recognition experiments, we consider the following criteria for dataset selection. Firstly, the data should be continuously spoken. Secondly, the corpus should support speaker independent recognition, which requires a large number of speakers being involved. Thirdly, the size of the vocabulary should also be large enough in order to provide for a variety of spoken sentences. Finally, the corpus should be spoken in English and available to the public. According to these criteria, we have chosen the GRID corpus (Barker and Cooke, 2007) for this pilot AVSR study.

The GRID corpus is a continuous audio visual speech corpus for an English small vocabulary task. It consists of high-quality audio and video recordings of 1000 sentences spoken by each of 34 talkers. The sentences in GRID are speech commands according to a very simple grammar. The total of 51 words within the vocabulary consist of 4 command words, 4 words representing color, 4 prepositions, 26 letters, 10 digits and 4 adverbs. Sentences are simple. Each sentence contains six words with syntactically identical structure *"command color preposition letter digit adverb"*, i.e. *"place green at B 4 now"*. The original audio and video data were recorded under clean acoustic conditions, and the video shows only a frontal view of each subject's face.

5.1.2 Acoustic Baseline System

The major purpose of this experiment is to find out whether articulatory information can contribute to an improved ASR. Therefore, a conventional phone-based classifier is trained as a baseline system. Since the articulatory transcription based AVSR is designed in an HMM/N-best decision framework, HMMs should also be used for the speech recognition task in the baseline ASR. The baseline ASR models are left-right HMMs with 3 emitting states. The monophone HMM models were further extended to context dependent triphone models. The test data is then recognized with the Viterbi algorithm using a simple language model.

5.2 Articulatory Transcription Approach

The articulatory transcription (AT) is a speech representation situated between the acoustic signal preprocessing level and the subword unit probability estimation level. This representation describes articulation-related information which is deemed relevant for the distinction between speech sounds.

In a conventional ASR, training information contains speech signals and the corresponding word transcriptions. The most common type of phonetic transcription uses a phonetic alphabet (such as the International Phonetic Alphabet) (International Phonetic Association, 1999), which labels words using a system of phonetic notations. For example, the word "please" can be labeled as the phonetic transcription [p l ih s]. Similar to the phonetic transcription, we can label words using a system of articulatory notations. Since articulatory information represents the movements of different articulators, the articulatory transcriptions should be defined for various articulatory features.

For each articulatory feature, we assume that each phone in the phonetic transcription maps to a set of values in the articulatory transcription. The phonetic transcriptions are based on the phone set ICSI (International Computer Science Institute) and the audio part of articulatory transcriptions is based on the following Table 5.1. This table lists the articulatory values used for different articulatory features, where the five audio channels are based on the design of (Kirchhoff, 1999) and the two visual channels have been defined by ourselves. Using these two tables we can easily convert the phonetic transcriptions into articulatory ones. For the articulatory feature "Voicing", for example, the word "please" can be labeled as [unvoiced voiced voiced unvoiced] according to the phonetic transcription [p l ih s]. Table A.1 given in the Appendix shows the mapping used in our approach.

There are several reasons for using articulatory transcriptions to build AVSR. Firstly, articulatory transcriptions can be relatively easy retrieved in contrast to the true geometry of articulators which is usually recorded by special equipments. Together with the Table A.1 and the phonetic transcriptions in hand, we are able to find the articulatory transcriptions by mapping phones and the corresponding articulatory values. Secondly, compared to a phone-based classification system, the individual AT-based classification system makes use of fewer classes, which therefore are better suited to be used in case of sparse training data. Thirdly, various channels of AT-based classification systems should lead to different recognition results. It is necessary to design a fusion component for investigating the synchronicity issues

Table 5.1: Articulatory transcriptions used in AVSR framework.

Channels	Values	Num. Classes
Voicing	voiced, unvoiced	2
Rounding	round, nil, flat	3
Manner	vowel, nasal, lateral, approximant, fricative	5
Place	dental, labial, retroflex, velar, high, mid, low	7
Front-Back	front, nil, back	3
Visual opening	open, close	2
Visual rounding	round, nil, flat	3

between articulatory features.

Motivated by the advantages described above, we propose a two-stage architecture (see Figure 5.1), where the articulatory information is extracted in parallel from both the speech signal and the video frames by means of articulatory transcriptions in the first stage. The second stage then combines these articulatory transcriptions outputs into AT-tuples and maps them to a corresponding phone stream. A lexical search maps this stream to word sequences as output.

In the first stage, we use articulatory information to train the multi-channels of HMMs. In contrast to (Kirchhoff, 1999) we do not attempt to detect the articulatory features in a pure bottom-up fashion, but train a number of independent word recognizers, where the words are defined in terms of articulatory transcriptions instead, as usual, in terms of phones. The speech signal is described by low level features, i.e. MFCC features in the audio channels and appearance-based features in the visual channels. In our experiment, seven AT-based HMM classifiers (5 from the audio signal and 2 from the visual channels) have been trained by Baum-Welch reestimation. The word recognizers are applied in parallel to the audio or visual data and their outcomes are word sequences which can also be interpreted in terms of sequences of articulatory values. This approach has the advantage that it allows us to integrate higher level information from a language model already during the AT-detection phase.

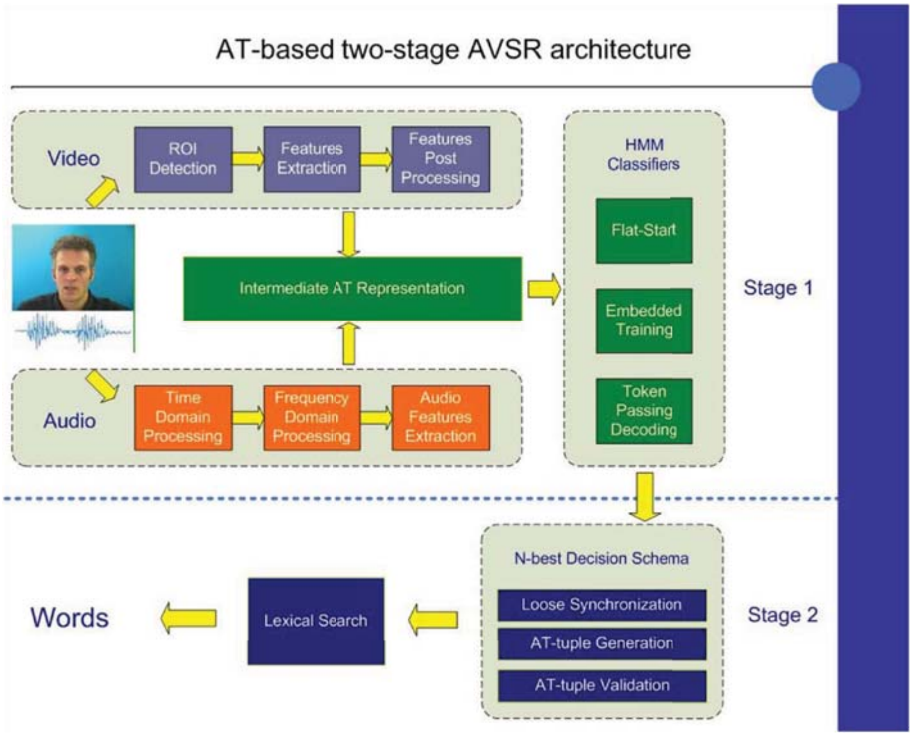


Figure 5.1: Articulatory Transcription based two-stage AVSR architecture.

Instead of selecting the single best decoding result, we determine the N-best hypotheses for all the AT-based classifiers as the outputs of the first stage. For recognition the Token Passing algorithm (Young et al., 1989) is used. Token Passing saves the best tokens at each word boundary, which gives the potential for generating a lattice of hypotheses rather than only a single best hypothesis. Since the tokens are saved at the word level, the output is actually a sequence of loosely synchronized hidden words. Synchronization is a loose one, since the HMM embedded training cannot guarantee a strict synchronization of the ATs within a word. However, thanks to the short pause models, which are usually easy to train, word boundaries can be detected during recognition. In accordance with this observation, we are able to represent the recognized words from various channels as AT sequences, where the word boundaries of all channels can be aligned at the same time points.

5.3 N-best Decision Schema

In the second stage, the output of the AT-based recognizers will be processed with the goal to combine the various channels into a single sequence of AT representations for which a meaningful phone representation exists. For this purpose, we propose an N-best decision scheme which computes the results of the first stage classifiers into a number of coherent AT tuples, which can be mapped to the phones contained in a code book.

For applying the N-best decision schema, two assumptions for the corpus are necessary to be considered. Firstly, all the training and testing sentences should have the same number of words. That requires that no insertion and deletion errors occurred during the first stage recognition. A strict grammar used as language model can help to fulfill this requirement. In the GRID corpus, the recognized words can be easily separated into all articulatory features because of the high accuracy recognition of short pauses between words. This makes the synchronization possible at the word boundaries. Secondly, all the words in the corpus should have a similar number of phones. This makes it possible to combine the articulatory values from different channels into a single AT tuple. In the GRID corpus, the number of phones in all words is between 1 to 5, and most words contain 2 to 4 phones, which helps the N-best decision schema to achieve fairly accurate combination results.

The N-best decision schema is similar to the mixture of experts (ME) architecture proposed by (Jacobs et al., 1991). Having available, however, the N-best output from the first stage, it seems more likely that the optimal results will be among them and a more reliable mapping between articulatory representations and phone representations can be established. In our experiments, we have always chosen the five best hypotheses. The N-best decision schema consists of five procedures, namely 1) Synchronization, 2) AT tuple generation, 3) Best output selection, 4) Weighting and 5) Lexical Search.

The N-best decision schema is invoked after the decoding stage of the AT-based classifiers. Assuming that the number of the classifiers is N_c, the decoded sentence by classifier n can be defined as a sequence of words W_{nk},

$$S_n = \{W_{nk} \mid n = 1, \ldots, N, k = 1, \ldots, N_w\}. \tag{5.1}$$

where N_w denotes the number of words recognized in the sentence. According to the above first assumption, N_w has the same value for all classifiers. After the process of "loose synchronization", the problem to decide on the best sentence is converted into a best word decision scenario.

A word W_{nk} can be defined as a sequence of articulatory values A_{nkm},

$$W_{nk} = \{A_{nkm} \mid n = 1, \ldots, N, k = 1, \ldots, N_w, m = 1, \ldots, N_a\}. \tag{5.2}$$

where N_a is the length of a word, which refers to the number of AT segments within that word. The value of N_a could be different for the same word recognized by various classifiers. Even in the same classifier, there is no identical N_a value among all the hypotheses. In order to combine the articulatory information from different classifiers, the recognized words are synchronized by normalizing the word length according to a majority vote among all the available output candidates of the AT-based classifiers. The normalization becomes necessary since the output of the first stage might be incoherent between alternative recognition hypotheses of the same classifier and across the different articulatory features. It is carried out as a greedy search based on two above mentioned assumptions. By selecting the optimal length, we look up the current recognized word from all N classifiers. Since the five best word candidates are collected in the system, there are in total $5N$ sequences of articulatory values A_{nkm} as word hypotheses. We choose the most common word length from all these word hypotheses as the word length of this recognized word. Those word hypotheses, which are only supported by a minority of AT-based recognizers, are excluded. Here the selected word length is denoted as $N_{\bar{a}}$.

Articulatory values can be combined into AT tuples where each component of the articulatory transcription tuple corresponds to the number of AT-based recognizers of the first stage. An AT tuple can be mapped to a phone. E.g. the tuple `[unvoiced, fricative, labial, nil, nil]` can be mapped to the phone [f]. We consider the possible mappings in the manually created AT-to-phone table shown in the Appendix A.2. Since the selected word length is $N_{\bar{a}}$, a final word will be generated by $N_{\bar{a}}$ AT tuples.

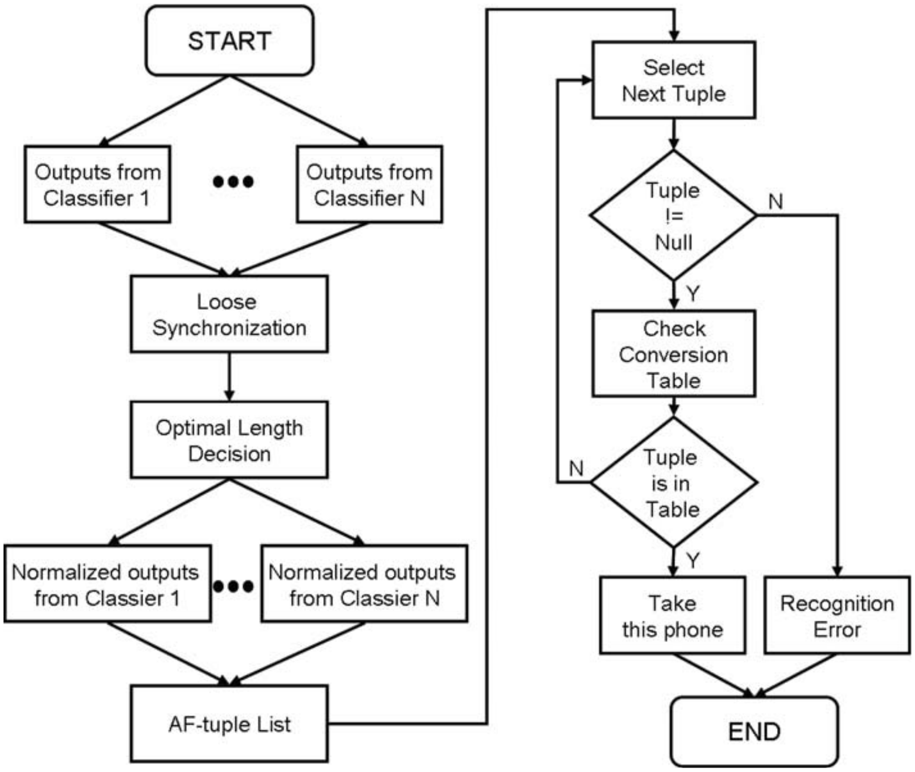


Figure 5.2: The flow chart of N-Best Decision Schema

AT tuples are generated taking into account the scores from the first-stage classifiers. The first AT tuple, for example, is generated by combining the topmost candidate decision from all the classifiers. The second one will replace the topmost candidate of the most unreliable classifier by its second best choice, etc. Since we have chosen only the five best candidates of the AT-based word recognizers and in many cases they agree in the proposed recognition results, we are able to consider all combination possibilities when generating AT tuples.

Ideally, the AT tuple derived from the best decision of each classifier is the most likely one from all candidates. However, since the combined results are based on inaccurate first stage classifiers, it might not always be possible to map the parallel feature assignments into phones. Therefore, we need to exclude such combinations from consideration. Figure 5.2 shows the flow chart of the N-best decision schema. The right part indicates the logic of best output selection. If the first output from the N-best list cannot be found in this table, it is replaced by another one from the list. If none of these tuples can be mapped into phones, a recognition rejection will be generated. The generated AT tuples are ranked based on the accumulated confidence score.

A phone stream is then defined as a sequence of phones which are admissible according to the AT-to-phone table. Eventually, this phone stream will be mapped into words according to the phone-to-word table. For this purpose, a pronunciation dictionary including some pronunciation variants is used. For instance, “five” is transcribed both as [f ai v] and [f ai f].

Weighting is used in the synchronization step. In order to vote for the optimal length of a word in the synchronization step, the decision could be weighted according to the recognition accuracy of the AT-based classifiers.

The example shown in Figure 5.3 illustrates the data flow within the N-best decision schema. The testing data is the sentence “bin red by t one please”. According to the first assumption of this approach, the word boundary is reliably detected by all first level AT-based word recognizers. Therefore, the synchronization can be achieved word by word. When the word “one” is processed, the five best decisions of the rounding classifier have a different length of words. Voting determines the “average” length among all word candidates from all AT classifiers to three segments and all candidates with another length are no longer considered. After this synchronization step, we select the ATs from all classifiers and combine them into AT tuples. Their number is limited in our example, because all the AT-tuples have the same value in this example, which can be mapped to the phone [n]. Together with the two neighboring phones eventually the word “one” is decoded.

5.4 The Phone Level Synchronization

In this section we analyze the information fusion and the temporal issues in articulatory transcription based AVSR. In the HMM/N-best decision framework, the decoding results of the first stage HMM classification are integrated in the second stage N-best decision schema. Therefore, the articulation transcription approach fuses the audio visual information and multiple channels of articulatory information at the decision stage.

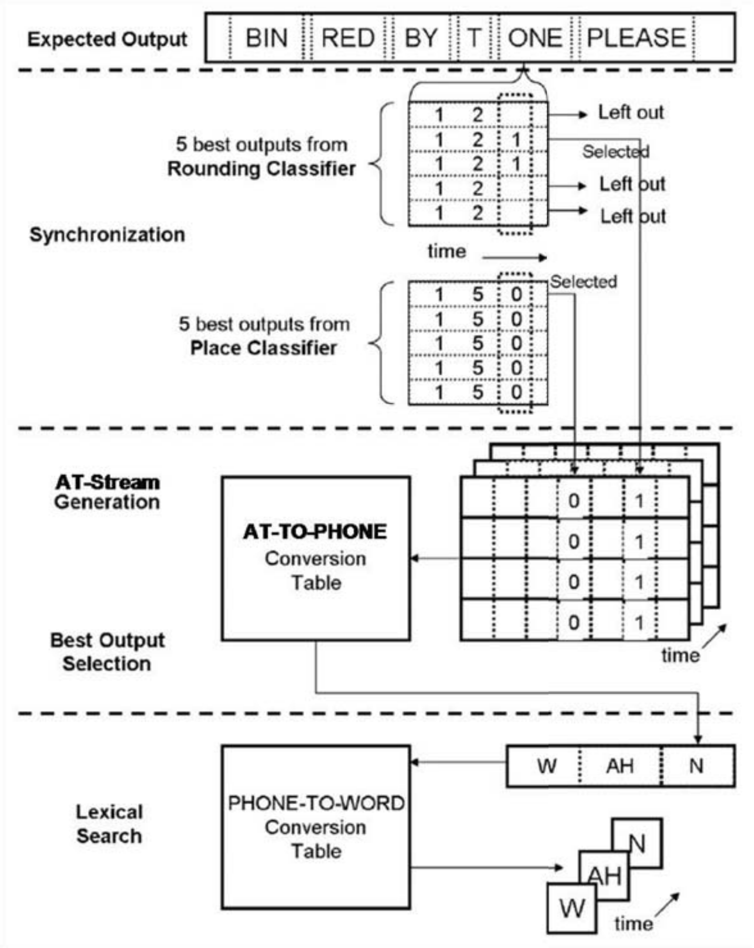


Figure 5.3: Example for the N-best decision schema. AT-classes are coded as numbers

As input of the N-best decision schema, sequences of recognized words are the sources for information fusion. According to the phone-to-AT conversion table, these words can be further decomposed into sequences of articulatory values. During information fusion, a word is segmented into phones. A combined tuple including various articulatory values is used to decide on the phone at the current time segment. Obviously the smallest time segments are not frames but phones. Therefore this approach synchronizes at the level of phones but leaves the frame levels unsynchronized.

However, according to the two assumptions introduced for the N-best decision schema, the phone level synchronization is not necessary to force the

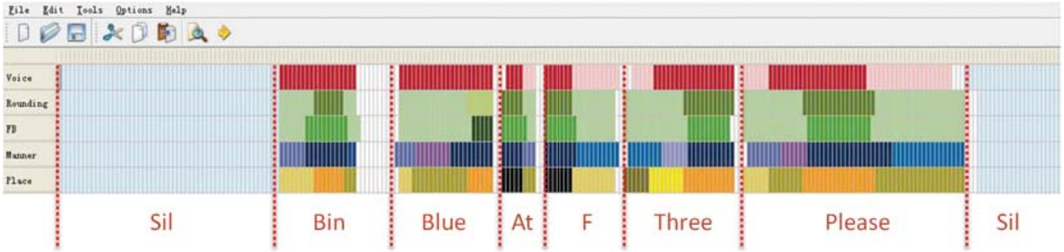


Figure 5.4: Alignment of AF Transcriptions.

articulatory values to share the same phone boundaries. For example, within the five audio channels, the phone /ih/ can be recognized by detecting the tuple of articulatory values $[voiced, vowel, high, front, flat]$. The $vowel$ value, however, can easily start several frames earlier than the value $flat$, or the $voiced$ value might continue longer than the value $front$. How to combine these values depends on the position of the value in the word. In a word with a length of $N_{\bar{a}}$ for example, there are in total $N_{\bar{a}}$ positions. Only the articulatory values with the same position can be combined into an AT tuple. Neither the starting and ending time of the value are considered in the N-best decision schema. This method loosely synchronizes the articulatory information at the phone level, which helps the system to model highly overlapping articulatory features.

Figure 5.4 shows the utterance *"bin blue at f three please"* force aligned in five channels of articulatory transcriptions. The different channels share similar word boundaries. Although the similarity of word boundaries is caused by the fairly accurate articulatory information for the GRID corpus. It might be useful information which can be used as input to the N-best decision schema to integrate the articulatory information at the phone level.

5.5 Results and Conclusion

In order to compute articulatory information, we classified the low level features into articulatory classes using left-right HMMs with 3 emitting states. The models are initialized with the flat start method (Young et al., 2006) and the HMM parameters are trained with maximum likelihood estimation. For recognition, the token passing algorithm is used without any pruning factor. We take the five best outputs from each individual channel and apply the

N-best decision scheme described above to them.

Figure 5.5 shows a comparison of three AT-based recognition systems with respect to their accuracy. Compared to the results of (Amer and Berndsen, 2003), where the articulatory features are also trained by HMMs, our system obtained better results in all individual classifiers. This performance is comparable to the one reported in (Kirchhoff, 1999), where Kirchhoff uses only bottom up information. Since the first stage of our system actually consists of AF-based word recognizers, the figure also presents the corresponding word recognition accuracy in the different channels. It is considerably lower than AF recognition rate because certain words can hardly be decoded using only AT classes.

Within such a two-stage architecture, the multi-channel AT-classifiers are trained and tested with either audio or video data. Their results are combined in a second stage, using the N-best decision schema. Table 5.2 shows the word recognition accuracy results for different versions of the second stage. As expected, the word recognition accuracy of the individual first stage AT-classifiers is considerably lower than that of the baseline triphone-based recognizer. After applying the N-best decision schema to combine the AT-decoding results, however, the overall word recognition accuracy rises even above that of the phone-based approach. When comparing the audio-only and audio-visual AT-based systems, the latter one gives slightly better results even under clean acoustic conditions and by further fine tuning the visual preprocessing algorithms a further improvement can be expected.

Table 5.2: AT-based word recognition accuracy after N-best decision

Systems	Results
Audio articulatory transcriptions	93.57
Audio-Visual articulatory transcriptions	93.71
Phone-based	90.34

The experiment carried out here, confirms the potential of using articulatory information for combining acoustic and visual cues for speech recognition purposes. However, this approach has two limitations. Firstly, the HMM/N-best decision approach is corpus dependent. The preconditions for using the

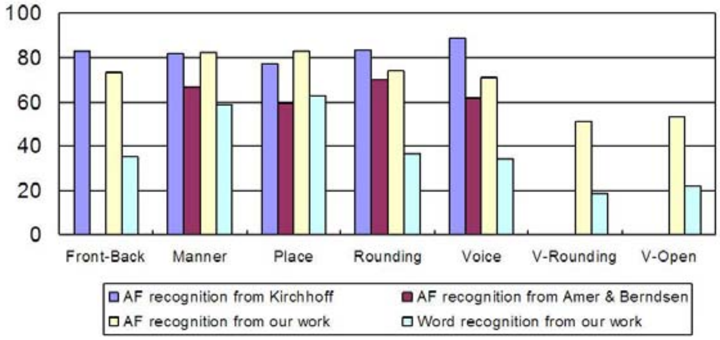


Figure 5.5: The first stage feature accuracy rates comparison on three AT-based recognition systems

N-best decision schema prevent this approach to scale up to arbitrary speech recognition tasks. If the speech corpus is built from a large vocabulary or according to a more freely defined grammar, the output of the first stage HMM classifiers could result in deletion or insertion errors, and it might become difficult to combine the articulatory information by means of the N-best decision schema. Secondly, the features used in the system are only low level ones, which still can be affected by noisy environments. If we use noisy data to test the models, the output of the first stage HMM classifiers will be inaccurate, which will also affect the integration in the N-best decision schema negatively.

Chapter 6

Using Articulatory Features in An ANN/HMM Framework

In this chapter we present an AVSR system which uses articulatory features instead of articulatory transcriptions. This approach is designed to solve the limitations described in the articulatory transcription based AVSR. The corpus and the baseline system for this experiment are first presented. Then the formal background of the ANN/HMM framework is introduced. In the third section we discuss the design and implementation of the articulatory feature approach. Then the synchronicity issue is analyzed. Finally, we present the experimental results and the conclusions for this approach.

6.1 Corpus and Baseline System

In the articulatory transcription based approach, two assumptions about the corpus have been made, which are necessary prerequisites for using the N-best decision schema. These assumptions, however, prevent the articulatory transcription approach to scale to large vocabulary tasks. In the articulatory feature approach, we try to remove these assumptions and evaluate the system on a corpus with a larger vocabulary and a more flexible sentence structure. For this purpose we have chosen the VidTIMIT corpus for training and testing.

The VidTIMIT corpus (Sanderson, 2009) is a continuous audio visual speech corpus for an English medium-sized vocabulary task. It contains 10 sentences spoken by 43 speakers each. The first two sentences for all speakers are the same, while the remaining eight ones differ between speakers. The sentences were chosen from the test section of the TIMIT corpus (Fisher et al., 1986). The corpus was recorded in 3 sessions, with a mean delay of 7 days between Session 1 and 2, and 6 days between Session 2 and 3. The recording was done in an office environment using a broadcast quality digital video camera. The corresponding audio signal is stored as a mono, 16 bit, 32 kHz WAV file.

In the articulatory transcription approach, the baseline system was an audio only phone-based speech recognizer. However, in the articulatory feature approach, the baseline system is trained with both audio and visual data. The low level audio and visual features and the phonetic transcription are used as training information to estimate the parameters of the HMMs. To compare the low level features and the articulatory features under the noisy conditions, the baseline system has also been tested on signals distorted with pink noise. Signals with different signal-to-noise ratios (SNR) are used as testing data to show the performance of different systems.

6.2 The Hybrid ANN/HMM approach

ANNs can be used for solving speech classification problems, such as word or phone recognition. By mapping temporal representations into spatial ones, or by using recurrences, ANNs can be applied to simple ASR tasks. However for the case of continuous speech recognition, ANNs were not successful in dealing with long time-sequences of speech signals, since the number of possible word sequences is in general infinite. Probabilistic models are better suited for such a task. To take advantage of the properties of both approaches, a combination of discriminative (ANNs) and informative (HMMs) classification is attempted. To combine both classification frameworks many approaches have already been investigated. For example, Bourlard et al. (Bourlard et al., 1996) proposed an architecture for continuous ASR in which an MLP was trained to estimate the posterior probabilities of HMM states. Instead of MLPs, there are other types of ANNs used as density models, such as time delay neural networks (TDNNs) (Dugast et al., 1994) and radial basis function networks (RBFs) (Huang and Lippmann, 1990). In this section, we mainly focus on the hybrid ANN/HMM architectures proposed by Bourlard and their variation.

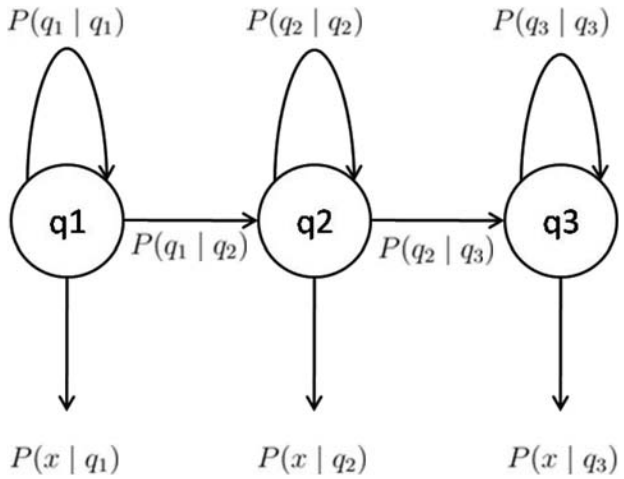


Figure 6.1: A three states hidden Markov model.

In the case of ASR, given the basic HMM equations, we would like to estimate the probability value $P(x \mid q)$. Assuming that a sentence is represented by a particular state sequence, $Q_1^N = \{q_1, q_2, \ldots, q_N\}$, Figure 6.1 shows a three states HMM where the state q of the model is not observed. The values $P(x \mid q)$ refer to the emission probabilities of the observed input speech signal data x given the hypothesized HMM state q. The other probability values $P(q_i \mid q_j)$ are the state transition probabilities.

In Bourlard's approach, MLPs are used to estimate the conditional posterior probabilities $P(q \mid x)$ of each HMM state q_i. By using Bayes' rule, the posterior probability distributions can be converted to the emission probabilities $P(x \mid q)$ required for HMMs as follows,

$$\frac{P(x \mid q)}{P(x)} = \frac{P(q \mid x)}{P(q)}, \tag{6.1}$$

where $P(q)$ is the prior probability of a phone. We assume that the prior probabilities of all phones are equal for decoding purposes. $P(x)$ is constant scaling factor for all classes and will not change during the classification. Thus, in practical systems, a scaled likelihood estimate is computed by MLP's output posterior probability.

The HMM has a temporal structure but the ANN has not. The outputs of the ANN nodes represent the emission probabilities for all the HMM states. Compared to the conventional HMM based ASR architectures, hybrid ANN/HMM systems have several advantages listed below:

- *Correlated Inputs*

  In HMM based ASR systems Gaussian mixture distributions are used to build the density function for acoustic models. Usually, features are assumed to be independent and no correlation between them is modelled, to keep the models as simple as possible. In hybrid ANN/HMM systems, on the other hand, the correlation between the features can be learned during the training of the ANN part.

- *Context Information*

  In HMM the first and second time derivatives are computed over a few adjacent frames to capture at least a limited amount of acoustical context information. Linear Discriminant Analysis (LDA) is used to maximize the inter-class variances and to minimize the inner-class variance. In the case of ANN/HMM, adjacent acoustic feature vectors $x_1, x_2, ..., x_n$ are given as input to the MLP, and the MLP provides a simple architecture to consider context information when calculating its output $P(q \mid x_1, x_2, ..., x_n)$. As an arbitrary function approximator, MLP allow any nonlinear transformations of the acoustic input.

- *Discriminative Training*

  In HMMs, we are concerned with estimating the joint probability distribution of the speech input. However, the ultimate goal of speech recognition is to compute the posterior probability of a sentence given the acoustic input. As a discriminative training method, MLPs may be used directly to compute class-conditional posterior probabilities. This computation is done in our hybrid ANN/HMM system at the frame level.

- *Flexibility*

  Using an MLP as an acoustic probability estimator provides us with an opportunity to combine diverse kinds of features. Therefore hybrid ANN/HMM architectures are well suited to integrate various levels of

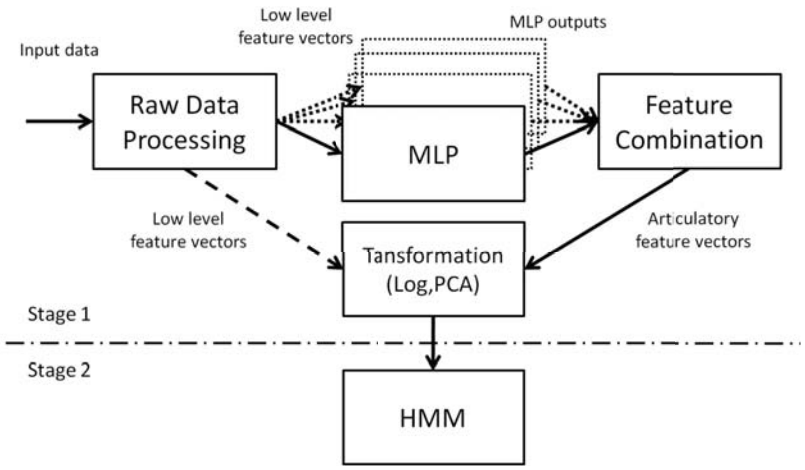


Figure 6.2: Architecture of a ANN/HMM approach for ASR. Low level feature vectors can also be combined with MLP outputs as indicated by the dashed line. Either a single MLP classifier or a range of parallel MLPs can be used depending on the number of articulatory features.

audio visual speech representations, such as, the low level feature vectors and articulatory feature vectors. This will be described in detail in this chapter.

6.3 Articulatory Feature Approach

In contrast to articulatory transcriptions, which retrieve the articulatory information from the transcription generation component, the articulatory feature is retrieved from the feature extraction component, which encodes the articulatory information. In conventional ASR, the feature extraction component usually computes low level features, for example, MFCCs for the audio channel and appearance-based ones for the visual input. ANNs attempt to transform input acoustic representations into compact but significant, low-dimensional representations which can be better modeled by the emission probabilities of the HMM than standard acoustic parameters. As introduced in Chapter 4, the ANN we used in our experiment is a multi-layer perceptron (MLP).

The architecture of an ANN/HMM system is shown in Figure 6.2. After calculating low level feature vectors, e.g. MFCC, MLPs can be used as nonlinear feature transformation components. The MLP output, which approximates the posterior probabilities for the given input features, can be further transformed into articulatory features for better matching the gaussian mixtures of a continuous HMM. This can be implemented, for example, by taking the logarithm of the MLP outputs. As an input for the HMM, either the articulatory feature vectors alone or a combination with low level feature vectors can be considered.

6.4 AF-based Tandem Approach

Many variations of the ANN/HMM approach have been proposed and showed comparable performance to conventional HMM based ASRs. Among these variations, (Hermansky et al., 2000) presents a method named the tandem approach which is suited for speech recognition tasks. The tandem approach uses the pre-nonlinear output of a neural network classifier as the input features for Gaussian mixture models (GMMs) of a conventional speech recognizer. The main difference to the conventional ANN/HMM approach is that the outputs of ANN are used as new features for HMM training rather as scaled emission probability directly.

As shown in Figure 6.3, an MLP is trained to estimate the posterior probabilities of possible subword units. The output of the MLP is used as input features for a Gaussian mixture based HMM system. Furthermore, in order to avoid skewed distribution from MLP, the input features for GM models are wrapped by certain transformations, such as logarithm or PCA (Hermansky and Morgan, 1994).

In the case of articulatory feature based AVSR, instead of using one single MLP classifier for posterior probability estimation, we train several parallel AF-based MLPs. Each individual MLP represents an AF feature channel, whose values are chosen according to knowledge about speech production. The number of units on the output layer of each MLP corresponds to the number of values in each articulatory feature. The definition of articulatory channels is the same as in Table 5.1. *Voicing, manner, place, front-back,* and *rounding* are five articulatory features for the acoustic channel, where the number of feature values per channel varies between 3 and 10. *Opening* and *Rounding* are two additional feature channels describing the position of the

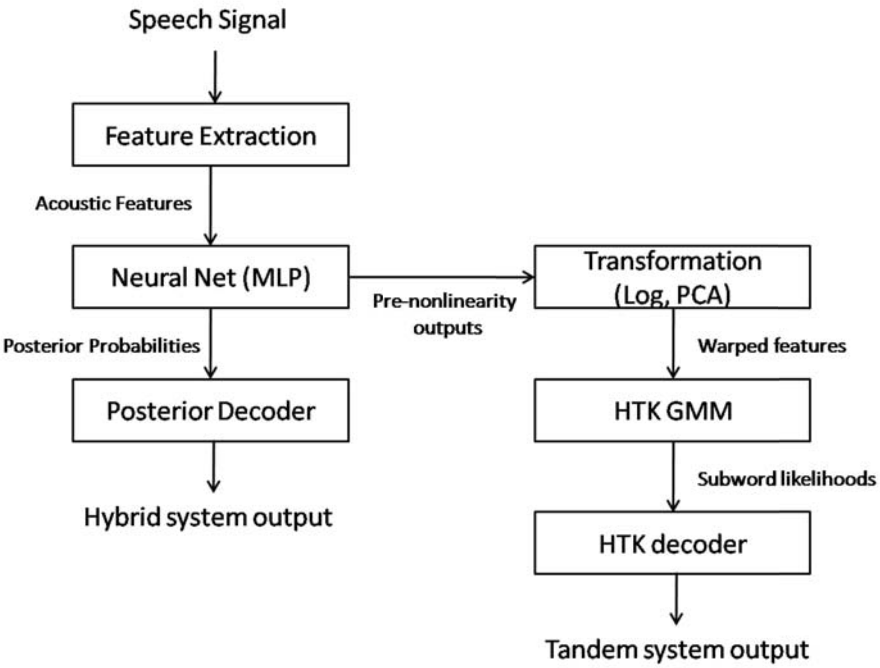


Figure 6.3: Architecture of a tandem approach for ASR. The left side represents the traditional hybrid ANN/HMM approach proposed by Bourlard. The right side describes the procedure of the tandem method.

articulators based on visual information.

Obviously, the number of feature values for each individual AF channel is much smaller than the number of classes in a usual phone inventory. This means that it requires less effort to train an MLP network, and better classification results can be expected. Furthermore, AFs might also help to achieve a better design for bimodal speech recognition, since using AFs as an intermediate abstract representation may provide an attractive option for combining information from the audio and visual channel.

According to the idea of AF-based tandem approach, Figure 6.4 shows a system framework consisting of two stages. Raw audio and video data are first processed in the feature extraction stage, where a series of parameterized feature vectors from both channels is generated. The feature classification stage transforms the feature vectors into articulatory features. MLPs with three layers are applied to train a set of articulatory feature models. As a second-level classifier, HMMs are applied in the second stage, which map the articulatory features into a sequence of different phonetic units.

Following (Kirchhoff, 1999), we have chosen the logistic function as activation functions for the nodes in the hidden layer and the softmax function for the nodes in the output layer. The softmax function ensures that the output activation values are non-linearly mapped to the range [0,1], which is necessary to provide the subsequent word recognizer with a valid probability distribution.

6.5 The Information Combination At Feature Level

In the AF based approach, the outputs of all MLPs are combined to generate a single articulatory feature vector as the input feature vector of a standard HMM-based speech recognizer. All the articulatory information is modeled only in the MLPs. On the feature level, feature vectors are always processed based on a specific time interval, such as 10ms as a time frame. Acoustic and visual low level features are used as input for the networks. For every

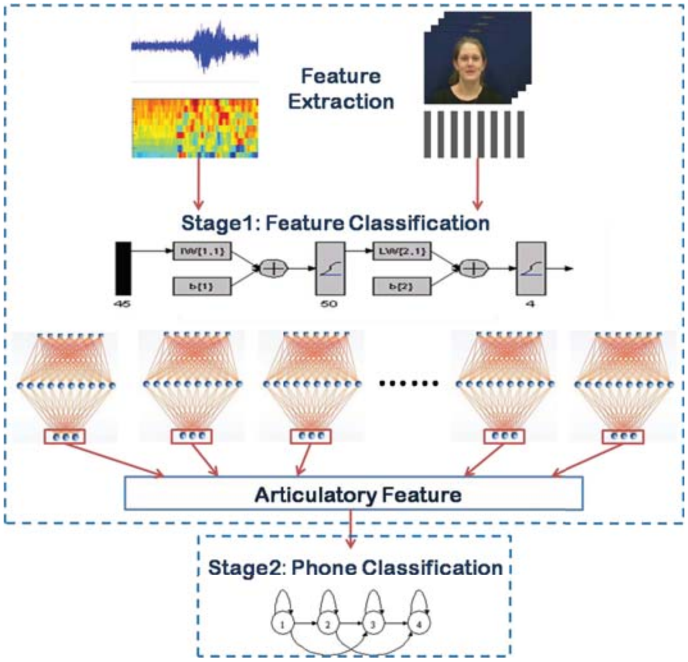


Figure 6.4: AF-based AVSR system.

time interval, a corresponding output vector is generated to represent the probability of the articulatory values in each channel for the current time frame. Although there are different channels of MLPs to retrieve various articulatory features, the combination of multiple articulatory features is still processed frame by frame. The synchronization of all information will not be achieved here. In the successive HMM classification stage, only one classifier is defined to run the word recognizer. After all the HMM decides on the feature vector to state mapping, i.e. whether a feature vector belongs to a phone internal state or rather indicates a phone transition. Since HMMs cannot model any asynchrony on a higher level, such as subphone, phone or word, we can only enforce synchronization for these information.

Using a combined articulatory feature vector is a solution to model the loosely synchronized articulatory gestures. All articulatory dynamics can be observed at the level of feature observation. Articulatory feature vectors describe the speech signal frame by frame. Instead of a sequence of phonetic units, an utterance can be explained by a composition of a sequence of frames. Phonetic units, such as subphones or phones alone, can hardly ex-

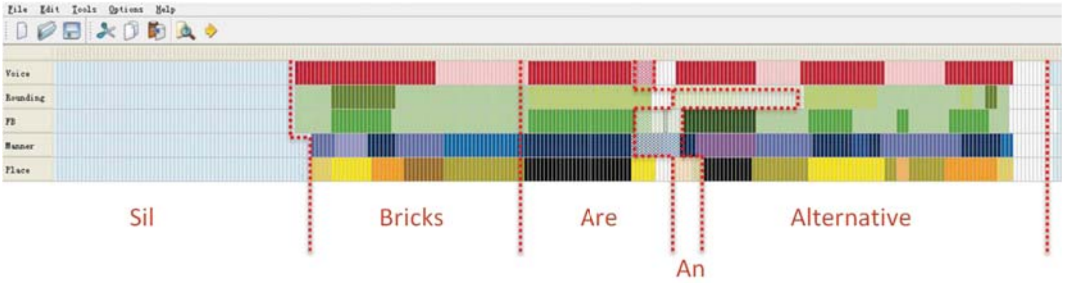


Figure 6.5: Alignment of AF Transcriptions.

press the pronunciation variability. However, the articulatory feature vectors for the frames can provide useful information to distinguish pronunciation variants for a phone. For example, the place of the velar for the phone /g/ in *gift* is more closer to the palatal than it is for the phone /g/ in *golf*. By using frames of articulatory feature vectors, the difference between changing from /g/ to /i/ and changing from /g/ to /o/ can be explicitly expressed.

An example of asynchronously aligned articulatory features is shown in Figure 6.5. The utterance “Bricks are an alternative” from the VidTIMIT corpus is analyzed using the force alignment graphical user interface. During feature extraction five channels of MLPs are used to extract the articulatory feature vectors. The combined feature vector is further concatenated with 39 dimensions of MFCCs. The data are then fed into five different articulatory features. In order to annotate the time information, we use the analyzing method similar to the one in Chapter 5. That is, instead of a representation based transcription for classification, five sets of articulatory transcriptions are used in these HMMs channels to train the word classifiers. Finally these force alignment results are presented in the figure to show temporal information in various channels.

Figure 6.5, shows an obvious asynchrony among the articulatory features at the word level. Some words, such as *silence* and *bricks*, maintain a roughly synchronized relation at the word boundary, which indicates a relative stable word recognition in all channels. However, some words, such as *an* and *alternative* don’t share similar word boundaries for all channels. Especially the boundary of the word “*an*”, exhibits a clear time difference between the channel for *rounding* and the other channels. By checking the raw audio data, we hear these two words as *a nalternative* rather than *an alternative*. This pronunciation variability, also known as coarticulation, is hard to be recognized because the model combination /a+n/ for the word *an* is badly trained. Therefore the boundary between *an* and *alternative* is placed differently by different articulatory classifiers.

6.6 Results and Conclusion

The raw audio signal was converted to a 9-dimensional RASTA-PLP (Hermansky et al., 1991) feature vector for each 10ms frame. The raw video signal was converted into appearance-based features as described in Chapter 3. In this experiment, the gray scaled pixel values are used as visual speech features. For dimensionality reduction, PCA has been applied and the first 20 components have been selected as visual features. Moreover, to raise the visual frame rate to that of the acoustic signal, visual vectors are interpolated linearly so that both signals are synchronously available.

For the MLP layer training stage, the 9-dimensional audio feature vectors and 20-dimensional visual feature vectors are then expanded to 45 dimensions and 100 dimensions by adding the two previous and the two following frame vectors in order to introduce some context information. These feature vectors were used as the input for all MLPs. The number of hidden units for the different MLPs is shown in Table 6.1.

Table 6.1: Number of hidden units used by each AF MLP.

Feature Channel	Hidden Units
voicing	50
manner	100
place	100
front-back	100
acoustic rounding	100
visual rounding	100
visual opening	50

In the second stage of our system, standard left-right mono-phone HMMs with 3 emitting states have been trained by means of HTK. The models are initialized with the flat start method according to the features calculated from the MLP layer and the HMM parameters are trained with the Baum-Welch algorithm.

To evaluate the performance of the system, first the preprocessed audio and visual data must be processed by the already trained MLP classifiers to calculate the posterior probabilities. The results are then combined and recognized with the Viterbi algorithm. All MLPs have been trained by means of MATLAB.

Table 6.2 shows a comparison of three AF-based feature extraction studies with respect to their frame level accuracy. For the purpose of this experiment the available data have been separated into a training set consisting of 330 utterances from the first 33 speakers and a test set with 100 utterances from the remaining speakers.

Table 6.2: Frame level AF-based MLP classification average accuracy.

Feature Channel	This Work	Work (Kirchhoff, 1999)
voicing	80.3%	89.1%
manner	54.6%	82.0%
place	47.7%	77.2%
front-back	66.8%	82.9%
acoustic rounding	71.5%	83.1%
visual rounding	58.7%	-
visual opening	61.2%	-

The *Manner* and *Place* feature channels have not reached a high accuracy is because of the large number of classes. To compare our results with the one of (Kirchhoff, 1999), one needs to take into account the different tasks (medium vocabulary vs. small vocabulary, namely OGI Numbers95) and the different corpus size (29 minutes vs. 160 minutes, in total including training and testing).

Figure 6.6 shows the WERs for different monophone systems on the Vid-TIMIT 1200-word vocabulary task. For the noise test sets, pink noise from Noisex database was added to the speech signal at various signal-to-noise

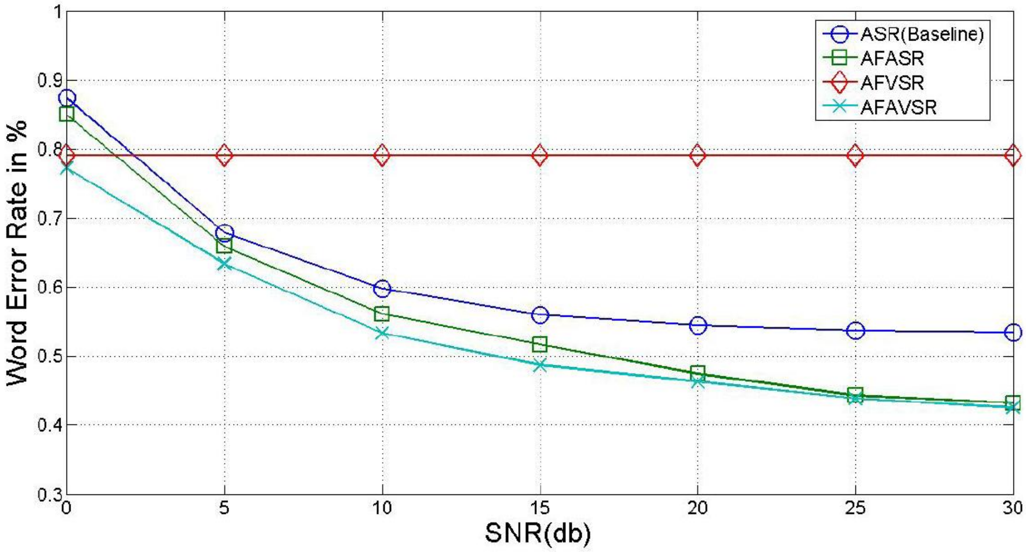


Figure 6.6: Comparison of the word error rates for different speech recognition system when pink noise is added.

ratios (SNR). Signals with 30dB SNR are assumed as clean ones. Two systems have only access to the acoustic channel and do not make use of any visual features: A phone-based audio-only speech recognizer (ASR) serves as a baseline, while AFASR is its counterpart based on articulatory features. The AF-based visual speech recognizer (AFVSR), on the other hand, has only access to the articulatory features from the visual channel. Its results therefore are independent of the acoustic noise level. The AFAVSR system, finally, combines all articulatory features from both channels, acoustic as well as visual ones.

Compared to the phone-based baseline, the AF-based tandem approach was able to achieve a significant reduction of the WER under all noise conditions. This is in contrast to the findings of (Kirchhoff, 1999), where the improvement was noticed only under noise, while in our case the less noisy conditions have profited most. That difference might be caused by the different amounts of training data available. The visual-only speech recognizer yields very poor results. It achieved a recognition rate of only 21% which is fairly low compared to 41% reported in (Kirchhoff, 1999). Nevertheless, by introducing the visual information, the combined AFAVSR system gained a noticeable improvement in accuracy compared to the audio-only recognizers. This again confirms that the use of visual information can compensate deficits of the audio channel.

In contrast to the articulatory transcription approach, the articulatory feature based one has two advantages. Firstly, the articulatory feature approach is not corpus dependent. There is no need to make special assumptions about the characteristics of the corpus data as for the articulatory transcription approach. Therefore, the approach can be scaled up to arbitrary AVSR tasks. Secondly, by estimating the probabilities of articulatory values, articulatory features together with low level features are more robust against noise (Figure 6.6). These results are in accordance with the motivation of using articulatory information in AVSR.

Although the combined articulatory feature vectors potentially show an improvement of AVSR, in practice it is still difficult to retrieve accurate articulatory feature vectors for each frame. As shown in the previous section, the accuracy of articulatory feature vector extractors is between 50% and 80%. When one specific articulatory feature in a particular time frame has been inaccurately processed, the combined articulatory feature vector cannot properly express the articulatory dynamics in that frame. Furthermore, the articulatory feature vectors are combined and synchronized at the frame level, which is a rigid synchronization requirement between different articulatory features. As discussed in Chapter 2, articulatory information is asynchronously arranged. However, this asynchronicity between articulatory features is difficult to be modeled using this approach.

Chapter 7

Encoding Articulatory Information with DBN Models

As described in Section 4.4, HMMs have the problems of interpretation and factorization, which presents one state at a time. This characteristics prevents us to model the details of articulatory information in AVSR. As an alternative, DBN is a generalization of HMM, which allows them to use arbitrary many state variables, therefore being able to model independent movements of articulators. We use DBN as the formal foundation in this chapter to design different systems for encoding articulatory information from audio and visual modalities. Specifically, a single channel AVSR, an audio-visual channel AVSR and an articulatory channel AVSR are described as three different types of DBN based systems.

7.1 Data Preparation

In contrast to the VidTIMIT corpus, the GRID is based on a smaller vocabulary, but provides more training and testing data. Furthermore, the language model of the GRID corpus is rather simple. Therefore, the GRID corpus is used for all experiments in this chapter. 5000 sentences (5 speakers) are selected as training data and 1000 sentences (1 speaker) are used for testing.

Again, MFCCs with a 10ms sampling period are taken as the acoustic feature vectors. The visual feature vectors are computed by the appearance based method with a 40ms sampling period. By upsampling the visual features to one per 10ms with linear interpolation, acoustic and visual observations can be combined for each state. The different DBN models have been developed using GMTK (the graphical model toolkit, (Bilmes and Zweig, 2002)).

For AVSR the audio and visual channel were designed to model phone and viseme features respectively. 35 phones and 13 visemes have been modeled based on acoustic and visual observations. For the models in the articulatory channel, it would be ideal to have seven channels available, including five audio and two visual articulatory features. Unfortunately it turned out to be infeasible to train models with a seven channel structure. Therefore, we designed a three channel feature set for the articulatory channel AVSR. The details are explained in Section 7.4.

7.2 Single Channel AVSR

The single channel AVSR uses only one phonetic channel. Figure 7.1 shows the training structure used in the system. In this graph we model three levels of phonetic units, namely $Word$, $Phone$ and $SubPhone$. A word is constructed out of several phones and each phone should contain three states of sub-phones. A $State$ variable is defined to directly map the probability values of $SubPhone$. The $Observation$ variable in the baseline system is designed for reading one stream of acoustic features, eg. MFCC, PLP, etc. It would be easy to consider multiple observation variables, eg. acoustic and visual features, as long as they share one common state variable. In this case, multiple streams are synchronized at the state level.

The meaning of all variables in Figure 7.1 and their Conditional Probability Distributions (CPDs) are explained as follows (Ravyse et al., 2006).

- $WordCounter$, denoted as WC, represents the word position in the current sentence.
  $P(WC_t = j \mid WC_{t-1} = i, WT_{t-1} = b, W_{t-1} = w)$
  $$= \begin{cases} 1 \quad if \quad j = i+1 \quad and \quad b = 1 \quad and \quad w \neq EOU \\ 1 \quad if \quad j = i \quad and \quad b = 0 \quad and \quad w \neq EOU \\ 1 \quad if \quad b = 1 \quad and \quad w = EOU \\ 0 \quad Otherwise \end{cases}$$

  We use here the term cardinality of a variable to represent the number

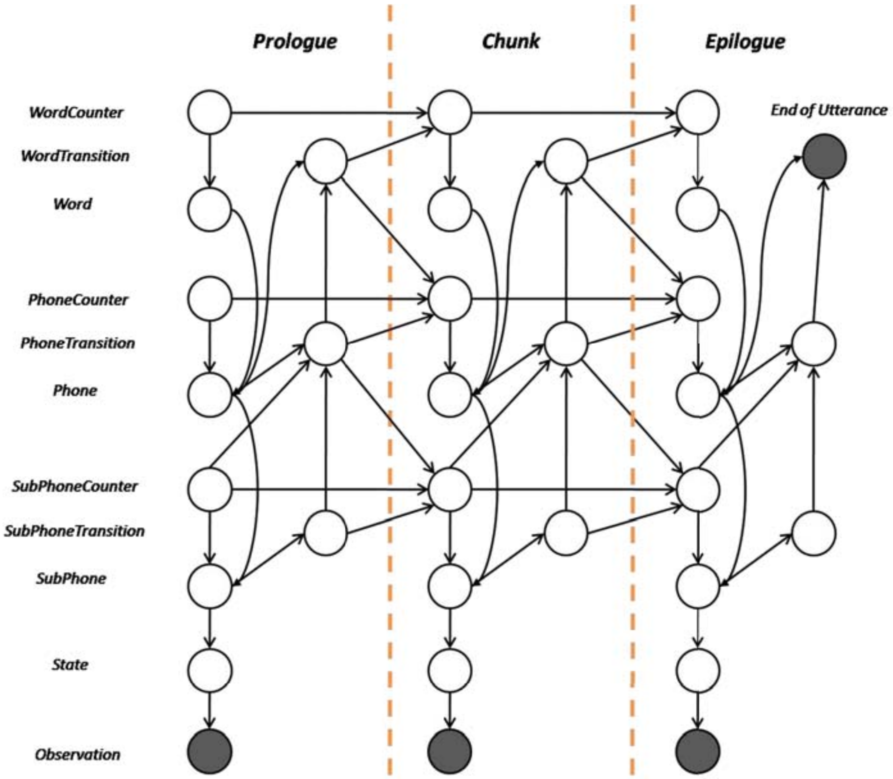


Figure 7.1: A 3-state monophone model for training without word alignment.

of values of the specific variable. For example, the cardinality of WC is the maximum possible number of words in an utterance. The value of WC is increased by one when a word transition happens and the previous word is not the end of the utterance.

- $WordTransition$, denoted as WT, represents the end of the current word and a transition to the next one.

$$P(WT_t = j \mid PT_t = i, P_t = p)$$
$$= \begin{cases} 1 & if \quad p \neq EOW \quad and \quad i = 1 \\ 0 & Otherwise \end{cases}$$

WT is set to 1 if there is a phone transition and the current phone is not the end of the word (EOW).

- $Word$, denoted as W, refers to the current word, determined by word counter.

  $P(W_t = w \mid WC_t = i)$
  $$= \begin{cases} 1 & if \quad w = words(i) \\ 0 & Otherwise \end{cases}$$

  $words(i)$ returns a word which is the ith word in the current utterance.

- $PhoneCounter$, denoted as PC, represents the phone position in the current word.

  $P(PC_t = j \mid WT_{t-1} = d, PC_{t-1} = i, PT_{t-1} = b)$
  $$= \begin{cases} 1 & if \quad j = i+1 \quad and \quad b = 1 \quad and \quad d = 0 \\ 1 & if \quad j = i \quad and \quad b = 0 \quad and \quad d = 0 \\ 1 & if \quad j = 1 \quad and \quad b = 0 \quad and \quad d = 1 \\ 0 & Otherwise \end{cases}$$

  The cardinality of PC is the maximum possible number of phones in a word. The value of PC is increased by one when a phone transition happens and the previous phone is not the end of the word. If there was a word transition in the previous frame, the phone counter should be set to 0, which means the beginning of a new word.

- $PhoneTransition$, denoted as PT, refers to the end of the current phone and a transition to the next one.

  $P(PT_t = j \mid P_t = p, SPC_t = b, SPT_t = i)$
  $$= \begin{cases} 1 & if \quad j = 0 \quad and \quad i = 0 \\ 1 & if \quad j = 1 \quad and \quad i = 1 \quad and \quad b = laststateof(b) \\ 1 & if \quad j = 0 \quad and \quad i = 1 \quad and \quad b \neq laststateof(b) \\ 0 & Otherwise \end{cases}$$

  PT is set to 1 if there is a sub-phone transition and the current sub-phone is the end of the phone. A boolean function $laststateof(b)$ checks whether b is the last sub-phone state of the current phone.

- $Phone$ is the current phone, denoted as P.

  $P(P_t = p \mid PC_t = i, W_t = w)$
  $$= \begin{cases} 1 & if \quad p = PhoneInWord(i, w) \\ 0 & Otherwise \end{cases}$$

$PhoneInWord(i, w)$ returns the ith phone in the word w.

- $SubPhoneCounter$, denoted as SPC, describes the sub-phone position in the current phone.

  $$P(SPC_t = j \mid PT_{t-1} = d, SPC_{t-1} = i, SPT_{t-1} = b)$$
  $$= \begin{cases} 1 \quad if \quad j = i+1 \quad and \quad b = 1 \quad and \quad d = 0 \\ 1 \quad if \quad j = i \quad and \quad b = 0 \quad and \quad d = 0 \\ 1 \quad if \quad i = 1 \quad and \quad b = 0 \quad and \quad d = 1 \\ 0 \quad Otherwise \end{cases}$$

  The cardinality of SPC is the maximum possible number of sub-phones in a phone. The value of SPC is increased by one when a sub-phone transition happens and the previous sub-phone is not the end of the phone. If there was a phone transition in the previous frame, the sub-phone counter should be set to 0, which means the beginning of a new phone.

- $SubPhoneTransition$ indicates when the model should advance to the next sub-phone, denoted as SPT. It is initialized with a non-deterministic conditional probability table.

- $SubPhone$, denoted as SP, refers to the current sub-phone.

  $$P(SP_t = sp \mid SPC_t = i, P_t = p)$$
  $$= \begin{cases} 1 \quad if \quad sp = SubPhoneInPhone(i, p) \\ 0 \quad Otherwise \end{cases}$$

  $SubPhoneInPhone(i, p)$ returns the ith sub-phone in the phone p.

- $State$, denoted as S, is a variable to map the $SubPhone$ variable.

- $Observation$, denoted as O, is a variable to keep the observation feature vectors.

- $EndOfUtterance$, denoted as EOU, is an observed value implying that the utterance ends at this frame.

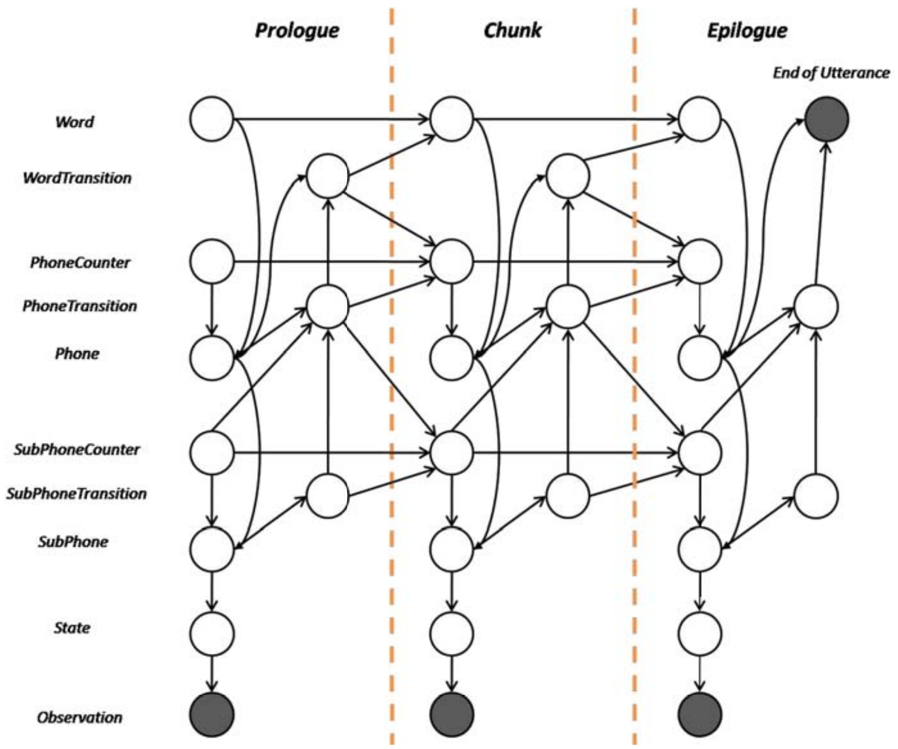


Figure 7.2: A 3-state monophone model for decoding with a bigram language model.

Figure 7.2 shows the structure of a standard phone based decoder with a bigram language model. In the case of decoding only the observation sequence $O_{1:T}$ is known whereas the underlying state sequence $S_{1:T}$ and their parents are hidden. The required likelihood is computed by summing over all possible state sequences, as shown in Equation 7.1.

$$\begin{aligned} P(O_{1:T}) = & \sum_{W_{1:T},WT_{1:T},PC_{1:T},PT_{1:T},P_{1:T}SPC_{1:T},SPT_{1:T},SP_{1:T},S_{1:T}} \\ & P(W_{1:T}, WT_{1:T}, PC_{1:T}, PT_{1:T}, P_{1:T}SPC_{1:T}, SPT_{1:T}, SP_{1:T}, S_{1:T}O_{1:T}) \end{aligned} \tag{7.1}$$

According to the chain rule and the conditional independence assumptions of Figure 7.2, equation 7.1 can be simplified (Equation 7.2), where each factor can be computed using a conditional probability distribution from the learned model parameters.

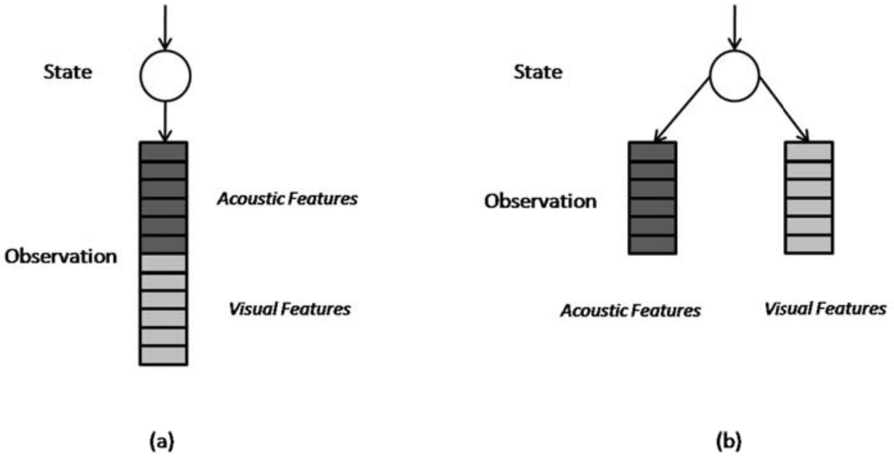


Figure 7.3: Audio visual information integration in DBN. (Bilmes, 2010)

$$
\begin{aligned}
&P(W_{1:T}, WT_{1:T}, PC_{1:T}, PT_{1:T}, P_{1:T}SPC_{1:T}, SPT_{1:T}, SP_{1:T}, S_{1:T}O_{1:T}) \\
&= \prod_t P(O_t \mid S_t)P(S_t \mid SP_t) \\
&\times P(SPT_t \mid SP_t)P(SP_t \mid P_t, SPC_t)P(SPC_t \mid SPC_{t-1}, SPT_{t-1}, PT_{t-1}) \\
&\times P(PT_t \mid SPC_t, SPT_t, P_t)P(P_t \mid PC_t, W_t)P(PC_t \mid PC_{t-1}, PT_{t-1}, WT_{t-1}) \\
&\times P(W_t \mid W_{t-1}, WT_{t-1})P(WT_t \mid P_t, PT_t)
\end{aligned}
\tag{7.2}
$$

Feature combination is the simplest way to implement an AVSR. A conventional 3-state monophone DBN model like the one in Figure 7.2 can be directly used as audio visual speech recognizer, except that the acoustic and visual feature vectors have to be combined at the observation level. Using the structures explained above, two ways of combination are available as shown in Figure 7.3. Figure 7.3(a) is to simply concatenate the two feature vectors. Figure 7.3(b) uses a factored model to combine audio and visual information. The advantage of method (b) is that a mixture coefficient (weight) for the Gaussian components can be predefined by the user.

Table 7.1 shows a comparison of three single channel DBN based recognition systems with respect to their accuracy. These systems mainly differ in the design of the observation level variables. We use “A” and “V” to denote the audio and visual channels. “L” and “H” represent the low level feature

Table 7.1: Recognition rates of single channel DBN systems

Input observations	Recognition Rates
AL	43.57
AL+VL	56.39
AL+VL+AH+VH	58.64

vectors and higher level articulatory feature vectors respectively. The same abbreviations are also used in Table 7.2 and 7.4. The first system uses only the audio low level feature vectors as the input, which realizes an audio only speech recognition system. The second system concatenates the audio and the visual low level feature vectors together to implement an audio visual speech recognition system. The third system further extends the feature vector by adding the articulatory features for audio and vision thus yielding an articulatory information based AVSR. The results show that the AVSR outperforms audio-only ASR with a recognition rate improvement of 12.8%. The articulatory information based AVSR can further improve the recognition rates by 2.3% compared to the AVSR using only the low level features. These results indicate that the combined use of low level and articulatory features at the observation level can lead to an improvement of AVSR.

7.3 Audio-Visual Channel AVSR

In contrast to the single channel DBN model above, audio and visual features can also be encoded by means of two channels of a graphical model where each channel uses the same structure as the single channel model depicted in Figure 7.2. Figure 7.4 shows the training structure of our audio visual multi-channel speech recognizer. The audio and the visual channel share the same word level variables, usually the word counter variable WC, the word variable W and the word transition WT. This structure ensures that the audio and visual channels are synchronized at the boundaries of each word. This constraint is implemented at the variable word transition variable WT, which has four parent nodes two from the audio and two from the visual channel. Only if both phone variables in the two channels reach the last phone position of a word, and both phones are allowed to transit, the word transition can be set to the value 1. WT, which is the parent of the phone counter variables from both channels, will reset both phone counters to 0 if

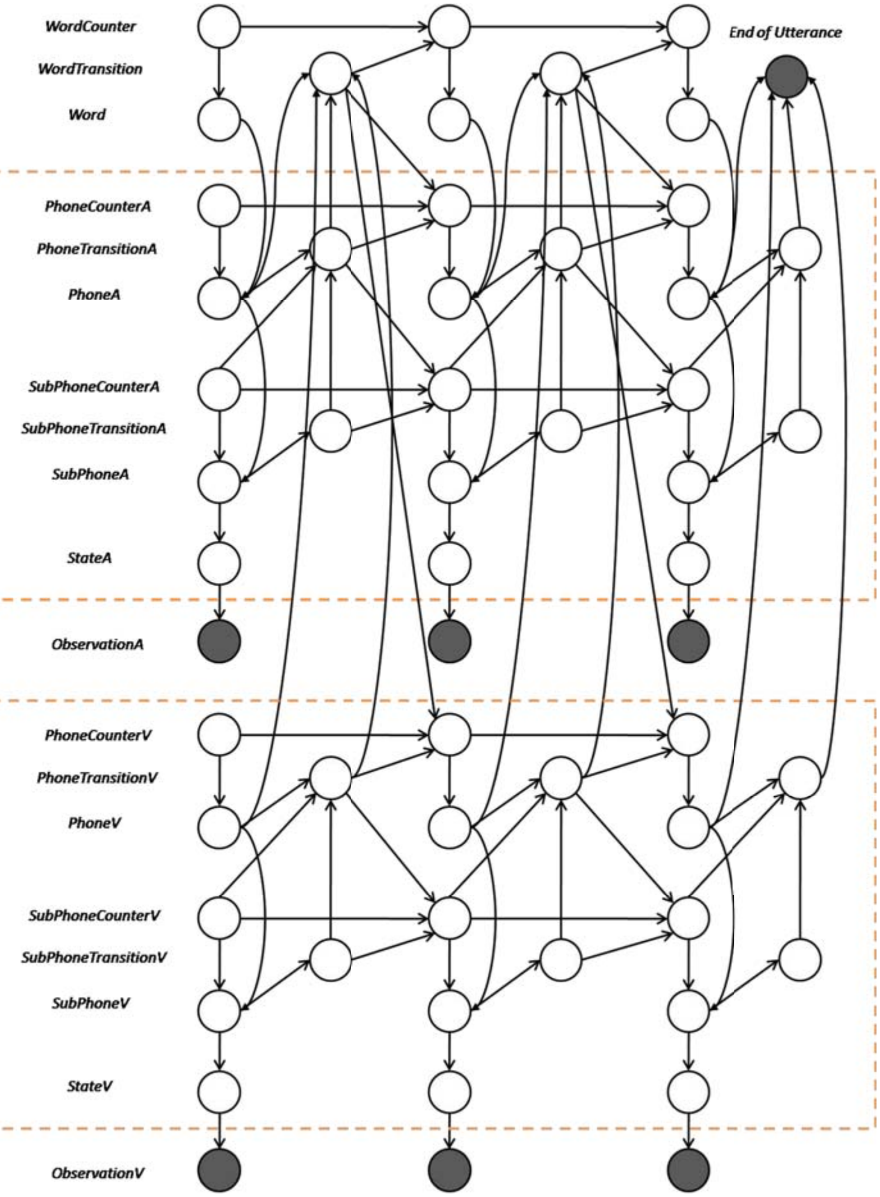


Figure 7.4: The training structure of the audio visual multi-channel speech recognition model.

a word transition happens.

The word transition considers information from both the audio channel and the visual channel. On the other hand, it affects the values of both the phone counter PCA in the audio channel, and PCV in the visual channel. The conditional probability distribution of a word transition variable is defined as,

$$P(WT_t = wt \mid PA_t = i, PV_t = j, PTA_t = m, PTV_t = n)$$
$$= \begin{cases} 1 & if \quad m = 1, n = 1, i = lastphone, j = lastphone \\ 0 & Otherwise \end{cases} \tag{7.3}$$

where PA and PV represent the phone variables of the audio and visual channel respectively. PTA and PTV are their phone transition variables. *lastphone* is a boolean function which is always true at the end of a word.

Within a word, all the variables of the audio and visual channels are independent of each other at different levels, such as phone level, subphone level and observation level. This independence makes it possible to model the asynchronicity between audio and visual information at different levels. On the phone level, for example, a transition of a phone in the audio channel depends only on the phone instance of the same channel.

Figure 7.5 illustrates another audio-visual speech recognition model named a coupled HMM (CHMM) (Nefian et al., 2002). The CHMM is a DBN model that allows some parent variables in each channel to interact, and at the same time to have their own observations. In CHMM, the audio and visual channels still have their independent observations. However, an increment of the phone counter variable from one channel is dependent on the phone transition variable of the other one. Figure 7.6 shows the decision tree of the phone counter variable for both, the audio and the visual channel.

In contrast to the model in Figure 7.4, where audio visual information is synchronized at the word level, the audio visual CHMM in Figure 7.5 constrains the synchronization at the boundaries of each phone. As shown in Figure 7.6, the phone counter in the audio channel PCA is increased by one, only if there is no word transition ($WT = 0$) but a phone transition in each channel

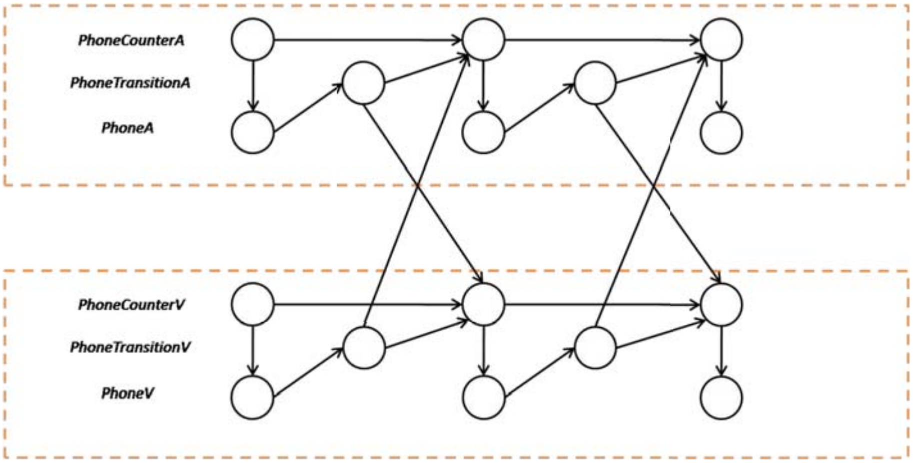


Figure 7.5: A coupled HMM (CHMM) audio visual speech recognition model. Only the phone level is shown. The variables of the other levels, e.g. word, subphone and observation levels, are the same as the variables in Figure 7.4 and not drawn here.

($PTA = 1$ and $PTV = 1$). The phone counter variable further affects its corresponding phone and phone transition variables, which makes the phone transition in each channel to be changed synchronously.

Table 7.2: Recognition rates of audio-visual channel DBN systems

Structure	Input observations	Recognition Rates
Asynchronous	AL $\bigcup$ VL	68.75
Asynchronous	AL+AH $\bigcup$ VL+VH	73.33
Coupled	AL $\bigcup$ VL	67.36
Coupled	AL+AH $\bigcup$ VL+VH	70.83

Table 7.2 presents the recognition rates of different audio-visual channel DBN systems. All the experiments are based on the GRID corpus and use the two channel architecture. The main differences are their training structures and the input observations. In particular, the first two systems are based on the asynchronous structure where the information fusion happens at the word level. We see an improvement while using articulatory features and low level features together. The same result can be observed by comparing the third and the fourth system where a coupled structure was used to integrate information at the phone level. It is interesting to see that the asynchronous

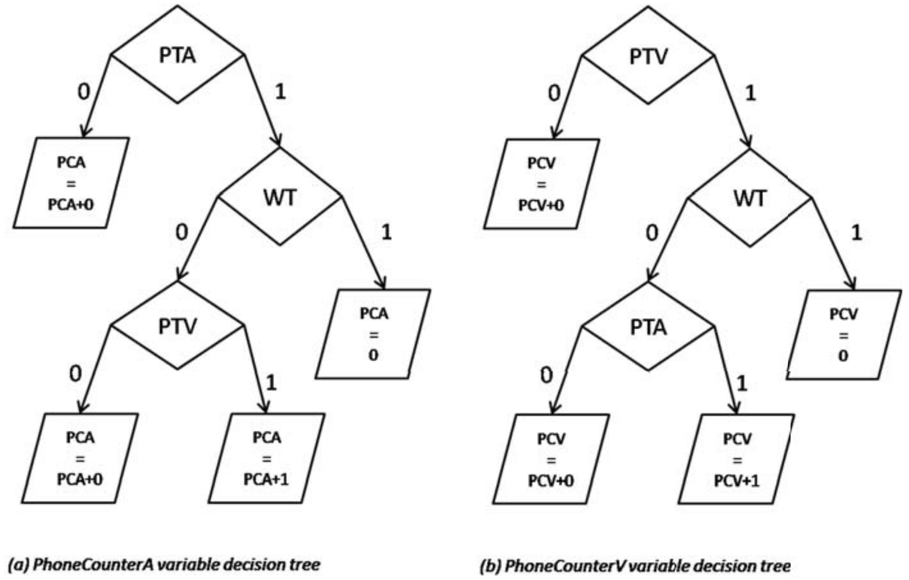


Figure 7.6: Decision trees for the variables $PhoneCounterA$ and $PhoneCounterV$.

structure based systems outperformed the coupled ones. These two structures differ in the level where information fusion is achieved. Obviously, the word level synchronization in the asynchronous models provides more freedom to deal with the asynchronicity than the stronger synchronization requirements of the coupled ones. This result indicates that coupling on the phone levels is too rigid and information fusion and synchronization should be attempted at a higher level.

7.4 Articulatory Channel AVSR

In the previous section, the model structures for an audio visual speech recognizer have been based on a multi channel DBN. In the articulatory models, this multi channel DBN structures will be further developed.

As shown in Figure 7.7, the model structure is composed of three channels, where each channel models a set of articulatory feature values. For the design of these three meta-features two basic principles have been adopted. Firstly, the resulting meta-features should be independent of each other. This principle avoids the correlation among different feature values. For example, if a

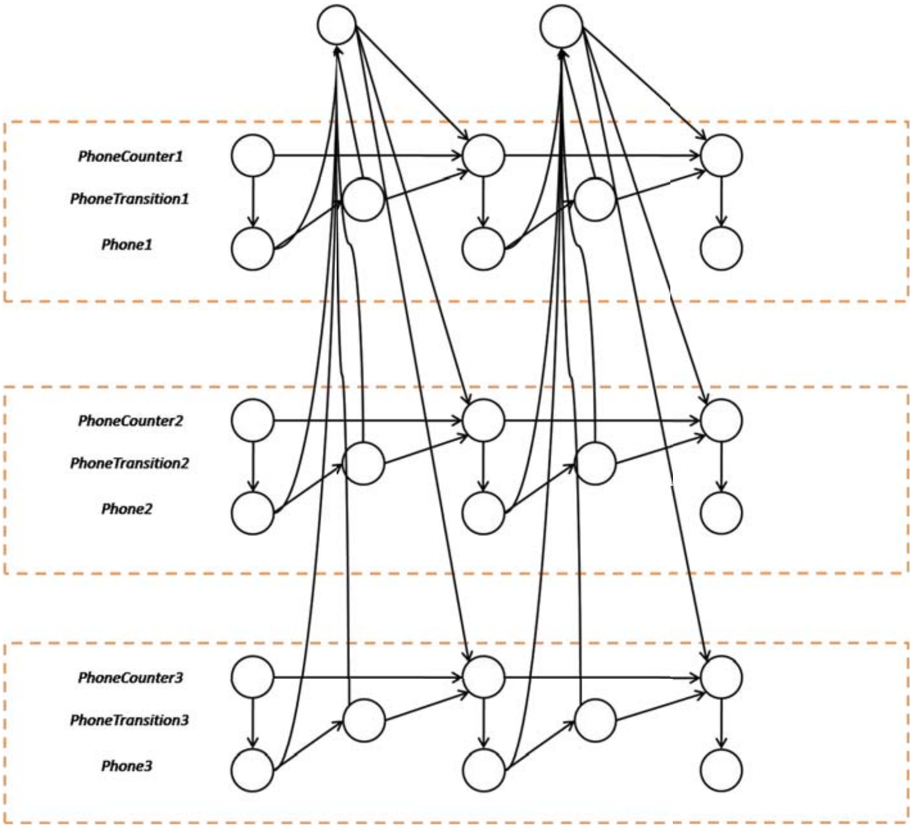


Figure 7.7: A three channel articulatory speech recognition model. Only the phone level is shown. The variables on the other levels, e.g. word, subphone and observation levels, are the same as the variables in Figure 7.4 and not drawn here.

phone has the value *unvoiced* in feature *voicing*, it must have a mapping with the value *nil* in feature *rounding*. That is, the "Voicing" and "Rounding" features are dependant to each other. Therefore, they are good candidates for being combined into one feature in order to reduce the number of channels. Secondly, there must be a combination of audio and visual articulatory channels in the model. For the experiments we have chosen two audio articulatory channels and one visual articulatory channel. The next subsection describes these articulatory feature channels in more detail.

Based on the articulatory feature set defined in Table 5.1, an articulatory feature set consisting of three meta-features has been defined in Table 7.3.

Table 7.3: Articulatory Meta-features used in different DBN channels.

Meta-Feature	Values	Cardinality
MaPl	nasal-coronal, nasal-labial, stop-coronal stop-labial, vowel-high, vowel-mid, vowel-low, approximant-retro, approximant-labial approximant-high, fricative-coronal fricative-labial, fricative-high fricative-dent, lateral, velar, sil	17
RoFB	round, nil-voiced, nil-unvoiced, flat-front, flat-back, sil	6
ViRo	round, nil, flat, sil	4

The meta-feature $MaPl$ is a combination of the *manner* and *place* features from Table 5.1. In the *manner* feature, only *lateral* and *velar* have not been changed. The other values *vowel*,*nasal*, *stop*, *approximant* and $fricative$ have been modified with additional information from the *place* feature. For example, the old *nasal* and *stop* values have been divided into four values by combining them with the information of the feature *place*. If only the *manner* feature is considered, the phones /*n*/ and /*m*/ have the same value for *nasal*. But these two phones can be distinguished by means of two composite values as $nasal - coronal$ and $nasal - labial$. Similarly, the *stop* value from the feature *manner* can be also divided into two values as $stop - coronal$ and $stop - labial$ to distinguish the phones /*b*/ and /*d*/, or /*t*/ and /*p*/.

The meta-feature $RoFB$, is a combination of the features *rounding*, $front - back$ and *voicing*. Similar to the values in the meta-feature $MaPl$, the value $flat$ is divided into the values $flat - front$ and $flat - back$ by combining the features *rounding* and $front - back$. The value *nil* is divided into the values $nil - voiced$ and $nil - unvoiced$ by combining features *rounding* and *voicing*. The detailed mapping between phone to these articulatory meta-features are shown in Appendix A.2.

The third channel is based on the visual articulatory information. Since the *visualopen* feature can be viewed as a subset of the *visualrounding* feature (ViRo), the $ViRo$ feature is used as articulatory information which can be observed by visual cues.

Table 7.4: Recognition rates of articulatory channel DBN systems

# Channels	Input observations	Recognition Rates
2	AL+AH(MaPl) $\bigcup$ AL+AH(RoFB)	34.67
2	AL+AH(MaPl) $\bigcup$ VL+VH(ViRo)	31.36
2	AL+AH(RoFB) $\bigcup$ VL+VH(ViRo)	8.33
3	AL+AH(MaPl) $\bigcup$ AL+AH(RoFB) $\bigcup$ VL+VH(ViRo)	39.35

Table 7.4 presents the recognition rates of different articulatory channel based DBN systems. The third one is the worst among all the systems, because the "rounding" information from both channels is redundant. Moreover, only a partial representation of articulatory information is used in this system. There is no synergy from all articulatory channels. The second system performs better because complementary information is available. The first system uses only the acoustic information. Since the acoustic channel "RoFB" is in general more reliable than the visual channel "ViRo", the acoustic-only system outperforms the second acoustic-visual system. The last system uses all three channels. Obviously, the combination of two acoustic and one visual articulatory channel in this system provides a more comprehensive model for articulatory information based AVSR.

Although our experiments show that DBN approaches provide improvements in modeling articulatory based AVSR, training a DBN is quite expensive compared with training HMMs, primarily because training the DBN requires assumptions about the maximum number of phones in a word and the maximum number of words in a sentence. We believe our work on DBN based AVSR is only the first step towards more powerful articulatory information models, and many issues remain to be resolved.

Chapter 8

Summary and Discussion

In this chapter we will give a summary of the work in this thesis and compare the results from our experiments to the initial motivation of using articulatory information in AVSRs. Finally, we conclude the findings and indicate possible future works in the area of articulatory information based AVSR.

8.1 Summary and Discussion

This thesis is worked in the CINACS (Cross-modal Interactions in Natural And Cognitive Systems) international research training group. CINACS focuses on the topics of cross-modal interactions and integration, such as mechanisms of multisensory perception and attention, cross-modal learning and association, multimodal representations for communication. Our thesis presents the research work in the area of bimodal speech recognition.

"Bimodal" refers to the acoustic and the visual modality, where the audio channel takes human speech as input while in the video channel continuously extracted lip information was dealt with. A motivation of this idea comes from the famous McGurk effect (McGurk and MacDonald, 1976), which demonstrates an interaction between hearing and vision in speech perception. We began the study by analyzing different components of an AVSR system. Various low level feature extraction methods in both channels were studied first. Then we reviewed different types of information fusion and classification methods. It was found that visual cues, namely the movements of lips, provide a means to directly observe articulatory information. Therefore, inspired by human speech production, we started to focus on the problem of using different sources of articulatory information in both channels to build more accurate speech recognition systems.

According to the different stages at which articulatory information can be used in ASR, we distinguished the following four approaches: 1) Using articulatory raw data describing the true geometry of the articulators, 2) Using articulatory transcriptions in an HMM/N-best decision framework, 3) Using articulatory features in an ANN/HMM framework and 4) Encoding articulatory information by means of DBN models.

The first approach directly observes the movements of articulators. Although this type of information naturally represents the fundamental processes of articulation, they are inconvenient to be recorded and used for AVSR. Therefore, this information has not been used in our work.

The idea of the second approach was to organize a bundle of parallel phone-based word recognizers in our system. These classifiers have the same input data, namely acoustic or visual speech signals, but use various articulatory transcriptions as targets. The recognized articulatory decisions are further integrated by a decision fusion component, which realizes an N-best decision schema. The reduced number of classes in each classifier helps to improve the recognition accuracy of the individual channels. A comparison with two other AF-based recognition systems shows that our work is competitive with respect to the accuracy. Adding two visual channels further improved ASR performance. Compared to the baseline system word recognition accuracy raised from 90.34% to 93.57% and 93.71% in the audio-only and audio-visual AF-based recognition respectively.

The third approach uses the articulatory information generated by a bottom-up feature extraction component. Articulatory features are extracted from low level acoustic and visual data using dedicated MLP classifiers. A combination of articulatory results and low level features is used as new inputs for a conventional phone-based HMM recognizer. Different monophone systems have been tested on the VidTIMIT 1200-word vocabulary task in audio-only, visual-only and audio visual mode. In contrast to the phone-based acoustics-only speech recognizer, the AF-based approach significantly improved the recognition rates. Its performance was further improved by adding the two articulatory features from the visual channel, although their reliability if used in isolation is still fairly low.

The last approach used DBN models to integrate articulatory information. In particular, a single channel AVSR, an audio-visual channel AVSR and an articulatory channel AVSR have been studied. The single channel AVSR uses only one phonetic channel by representing the conditional dependencies between different variables. Articulatory information can only be combined at the state level. The audio-visual channel AVSR used two channels to represent audio information and visual information. Both information sources have been modelled using low level features and articulatory features. Ideally, the articulatory information can be fused at the phone and word level. The articulatory channel AVSR should be modeled with a complete set of articulatory channels. However, due to computational restrictions the seven articulatory channels had to be reduced three. Therefore, the results are of limited use but show that articulatory information can contribute to an improvement of AVSR when used together with low-level features.

The results of our experiments with different articulatory information systems can be further analyzed from another point of view.

If we change the perspective from using articulatory information to that of applying information fusion, we can identify three methods for integrating audio and visual information. Early fusion (Potamianos et al., 2004) can be used for the articulatory information generated by feature extraction. Audio and visual articulatory features are simply concatenated after feature extraction. Decision fusion (Potamianos et al., 2004) can be applied in the system with articulatory information retrieved from transcription. By using the N-best decision schema, the recognized auditory and visual articulatory gestures are combined and mapped to their corresponding phones. Fusion within the model can be achieved by means of DBNs where dependencies between auditory and visual articulatory gestures are modelled.

Irrespective of where the information fusion takes place, the synchronicity problem between the parallel channels of articulatory information always needs to be addressed. In the system which uses articulatory information as features, we combine the acoustic and visual information at the frame level. In the system where the articulatory information was generated from transcriptions, decision fusion is achieved in a loosely synchronized way. The articulatory gestures are fused at the phone level. For the system with articulatory information in the models, we also used a loosely synchronized method, but the fusion happens among several frames.

In the thesis, articulatory features are used as an intermediate abstract representation between the acoustic signal preprocessing level and the subword unit probability estimation level. By means of integrating articulatory features into various AVSR systems, we have shown that articulatory information can be used for improving the performance of audio visual information fusion.

Furthermore, we analyzed the synchronicity problems at different phonetic levels (e.g. subphone, phone, word level). The results have shown that it is better to use loose synchronization of articulatory information at the word level. Too strong synchronization and asynchronous information may negatively affect the accuracy of the system.

Finaly, we tried to modelling the multi-channel information fusion using the graphical models. In contrast to conventional HMM based AVSRs, the DBN approaches provide the possibility to define different variables which might play important roles in AVSR information fusion. The results of DBN based systems have also shown that the audio-visual articulatory information helps to improve the performance in compared to acoustic only information.

Although all approaches to articulatory information based AVSR show possibilities for improving recognition accuracy, there are still some drawbacks to be discussed.

The articulatory transcription approach has the limitation of scalability. The HMM/N-best decision approach is corpus dependent. Two assumptions of the N-best decision schema prevent this approach to scale up to arbitrary speech recognition tasks. If the speech corpus is based on a large size vocabulary or a more freely defined grammar, the output of the first stage HMM classifiers could result in deletion or insertion errors. It becomes difficult to combine the articulatory information in an N-best decision schema.

In the articulatory feature approach, it is not easy to robustly retrieve accurate and complete articulatory feature vectors for each frame. The accuracy of articulatory feature vector extractors lies between 50% and 80%. When one specific articulatory feature in a particular time frame has been inaccurately processed, the combined articulatory feature vector cannot properly express the articulatory dynamics in that frame.

The articulatory modeling approach is flexible enough to model loose synchronization at different phonetic levels. However, it is difficult to scale up due to computational resource limitations. Moreover, the approach is also sensitive to data sparseness and the details of synchronization between the channels. This leads to difficulties in comparing recognition results across different AVSR systems.

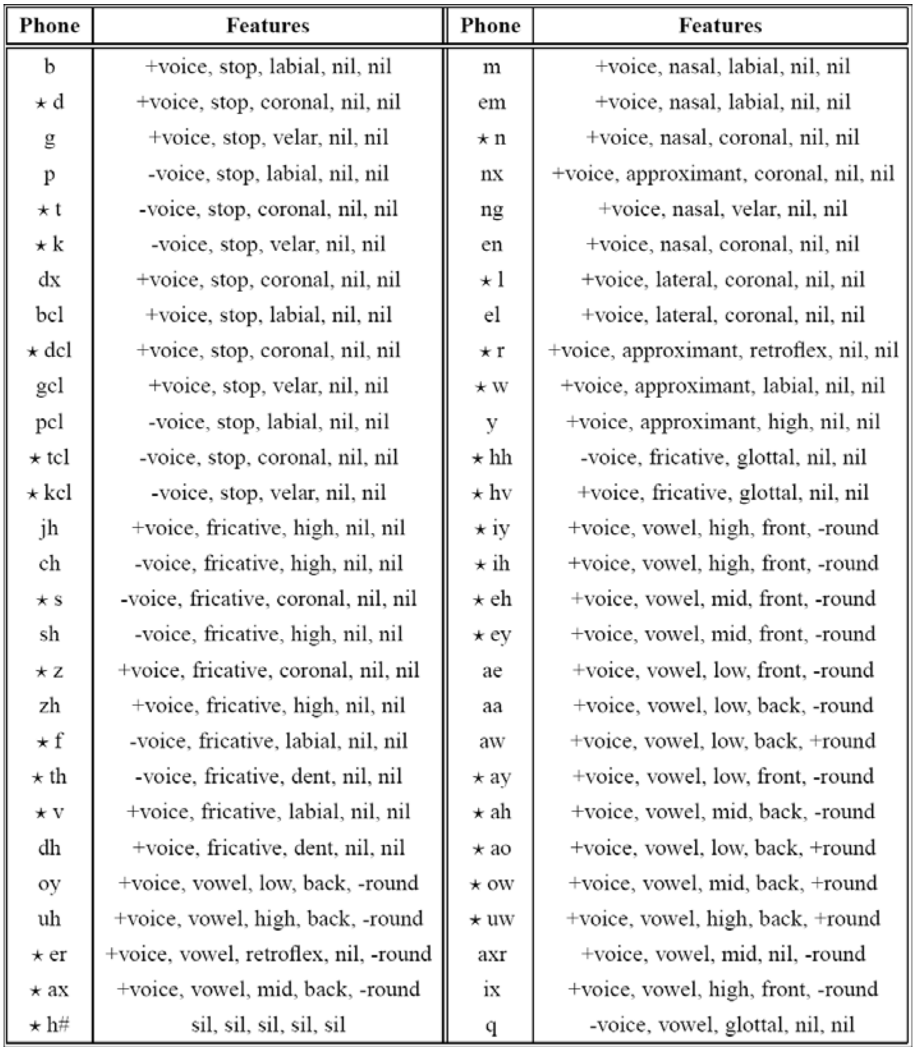
8.2 Future Work

A number of issues are still not entirely resolved within the scope of this thesis and require additional efforts in the future. Firstly, the performance of articulatory information based audio visual speech recognition is highly depended on the articulatory feature itself. If it is the case that we cannot fully discover the articulatory information, the whole idea can be questioned. That is, at the level of articulatory features, several things need to be improved. In this thesis, the articulatory features and their transcriptions are all converted from low level acoustic and visual data. The mapping between acoustic-visual signals and articulatory features is based on MLP classifiers. Although MLPs have shown a good performance among the system in this study, there may be other more powerful classifiers which can better retrieve the articulatory features, e.g. Supported Vector Machine (SVM).

Secondly, in order to realize the articulatory channel AVSR in Section 7.4, we designed a set of articulatory meta-feature transcriptions to describe three different articulatory information channels. Using these transcriptions enabled us to realize the graphical model based articulatory AVSR systems, but a better design of the phone to articulatory meta-feature mapping might further improve the performance of the system. In addition to the principles introduced in Section 7.4, two more aspects should be considered in order to design better articulatory meta-features. One aspect is that the number of values in a meta-feature should not be too large. Since MLPs are used for conversion from low level features to articulatory features, a large number of output values in one classifier could lead to inaccurate articulatory features. Another aspect is the number of channels in such systems. The number of channels was limited to three because of extreme computational requirements. Ideally, a structure with all seven articulatory information channels should be implemented in order to investigate relationships across all the channels.

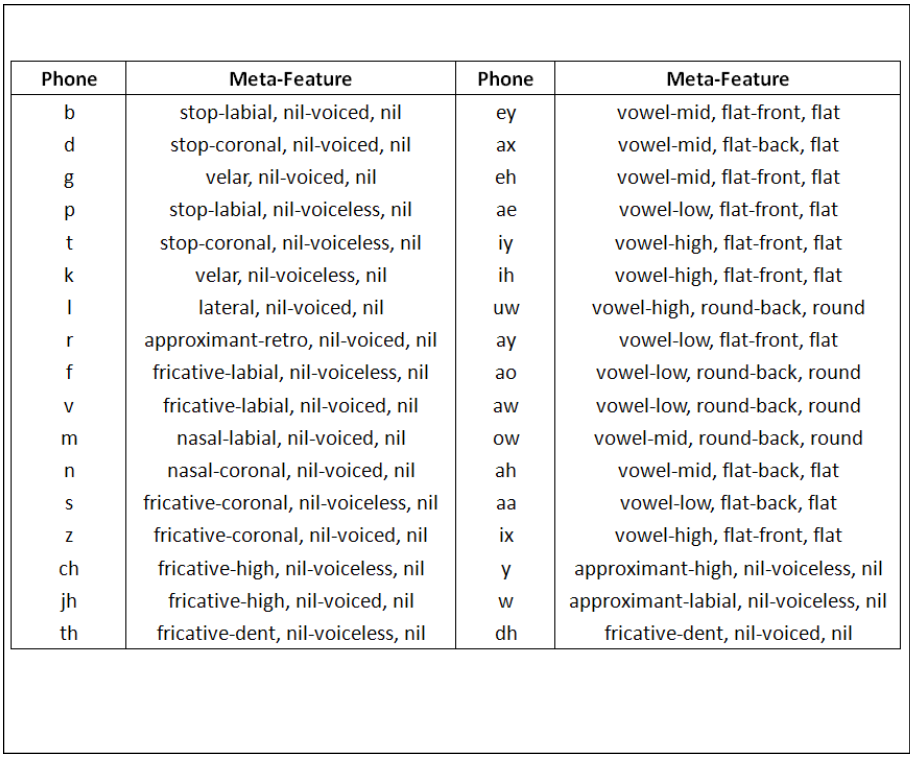
Thirdly, the definition of articulatory information synchronization is a good perspective to study the relationships between different acoustic and visual articulatory features. However, concrete asynchronous relationships between the acoustic and the visual channel have not been modelled yet. Additional dependencies between acoustic and visual variables might be necessary to create more flexible and context-dependency pronunciation models.

Appendix A



Phone	Features	Phone	Features
b	+voice, stop, labial, nil, nil	m	+voice, nasal, labial, nil, nil
⋆ d	+voice, stop, coronal, nil, nil	em	+voice, nasal, labial, nil, nil
g	+voice, stop, velar, nil, nil	⋆ n	+voice, nasal, coronal, nil, nil
p	-voice, stop, labial, nil, nil	nx	+voice, approximant, coronal, nil, nil
⋆ t	-voice, stop, coronal, nil, nil	ng	+voice, nasal, velar, nil, nil
⋆ k	-voice, stop, velar, nil, nil	en	+voice, nasal, coronal, nil, nil
dx	+voice, stop, coronal, nil, nil	⋆ l	+voice, lateral, coronal, nil, nil
bcl	+voice, stop, labial, nil, nil	el	+voice, lateral, coronal, nil, nil
⋆ dcl	+voice, stop, coronal, nil, nil	⋆ r	+voice, approximant, retroflex, nil, nil
gcl	+voice, stop, velar, nil, nil	⋆ w	+voice, approximant, labial, nil, nil
pcl	-voice, stop, labial, nil, nil	y	+voice, approximant, high, nil, nil
⋆ tcl	-voice, stop, coronal, nil, nil	⋆ hh	-voice, fricative, glottal, nil, nil
⋆ kcl	-voice, stop, velar, nil, nil	⋆ hv	+voice, fricative, glottal, nil, nil
jh	+voice, fricative, high, nil, nil	⋆ iy	+voice, vowel, high, front, -round
ch	-voice, fricative, high, nil, nil	⋆ ih	+voice, vowel, high, front, -round
⋆ s	-voice, fricative, coronal, nil, nil	⋆ eh	+voice, vowel, mid, front, -round
sh	-voice, fricative, high, nil, nil	⋆ ey	+voice, vowel, mid, front, -round
⋆ z	+voice, fricative, coronal, nil, nil	ae	+voice, vowel, low, front, -round
zh	+voice, fricative, high, nil, nil	aa	+voice, vowel, low, back, -round
⋆ f	-voice, fricative, labial, nil, nil	aw	+voice, vowel, low, back, +round
⋆ th	-voice, fricative, dent, nil, nil	⋆ ay	+voice, vowel, low, front, -round
⋆ v	+voice, fricative, labial, nil, nil	⋆ ah	+voice, vowel, mid, back, -round
dh	+voice, fricative, dent, nil, nil	⋆ ao	+voice, vowel, low, back, +round
oy	+voice, vowel, low, back, -round	⋆ ow	+voice, vowel, mid, back, +round
uh	+voice, vowel, high, back, -round	⋆ uw	+voice, vowel, high, back, +round
⋆ er	+voice, vowel, retroflex, nil, -round	axr	+voice, vowel, mid, nil, -round
⋆ ax	+voice, vowel, mid, back, -round	ix	+voice, vowel, high, front, -round
⋆ h#	sil, sil, sil, sil, sil	q	-voice, vowel, glottal, nil, nil

Figure A.1: Phone to articulatory transcription conversion table for our experiments.



Phone	Meta-Feature	Phone	Meta-Feature
b	stop-labial, nil-voiced, nil	ey	vowel-mid, flat-front, flat
d	stop-coronal, nil-voiced, nil	ax	vowel-mid, flat-back, flat
g	velar, nil-voiced, nil	eh	vowel-mid, flat-front, flat
p	stop-labial, nil-voiceless, nil	ae	vowel-low, flat-front, flat
t	stop-coronal, nil-voiceless, nil	iy	vowel-high, flat-front, flat
k	velar, nil-voiceless, nil	ih	vowel-high, flat-front, flat
l	lateral, nil-voiced, nil	uw	vowel-high, round-back, round
r	approximant-retro, nil-voiced, nil	ay	vowel-low, flat-front, flat
f	fricative-labial, nil-voiceless, nil	ao	vowel-low, round-back, round
v	fricative-labial, nil-voiced, nil	aw	vowel-low, round-back, round
m	nasal-labial, nil-voiced, nil	ow	vowel-mid, round-back, round
n	nasal-coronal, nil-voiced, nil	ah	vowel-mid, flat-back, flat
s	fricative-coronal, nil-voiceless, nil	aa	vowel-low, flat-back, flat
z	fricative-coronal, nil-voiced, nil	ix	vowel-high, flat-front, flat
ch	fricative-high, nil-voiceless, nil	y	approximant-high, nil-voiceless, nil
jh	fricative-high, nil-voiced, nil	w	approximant-labial, nil-voiceless, nil
th	fricative-dent, nil-voiceless, nil	dh	fricative-dent, nil-voiced, nil

Figure A.2: Phone to articulatory feature conversion table for our experiments.

References

Alais, D., Carlile, S., and Knudsen, E. (2005). Sync-Proceedings of The National Academy of Sciences of The United States of America. volume 102, pages 2244–2247. National Academy of Sciences.

Amer, T. and Berndsen, J. (2003). HARTFEX: A Multi-Dimensional System of HMM Based Recognizers for Articulatory Feature Extraction. In *8th European Conference on Speech Communication and Technology*, pages 2541–2544, Geneva, Switzerland.

Barker, J. and Cooke, M. (2007). Modelling Speaker Intelligibility in Noise. *Speech Communication*, 49:402–417.

Basu, S., Oliver, N., and Pentland, A. (1998). 3D Modeling and Tracking of Human Lip Motions. In *IEEE International Conference on Computer Vision*, pages 337–343.

Baum, L. and Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563.

Bilmes, J. (2010). Graphical Models in Speech and Language Research. `http://ssli.ee.washington.edu/~bilmes/gmtk/doc.pdf`.

Bilmes, J. and Zweig, G. (2002). The Graphical Models Toolkit: An Open Source Software System for Speech and Time-series Processing. In *Proceedings of Acoustic Speech and Signal Processing*, volume 4, pages 3916–3919.

Bourlard, H., Konig, Y., Morgan, N., and Ris, C. (1996). A New Training Algorithm for Hybrid HMM/ANN Speech Recognition Systems. In *Proceedings of European Association for Signal Processing(EUSIPCO'96)*, pages 101–104.

Browman, C. and Goldstein, L. (1992). Articulatory Phonology: An Overview. *Phonetica*, 19:155–180.

Browman, C. and Goldstein, L. (1993). Dynamics and Articulatory Phonology. *Status Reports on Speech Research, SR-l 13*, pages 51–62.

Chen, T. (2001). Audiovisual Speech Processing. *IEEE Signal Processing Magazine*, 18(1):9–21.

Chiou, G. and Hwang, J. (1997). Lipreading from Color Video. *IEEE Transactions on Image Processing*, pages 1192–1195.

Cootes, T., Edwards, G., and Taylor, C. (2001). Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:681–685.

Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1995). Active Shape Models—Their Training and Application. *Comput. Vis. Image Underst.*, 61(1):38–59.

Cybenko, G. (1989). Approximation by Superpositions of A Sigmoidal Function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition. In *IEEE Transactions on Audio, Speech, and Language Processing.*

Davis, S. and Mermelstein, P. (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *Readings in speech recognition*, pages 65–74.

Deng, L. and Sun, D. (1993). Speech Recognition Using Atomic Speech Units Constructed From Overlapping Articulatory Features. In *Proceedings of European Conference on Speech Communication and Technology(EUROSPEECH 1993)*, pages 1635–1638, Berlin, Germany.

Dugast, C., Devillers, L., and Aubert, X. (1994). Combining TDNN and HMM in a Hybrid System for Improved Continuous Speech Recognition. *IEEE Transactions Speech and Audio Processing*, 2:217–223.

Dupont, S. and Luettin, J. (2000). Audio-Visual Speech Modelling for Continuous Speech Recognition. *IEEE Transactions on Multimedia*, 2:141–151.

Erler, K. and Freeman, G. (1996). An HMM-Based Speech Recognizer Using Overlapping Articulatory Features. *The Journal of the Acoustical Society of America*, 100(4):2500–2513.

Fisher, W., Doddington, G., and Goudie-Marshall, K. (1986). The DARPA Speech Recognition Research Database: Specifications and Status. In *Proceedings of DARPA Workshop on Speech Recognition*, pages 93–99.

Frankel, J. and King, S. (2001). ASR: Articulatory Speech Recognition. In *7th European Conference on Speech Communication and Technology*, pages 599–602. International Speech Communication Association.

Freund, Y. and Schapire, R. (1997). A Decision-Theoretic Generalization of On-Line Learning and An Application to Boosting. *Journal of Computer and System Sciences*, 55:119–139.

Frydenberg, M. (1990). The Chain Graph Markov Property. *Scandinavian Journal of Statistics*, 17:333–353.

Galatas, G., Potamianos, G., Papangelis, A., and Makedon, F. (2011). Audio Visual Speech Recognition in Noisy Visual Environments. In *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 19:1–19:4.

Gasteratos, A., Vincze, M., and Tsotsos, J., editors (2008). *Computer Vision Systems, 6th International Conference, ICVS 2008, Santorini, Greece, May 12-15, 2008, Proceedings*, volume 5008 of *Lecture Notes in Computer Science*. Springer.

Ghosh, P. and Narayanan, S. (2011). Automatic Speech Recognition Using Articulatory Features from Subject-Independent Acoustic-to-Articulatory Inversion. *Journal of the Acoustical Society of America*, 130:EL251–EL257.

Gowdy, J., Subramanya, A., Bartels, C., and Bilmes, J. (2004). DBN based Multi-Stream Models for Audio-Visual Speech Recognition. In *Proceedings of International Conference on Acoustic, Speech, and Signal Processing(ICASSP 2004)*, pages 993–996. IEEE, Canada.

Graf, H., E.C., and Potamianos, M. (1997). Robust Recognition of Faces and Facial Features with a Multi-Modal System. In *Proceedings of Inter-*

national Conference on Systems, Man and Cybernetics, pages 2034–2039. IEEE Computer Society.

Green, G. (1976). Temporal Aspects of Audition. Technical report, Oxford University.

Hermansky, H. (1990). Perceptual Linear Predictive (PLP) Analysis of Speech. *Journal of Acoustic Soceity*, 87:1738–1752.

Hermansky, H., Daniel, P., and Sharma, S. (2000). Tandem Connectionist Feature Extraction for Conventional HMM Systems. In *Proceedings of International Conference on Acoustic, Speech, and Signal Processing(ICASSP 2000)*, pages 1635–1638.

Hermansky, H. and Morgan, N. (1994). RASTA Processing of Speech. 2(4):578–589.

Hermansky, H., Morgan, N., Bayya, A., and Kohn, P. (1991). RASTA-PLP Speech Analysis.

Hill, A., Taylor, C., and Cootes, T. (1992). Object Recognition by Flexible Template Matching using Genetic Algorithms. In *Proceedings of the Second European Conference on Computer Vision*, pages 852–856, London, UK. Springer-Verlag.

Huang, W. and Lippmann, R. (1990). HMM Speech Recognition with Neural Net Discrimination. *Advances in neural information processing systems 2*, pages 194–202.

International Phonetic Association (1999). *Handbook of The International Phonetic Association: A Guide to The Use of The International Phonetic Alphabet*. Cambridge University Press.

Jacobs, R., Jordan, M., Nowlan, S., and Hinton, G. (1991). Adaptive Mixture of Local Experts. *Neural Computation*, 3:79–87.

Jain, L. and Medsker, L. (1999). *Recurrent Neural Networks: Design and Applications*. CRC Press, Inc., Boca Raton, FL, USA.

Kanokphara, M. and Berndsen, J. (2005). Better HMM-Based Articulatory Feature Extraction with Context Dependent Model. In *Proceedings of the*

18th International Florida Artificial Intelligence Research Society Conference, pages 370–374.

King, S., Stephenson, T., Isard, S., Taylor, P., and Strachan, A. (1998). Speech Recognition via Phonetically Featured Syllables. In *Proceedings of International Conference on Spoken Language Processing*, pages 1031–1034.

Kirchhoff, K. (1998). Combining Articulatory and Acoustic Information for Speech Recognition in Noisy and Reverberant Environments. In *Proceedings of International Conference on Spoken Language Processing(ICSLP 1998)*, pages 891–894.

Kirchhoff, K. (1999). *Robust Speech Recognition Using Articulatory Information*. PhD thesis, University of Bielefeld, Germany.

Kirchhoff, K., Fink, G., and Sagerer, G. (2000). Conversational Speech Recognition Using Acoustic and Articulatory Input. In *Proceedings of International Conference on Acoustic, Speech, and Signal Processing(ICASSP 2000)*, pages 1435–1438.

Kohonen, T., Schroeder, M., and Huang, T., editors (2001). *Self-Organizing Maps*. Springer-Verlag New York, Inc., New Jersey, USA.

Luettin, J., Potamianos, G., and Neti, C. (2001). Asynchronous Stream Modeling for Large Vocabulary Audio-Visual Speech Recognition. In *Proceedings of International Conference on Speech, and Signal Processing(ICASSP 2001)*, pages 169–172.

Luettin, J., Thacker, N., and Beet, S. (1996). Speechreading Using Shape and Intensity Information. In *Proceedings of International Conference on Spoken Language Processing*, pages 58–61.

Matthews, I., Cootes, F., Bangham, J., Cox, S., and Harvey, R. (2002). Extraction of Visual Features for Lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:2002.

McGurk, H. and MacDonald, J. (1976). Hearing Lips and Seeing Voices. *Nature*, pages 746–748.

Murphy, K. (2001). A Brief Introduction to Graphical Models and Bayesian Networks. `http://www.cs.ubc.ca/~murphyk/Bayes/bayes.html`.

Nefian, A. V., Liang, L., Pi, X., Liu, X., and Murphy, K. (2002). Dynamic Bayesian Networks for Audio-Visual Speech Recognition. *EURASIP J. Appl. Signal Process.*, 2002(1):1274–1288.

Osuna, E., Freund, R., and Girosi, F. (1997). Training Support Vector Machines: An Application to Face Detection. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pages 130–136.

Papcun, G., Hochberg, J., Thomas, T., Laroche, F., Zacks, J., and Levy, S. (1992). Inferring Articulation and Recognizing Gestures from Acoustics with a Neural Network Trained on X-ray Microbeam Data. *The Journal of the Acoustical Society of America*, 92(2):688–700.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Petajan, E. (1984). *Automatic Lipreading to Enhance Speech Recognition (Speech Reading).* PhD thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA.

Poggio, T. and Sung, K. (1995). Finding Human Faces with a Gaussian Mixture Distribution-based Face Model. In *In Asian Conf. on Computer Vision*, pages 437–446.

Potamianos, G., Graf, H., and Cosatto, E. (1998). An Image Transform Approach for HMM Based Automatic Lipreading. In *Proc. Int. Conf. Image Processing*, pages 173–177.

Potamianos, G. and Neti, C. (2001). Improved ROI And Within Frame Discriminant Features for Lipreading. In *Proceedings of International Conference on Image Processing, Thessaloniki, Greece*, pages 250–253.

Potamianos, G., Neti, C., Luettin, J., and Matthews, I. (2004). Audio-visual Automatic Speech Recognition: An Overview. In *Issues in Visual and Audio-visual Speech Processing.* MIT Press.

PVCrp.com (2010). Speech and Voice Production. Website. `http://pvcrp.com/speech___voice_production.php/`.

Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286.

Ravyse, I., Jiang, D., Jiang, X., Lv, G., Hou, Y., Sahli, H., and Zhao, R. (2006). *DBN Based Models for Audio-Visual Speech Analysis and Recognition*, volume 4261, pages 19–30.

Rowley, H., Baluja, S., and Kanade, T. (1996). Neural Network-based Face Detection. *IEEE Transactions On Pattern Analysis and Machine intelligence*, 20:23–38.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Internal Representations by Error Propagation. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*, pages 318–362.

Saenko, K., Darrell, T., and Glass, J. (2004). Articulatory Features for Robust Visual Speech Recognition. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 152–158, New York, NY, USA. ACM.

Sanderson, C. (2009). *Biometric Person Recognition: Face, Speech and Fusion*. VDM-Verlag.

Senior, A. (1999). Recognizing Faces in Broadcast Video.

Serences, J., Ester, E., Vogel, E., and Awh, E. (2009). Stimulus-Specific Delay Activity in Human Primary Visual Cortex. *Psychological Science*, 20:207–214.

Stevens, J. (1957). A Comparison of Ratio Scales for The Loudness of White Noise And The Brightness of White Light.

Stevens, S. and Volkmann, J. (1940). The Relation of Pitch to Frequency: A Revised Scale. *The American Journal of Psychology*, 53(4):329–353.

Thomas, J. Z. . T. (1994). A New Neural Network for Articulatory Speech Recognition and Its Application to Vowel Identification. *Computer, Speech and Language*, 8:189–209.

Viola, P. and Jones, M. (2001). Robust Real-time Object Detection. *International Journal of Computer Vision*, 57:137–154.

Yang, M., Kriegman, D., Member, S., and Ahuja, N. (2002). Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:34–58.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2006). *The HTK Book, Version 3.4*. Cambridge University Engineering Department, Cambridge, UK.

Young, S., Russell, N., and Thornton, J. (1989). Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems. Technical report, Engineering Department,Cambridge University.

Zweig, G. and Russell, S. (1998). Speech Recognition with Dynamic Bayesian Networks. In *Proceedings of National Conference on Artificial Intelligence(AAAI-98)*, pages 173–180.

Zwicker, E. (1961). Subdivision of The Audible Frequency Range into Critical Bands. *Journal of Acoustical Society of America*, 33:248.