Université d'Ottawa · University of Ottawa

Voice Stream Based Lip Animation For Audio-Video Communication.

By

Michel D. Bondy ©

May 2001

Submitted to the

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Information Technology and Engineering (SITE)

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering.

Acknowledgements

I want to use this space to thank the people who have helped me achieve success in my studies as well as in other aspects of my life. I thank my supervisor Dr.Emil Petriu for his support and his patience. I thank my grandfather "Pepé" Henri Deschamps because without him I would never have considered continuing my education. I also thank my parents Denis and Viviane. Without their love and support I could never have become the successful person, athlete, and student I am. I cannot forget my loving sister. Her rivalry never lets me slow down. Thanks also to Gaby for letting me measure the shape of her lips.

Abstract

Voice Stream-Based Lip Animation for Audio-Video Communication

This thesis describes a system that uses the voice track to determine the shape of a speaker's lips for use in a model-based audio-video communication system. A parametric model (deformable template) is used to measure the shape of a speaker's lips. The system uses Linear Predictive Coding (LPC) analysis and the LPC cepstral coefficients for audio recognition. As each individual has his/her own typical lip positions while uttering various speech sounds the system is speaker dependent. A vector quantization algorithm is used to create a compact and speaker dependent cepstral coefficient to lip shape parameter mapping. This mapping is used along with the audio analysis to determine the shape of a speaker's lips from the voice stream.

Animation of a parametric lip contour from the voice track provides a convenient solution to the non-trivial problem of voice-image synchronization in audio-video communication. As strong correlations can be found between the lip contour shape and the properties of the voice track it becomes possible to use the voice to directly drive the lip contour shape in the model-based video rendering/animation process. This allows the further reduction of the bit rate for audio-video storage or transmission. Parametric animation of the lip contour from the voice track signal can find obvious application to the animation of talking avatars (virtual humans). The visual information added by animating the lip contours can also increase the intelligibility of audio messages for persons with impaired hearing.

Table Of Contents

List of Figures

List of tables

Definition of terms

LPC	Linear Predictive Coding
RGB	Red Green Blue color model
HSV	Hue Saturation Value color model
CRT	Cathode Ray Tube
WIN32 API	Windows 32 bit Application Programming Interface
AVI	Audio Video Interleaved
DSP	Digital Signal Processing
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
IFFT	Inverse Fourier Transform
Log	Logarithm function base 10
Abs	Magnitude of a complex number
Re	Real part of a complex number
VQ	Vector Quantization

CHAPTER 1 – Introduction

1.1 Introduction

This thesis describes a system that uses the voice track to determine the shape of a speaker's lips for use in a model-based audio-video communication system. A parametric model (deformable template) is used to measure the shape of a speaker's lips. The system uses Linear Predictive Coding (LPC) analysis and the LPC cepstral coefficients for audio recognition. As each individual has his/her own typical lip positions while uttering various speech sounds the system is speaker dependent. A vector quantization algorithm is used to create a compact and speaker dependent cepstral coefficient to lip shape parameter mapping. This mapping is used along with the audio analysis to determine the shape of a speaker's lips from the voice stream.

Animation of a parametric lip contour from the voice track provides a convenient solution to the non-trivial problem of voice-image synchronization in audio-video communication. As strong correlations can be found between the lip contour shape and the properties of the voice track it becomes possible to use the voice to directly drive the lip contour shape in the model-based video rendering/animation process. This allows the further reduction of the bit rate for audio-video storage or transmission. Parametric animation of the lip contour from the voice track signal can find obvious application to the animation of talking avatars (virtual humans). The visual information added by animating the lip contours can also increase the intelligibility of audio messages for persons with impaired hearing.

1.1.1 Project context

Driving the lips from the audio signal fits into the model-based audio/video communication system described in [Assaf97]. The system envisioned in [Assaf97] consists of a muscle based animated face model for image synthesis and a parametrically animated vocal tract model for voice synthesis. The goal of such a system is to reduce the data rate needed for real-time audio/video communication. A

structural diagram of this system is contained in Figure 1. The synthetic face on the receiver end is animated using a muscle-based model with 44 basic muscle movements. The muscle model is fitted in 3D to each individual speaker's face and

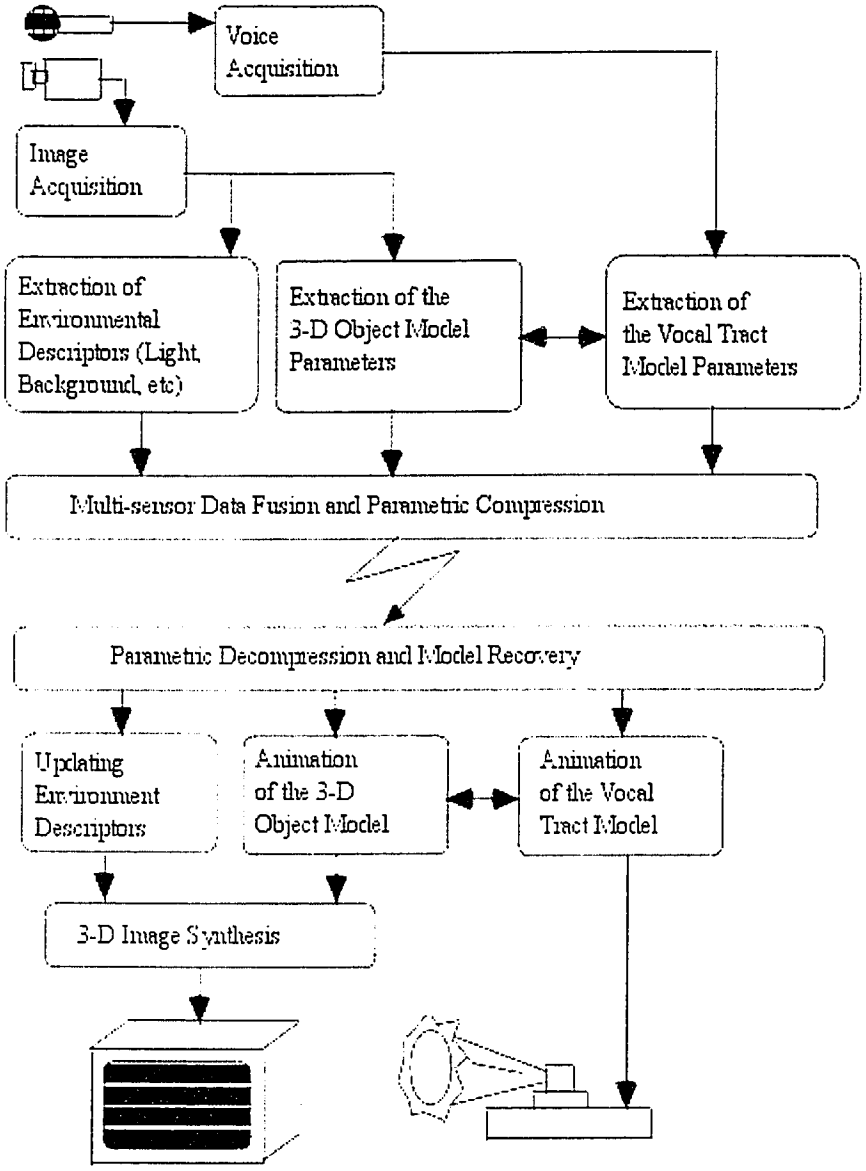

Figure 1 Structural diagram of the video-telephony system of [Assaf97].

exists on both the emitter and receiver ends. The live images of the speaker's face are analyzed and the optimal parameters of the facial model are extracted. Because of the existence of the model on the receiver end, only parameters describing the state of the facial model need to be transmitted in order to animate the model. Once the parameters are received at the receiver end, the model is animated with movements

that closely follow the movements of the live subject. This coding scheme necessitates a low bit rate to transmit or store the images of a speaker's talking head.

Model based coding/decoding is also used to compress the data needed to transmit the speaker's voice. The voice is coded and decoded using mathematical models that parameterize the state of the vocal source and the vocal tract. On the sender end, the optimal parameters of the voice model are determined from the live voice signal. These parameters are transmitted to the receiver end where the voice is reconstructed allowing for low bit rate transmission or storage of the speaker's voice signal.

The authors of [Assaf97] refer to a "read my lips" synchronization method where the temporal correlations between the video model parameters and the voice model parameters are used to correct the temporal skew that can crop up when transmitting digital information over a network.

Instead of exploiting the temporal correlation that exists between the audio and video parameters to correct the temporal skew, it is the goal of this thesis to show that the parameters of the coded voice can be used directly to drive the shape of the lips. A unified approach to coding both the audio and video portions of the communication system using only one set of parameters achieves the desired audio/video synchronization because both the audio and video streams are synthesized from the same data. Because the lip shape parameters are extracted from the voice model parameters the lip shape parameters extracted from the live images of the speaker have become redundant and therefore need not be transmitted further reducing the required amount of transmitted data.

1.1.2 Entire audio/video communication system

A structural diagram of a new model based audio/video communication system that uses the voice model parameters to drive the shape of the lips is given in Figure 2.

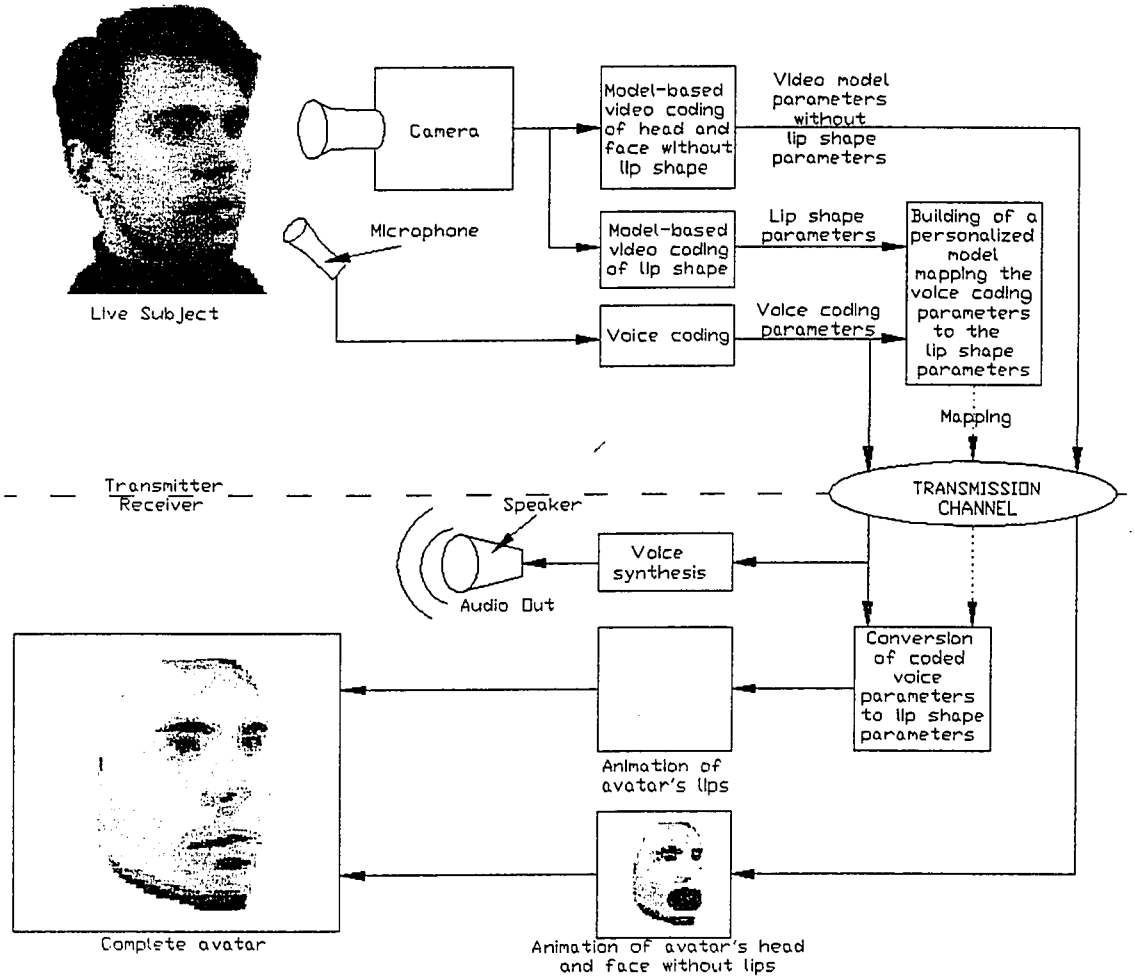

Figure 2 Entire model-based audio/video communication system

Figure 2 illustrates the global structure of a system that is an interconnection of many non-trivial subprojects. The input of the entire model-based audio/video communication system is taken using the camera and microphone focused on the live subject. From the input audio signal the optimal parameters of the voice model are calculated and from the input image stream the optimal parameters for the head model are extracted. These values are transmitted to the receiving end where the coded voice parameters are used to reconstruct the voice signal. The coded voice parameters are also used to drive the lip shape model. The video model parameters are used to animate the features of the synthetic head model (but not the lips). The animated lip model as well as the animated head model is combined to form the complete avatar.

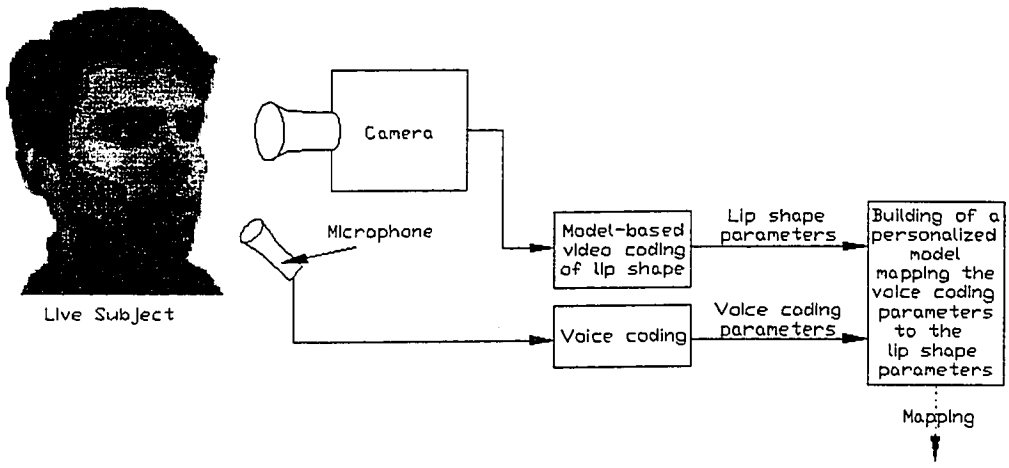

Figure 3 Off-line training stage

1.1.2.1 Training stage

To be able to estimate lip shape from the audio stream a speaker dependent mapping linking the voice model parameters to the lip shape parameters is required. In order to create the mapping an off-line training stage is introduced. Figure 3 highlights the elements of the system that are in use during the training stage. The first step in the training stage is to record a reference audio/video file. This video is recorded under laboratory conditions (e.g. low background noise, special blue makeup marks the lips see section 5.3). The voice is analyzed to extract the optimal voice model parameters for each audio frame and the video is analyzed to extract the optimal lip model parameters for each video frame.

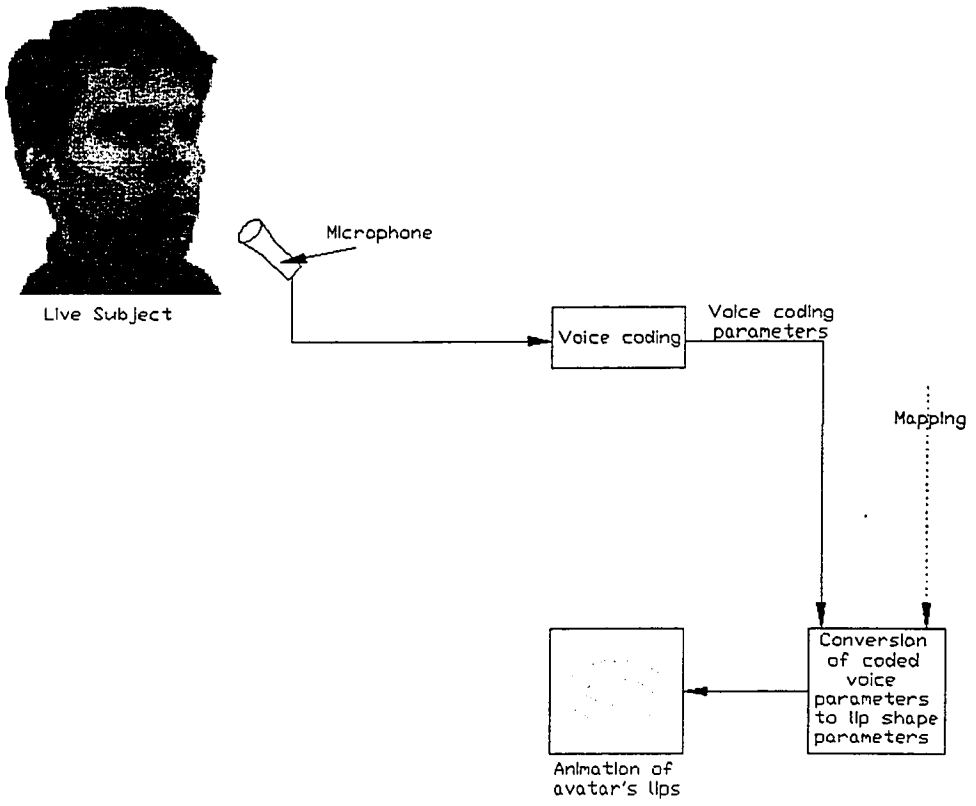

Figure 4 Animation stage

The parameters of the time aligned audio and video frames are compressed using a vector quantization algorithm and associated one to the other using a weighted average algorithm.

1.1.2.2 Animation stage

Once the mapping is complete the lip model can be animated from the coded voice parameters. The system therefore moves into the lip model animation stage – highlighted by the bold components of Figure 4. To animate the lip model only the parameters of the coded voice need be calculated. The coded voice parameters are used along with the mapping to choose the best set of lip model parameters for each

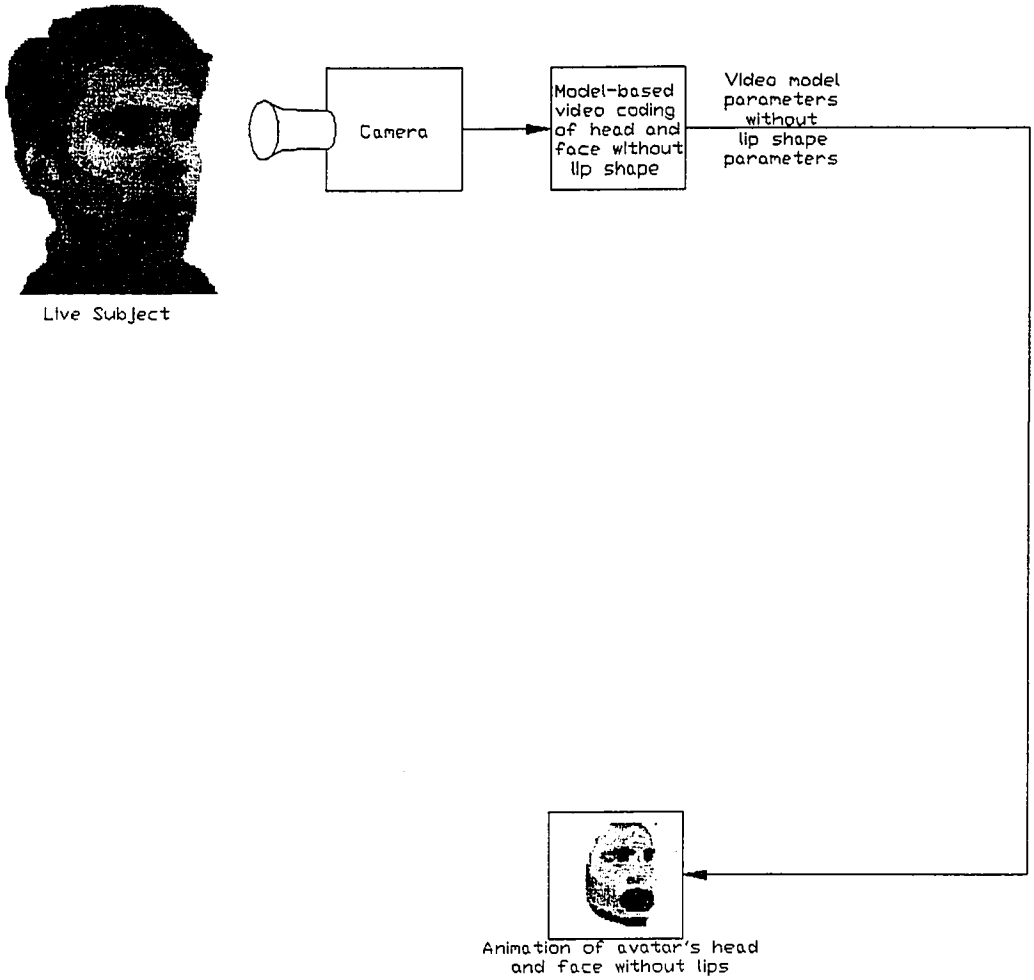

Figure 5 Model-based video coding

set of coded voice parameters. The chosen parameters are used to animate the lip shape model.

1.1.2.3 Model based video coding and decoding

The next part of the audio/video communication system, highlighted in Figure 5, to consider is the model-based video coding of the head features. [Codea99] has implemented the real-time head *pose* (*position, orientation and scale estimation*) extraction. Further discussion of the model based video coding is beyond the scope of this thesis. However, summaries of several recent articles related to model based video coding can be found in the *state of the art* section.

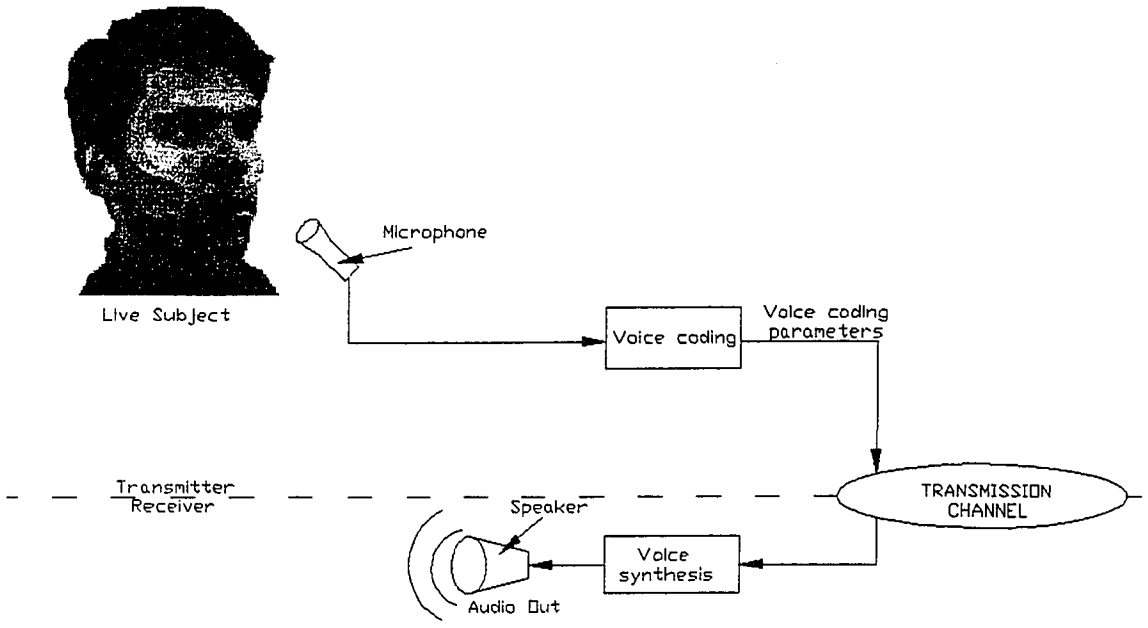

Figure 6 Voice coding-decoding

1.1.2.4 Low bit rate voice coding and decoding

Figure 6 highlights the elements of the global communication system that are used to perform the voice coding and decoding. Several methods of low bit rate voice coding and decoding exist. One example of a low bit rate voice coder and decoder that has reached commercial maturity is the MELP (Mixed-Excitation Linear Predictive) algorithm. The details and characteristics of one company's implementation of the MELP algorithm is given in annex 1. The parameters created by the audio analysis of the training stage and the animation stage are compatible for use in a variation on the MELP algorithm or other linear predictive voice-coding algorithm. However the conversion between the voice analysis algorithm presented in this thesis to the MELP or other linear predictive voice-coding algorithm is beyond the scope of this thesis.

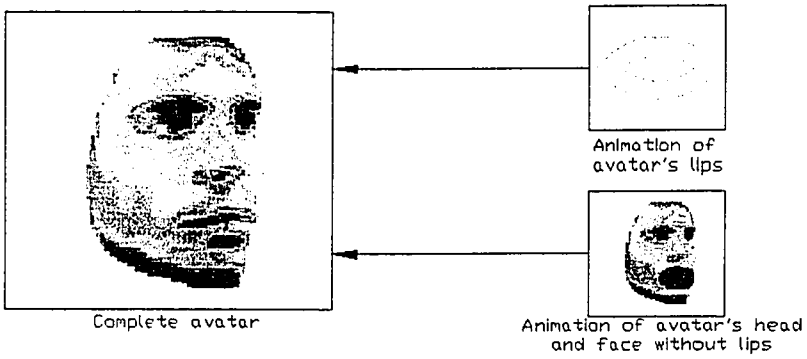

Figure 7 Integration of lip model with face model

1.1.2.5 Integration of animated lip model with head model

Once the whole model based video coding decoding system is complete it is only a matter of creating a mechanism to insert the lip shape parameters calculated using the voice parameters into the rest of the animated head model in order to obtain the complete avatar (see Figure 7). The integration of the lip model with the rest of the avatar's head is a component beyond the scope of this thesis and remains a direction for future work.

1.1.3 Focus on lip animation from the voice stream

The scope of this thesis is limited to the sections of the model based audio/video communication system described in Figure 2 that are related to the generation of the lip model parameters from the audio signal. Therefore only the concepts and algorithms related to the training stage illustrated in Figure 3 and the animation stage illustrated in Figure 4 are presented within this document.

1.2 State of the art

Generating the lip shapes from the voice signal is a multi-disciplinary task requiring expertise in many distinct fields of study. From audio analysis to video modeling techniques, knowledge in these subjects is just the beginning. This section provides a short survey of the current literature on a variety of subjects needed to complete this project. The subject headings are:

- Audio-video synthesis from text
- Audio-video synthesis from speech
- Other audio-visual speech synthesis systems
- Audio speech synthesis
- Face detection
- Lip sound synchronization
- Audio-video recognition
- Deformable templates
- Rounding anticipation
- Energy Minimization (Optimization)
- Lip-Vocal tract relations
- 3D lip/face model extraction
- Color models
- Lip-reading
- Speech coding, Pitch/Formant estimation LPC analysis
- Image coding
- Vector quantization

1.2.1 Audio-video synthesis from text

[Breen96] developed a talking head animated by their text-to-speech system. This paper describes a system that generates visual speech in real time. [Benoit98] created a text-to-visual speech synthesizer that animates a synthetic head and synthesizes a synthetic voice from text. The authors describe different structures used to create their text-to-visual speech synthesizers. [Ezzat00] presents a text-to-audiovisual speech synthesizer that uses visemes. The visemes are arranged according to the phonemes contained within the text and morphing between visemes is used to give a realistic talking face. [Ip96] created a facial model using non-uniform rational b-splines. This

facial model is animated from a text to audio-visual speech system. [Masuko98] produced visual speech from text using hidden Markov models. Coarticulation is implicitly incorporated into the mouth shapes. [Olives99] uses a Finnish commercial auditory text to speech synthesizer and generates facial animation synchronous the synthesized speech. [Ng93] animated a lip model using a text-to-phoneme system.

1.2.2 Audio-video synthesis from speech

[Faruquie00] presents a translingual audio driven facial animation system. The recognition is done in English and synthesis in another. [Karunaratne99] describes a visual speech synthesis system that uses phonemes extracted from the speech signal and converts them into visemes. The visemes are then combined with emotional cues. [Lagana96] discusses a real time audio-video synthesis system that uses a Kohonen neural network to relate audio input to articulatory estimates. [Lepsoy98] describes a speech driven facial animation system that maps the articulatory parameters used in phonetics into ASM coefficients that are used to illustrate lip shape. [Lewis91] created a speech driven mouth animation system that uses linear prediction to recognize phonemes. The phonemes are associated with mouth positions for key frames. [Luo94] created a speech driven mouth animation system that classifies sounds into one of 9 visemes using recurrent neural networks. [McAllister98] use the absolute value of the Fourier transform of a speech segment to estimate the shape of the speaker's mouth. This method could lead to a system that is speaker independent. [Morishima99] describes a real time multi-user communication system with voice driven synthetic face animation system with facial expressions. [Yang00] used a set of images to synthesize visual speech. The images are selected automatically. They describe an audio/video translation agent allowing communication between people who speak different languages. [Yamamoto98] used Hidden markov models to generate lip movements from the speech signal.

1.2.3 Other audio-visual speech synthesis systems

[Bregler97] developed a system called Video Rewrite. This system can change original movie footage to make a character say something he/she did not in the original movie. The order of the sounds and the corresponding lip shapes are rearranged to allow the movie character to say anything the user of such a system would want. [Brooke94] use hidden Markov models and principal component analysis to arrange images of a real face to create audio-video speech. [Cohen93] describes how the visual component of audio-video speech can offer much information. Lofqvist's (1990) coarticulation model is implemented. [Lavagetto96] created an audio-video communication system. The talker's face is modeled with a wire-frame. The audio-video parameters are used at the destination to synthesize a talking head. [Pelachaud96] created a system for the 3D animation of facial expression and head movements and speech synthesis using their own programming language. They are also working toward incorporating emotions into the generated facial expressions. [Sahandi98] provides a brief study of available facial animation and speech synthesis techniques, and a discussion of synchronization issues between the audio and video streams. [Storey88] explains how the practice of lip reading has stimulated research into developing visible speech to help in the comprehension of the speech signal and describes a simple method of audio-visual speech synthesis. [Williams00] created a visual speech synthesis system to assist persons with impaired hearing. Their system uses a correlation HMM to integrate HMMs from the audio and video to enhance synchronization of visual speech synthesis. [Hara00] created a robotic head with lips that can articulate to mimic the shape of the Japanese vowels. [Badin96] produced an anthropormorphic speech robot. This speech robot synthesizes speech by controlling the actuators that control the jaw, tongue lips and larynx.

1.2.4 Audio speech synthesis

[Cranen96] aims to improve speech synthesis by creating a parametric glottal model. The parameters of the model are chosen to copy as closely as possible the actual

human glottal geometry. [Rabiner95] discuses the influence modern voice processing advances has had on the latest products and services.

1.2.5 Face detection

[Dai96] presents an algorithm that can isolate faces in images within complex backgrounds. [Zhang97b] introduces a head tracking system to localize the head in a sequence of videophone images.

1.2.6 Lip sound synchronization

[Chen96] shows that using speech analysis and image processing together can solve audio-video synchronization problems.

1.2.7 Audio-video recognition

[Brooke96a] use principal component analysis and hidden Markov models for image compression. These features vectors also enable audio-video recognition of speech. [Brooke96b] state that visual information can help the understanding of a noisy audio signal and show how facial information can increase the recognition rate for automatic systems. Brooke96c] use the radial function representation of the lip contours as recognition features for lip shape. The results are relevant for video speech recognition and visual speech synthesis. [Cosi96] recognize phonemes using audio and video input and a feed forward recurrent back-propagation neural network. [Hennecke96] created a speech reading system using both audio and video to perform recognition. With an improved deformable template algorithm for tracking the lips. [Jourlin98] discusses the asynchronicity that can occur between the audio and video sources when performing audio-video speech recognition. He also discusses how audio video speech recognition can be applied towards speaker identification. [Petajan00] explains how the MPEG-4 Face Animation standard encourages research into visual speech recognition by facilitating the modeling of the audio and video streams. [Rogozan98] describes a

method of dynamically adapting the weight values given to the audio and video sources of information for use in a visual speech recognition system. [Tomlinson96] designed a bimodal speech recognition system that uses hidden Markov models to account for temporal variation between the audio and video stream. [Yu99] created an audio-video voice recognition system that models lip movements by considering each pixel's intensity as a function of time. Wavelet and Fourier transforms are applied to the intensity functions.

1.2.8 Deformable templates

[Coughlan00] detects an open hand in a cluttered grayscale image using a deformable template. [Escolano97] uses a circular deformable template to initialize an elliptical deformable template that tracts an inflating balloon inside an ultrasound image of a coronary artery. [Esme96] presents a deformable template optimized using a genetic algorithm and used to capture eye and lip contours. [Figueiredo97] uses Fourier and B-spline contour descriptors to create a deformable template with an adaptive degree of smoothness. [Hennecke94] describe the use of deformable templates in order to tract outer and inner lip contours. [Jain98] contains a review of research in deformable templates. The different types of templates are categorized. The details of one template are presented whose energy function uses edges, texture, color and region information and model is deformed using wavelets, splines or Fourier descriptors. [Jain96] presents a system that finds objects within an image by matching the objects in the image and a library of deformable templates. Their method works even when the object within the image is rotated or scaled. [Jyh-Yuan97] proposes a system for extracting features of the eye using deformable templates. The energy minimization process is replaced with a region-based approach. The method also incorporates a Canny edge operator. [Kober94] compares a model based approach (deformable template) and a knowledge based approach to detecting and representing the mouth in real video images. [Liu99] locates the head within an input sequence of images. Then deformable templates are used to track facial feature (eyes mouth nostrils). [Mirhosseini98] presents an algorithm to automatically extract mouth contours from face

images. The optimization process adapts the deformable template model hierarchically. [Ngan96] proposes the use of deformable templates adapt a wire frame mesh to facial images. [Rabi97] uses a deformable template consisting of two degree polynomials is to model the mouth shape of a speaker. [Rao94] describes an algorithm for modeling the shape of the mouth. The minimization of the model is done in closed form. [Rao95] proposes a 2D model similar to the 2D hidden Markov model structure. The system is tested my analyzing facial images. [Saji97] proposes deformable templates for extracting the position of facial features through a sequence of facial images. [Sakalli98] created a system of very low bit rate coding of human faces using deformable templates, wavelet decomposition and residual vector quantization. [Sungyun98] developed an efficient algorithm for face tracking within color image sequences using deformable templates. [Tawfik99] describes a scheme for object localization and classification of based on shape. [Wang00] presents a system for extracting eye and mouth features using deformable templates. The energy function is optimized using a genetic algorithm. [Yuille92] explains a method of extracting eye features from facial images using deformable templates. An energy function linking topology of the image intensity to the shape of the template is defined. The parameters minimizing the energy function are used to describe the features of the eye. [Yuille00] discusses the optimization of deformable parameters in terms of Bayesian probability theory. The paper discusses 3 algorithms, Dijkstra, Dynamic Programming and Twenty Questions. [Xie94] presents an improvement on the method contained in [Yuille92]. The energy function for the eye is modified. The order terms are normalized between 0 and 1 so the user needs no expert knowledge. All parameters are optimized simultaneously reducing processing time. [Xu98] generalizes the gradient vector flow (GVF), an external force for deformable templates, to include 2 spatially varying weighting functions. [Zhang97a] describes a deformable template for mouth feature estimation that uses the corner points of the lips and models the lip outline. A method of determining whether the mouth is open or closed is presented.

1.2.9 Rounding anticipation

[Cathiard92] studied rounding anticipation in French and Greek speakers. [Jourlin96] includes anticipation and retention phenomena in a audio-visual recognition system.

1.2.10 Energy Minimization (Optimization)

[Azencott97] used energy minimization by simulated annealing for the purpose of matching segmented parts of objects. [Fabian97] tested a simulated annealing algorithm by finding the global minimum of simple functions defined on a subset of k-dimensional Euclidean space. [Tsallis96] used a stochastic algorithm of simulated annealing to find the global minimum of a function. This algorithm is faster than the "Boltzman machine" and fast "Cauchy machine" simulated annealings. [Bresinski99] Generalizes the gradient descent method to a multiparameter descent method.

1.2.11 Lip-Vocal tract relations

[Badin94] studied the influence of the lips and vocal tract have on the acoustic properties of the voice signal. [Hani98] relates the shape of the vocal tact to facial behavior and have shown through the simultaneous measurements of the vocal tract shape, the facial shape and lip shape, that lip shape is related to speech acoustics. [Lindblom71] studied the relations between lip tongue jaw and larynx and the acoustic signal.

1.2.12 3D lip/face model extraction

[Basu98] uses a statistical and physical model of the lips to extract their 3D shape from 2D images. [Benoit96] performed some experiments testing the intelligibility of noisy voice combined with real images of face, real images of lips only, and synthetic lips only. [Guiard-Marigny96] have defined a 3D lip model in the context of creating an

audio-video speech synthesizer. [Deng00] designed a system of measuring the 3D and 4D (3D plus time) shape of soft-tissue body parts using ultrasound. [Ip96] created a NURBS facial model to be used to generate facial gestures synchronously with a text to speech synthesizer. [Kapfer97] uses color and motion information and a shape model to detect the human face in a sequence of images. [Kervrann97] created a statistical method isolating a speaker's face and mouth in a sequence of images. [Lallouache88] describes a system for the measurement of lip shape and jaw movement. [Li95] used a Hough transform to identify the straight lines of the cheek and curved chin lines to characterize the shape of the face. [Perkell92] describes an electromagnetic system used to measure the shape of the vocal tract. [Petajan96] describes a facial feature localization system used to measure the inner contour of the lips for use in an automatic speech reading system. [Sbottka98] describes a system that localizes facial contours within a sequence of images and isolates facial features within the facial contour. [Chan99] evaluates the performance of 4 motion trackers based on g-snake for tracking the mouth in a sequence of images. [Kaucic96] presents a real-time lip tracking system that provides accurate lip shape tracking without the use of makeup to mark the lips for use in an audio video speech recognition system.

1.2.13 Color models

[Swenson98] presents a method of real time conversion of the RGB color model to other color models (CMY, YIQ, HSV HLS, CIEXYZ, CIExyY, CIELa*b* and YcrCb). [Smith96] introduces a new color model HWB explains the advantages of this model over HSV and HSL.

1.2.14 Lip-reading

[Blokland98] argues that low bit rate video in an audio-video communication system can be a distraction instead of help. [Bothe93] describes a text to visual speech synthesis system using 38 key-pictures to aid people learning to lip-read. [Bothe94] presents a system to aid people learning to lip-read that models visual speech using artificial neural

networks. [Finn88] created a system to perform automatic speech recognition that uses only video images of a speaker's lips as input. [Montgomery86] created an automatic speech recognition system whose only input is distances between specific points on the speaker's face marked by reflective dots. [Luettin96a] uses parameterized lip shape and intensity information from the mouth area to create speech reading system. The described system uses principal component analysis and hidden Markov models. [Luettin96b] create a limited vocabulary speaker independent recognition system with hidden Markov models. The contour of the speaker's lips are parameterized with active shape models.

1.2.15 Speech coding, Pitch/Formant estimation LPC analysis

[Bouabana98] studied the profiles of the tongue in an X-ray image sequence in order to model the articulatory movements to the acoustic signal produced. [Choi00] propose a method of incorporating the line spectrum pairs (LSPs), used in voice coding, into a voice recognition system. [Markel72] proposes a method of estimating the pitch period of a speech segment for use as an excitation source in a linear predictive voice-coding synthesizer. [Painter96] has implemented several voice coding algorithms namely FS-1015 LPC-10e, the FS-1016 CELP, the ETSIGSM, the IS54 VSLP, the G721 ADPCM, the G.722 subband, and the G.728 LD-CELP under the MATLAB environment. [Pham98] describe a new geostatistical approach to the linear prediction analysis of speech. [Schroeder85] introduces linear predictive analysis and discusses some basic aspects related to the LP speech analysis and coding. [Woodard97] studied the performance of six different types of Codebook Excited Linear Predictive voice coder/decoder systems with bit rates between 8 and 4kbit/s. [Childers95] describes a new model of the glottal source excitation based on polynomials for a linear predictive synthesizer. [Rouat97] proposes a vocal source model to estimate the pitch and to make the voice/unvoiced decision for a linear predictive voice synthesizer system. [Lee96] used Shannon's information theory to judge the suitability of speech features for use in speech recognition systems.

1.2.16 Image coding

[Huitao00] discusses a real-time hierarchical algorithm for use in a model-based video coding videophone communication system. [Kim00] presents a generalized predictive shape coding algorithm for the vertex-based coding of the boundary of an object within a digital image. [Musmann95] proposes a layered model based video coding system for the compression of videophone image sequences. [Zhang98] describes a method of automatic model molding of the general face model to the specific user's face. [Zhong98] proposes a new algorithm for the morphing of 3D wireframe models of the face and head.

1.2.17 Vector quantization

[Gray00] introduces the fundamentals of vector quantization. [Kovesi99] proposes a weighted Euclidean distance measure to perform vector quantization of Line Spectrum Pair parameters.

1.3 Biological basis of human voice production

Human speech is a complex process involving several different structures and organs. This section presents a review of the basic biological elements involved in the production of human speech [Kunt88], [Adjoudani97], [Rabiner93] [Flanagan70].

1.3.1 The anatomic elements of the human speech production system

The anatomic elements involved in the production of the human voice can be divided into 3 principal groups. The chest cavity that produces air pressure, the larynx that creates sound as air passes through it, and the vocal tract that modifies the sound produced by the larynx.

1.3.1.1 Chest and lungs

Within the chest cavity there is a muscle called the diaphragm that helps us to breathe by puling air into and, pushing air out of our lungs. To speak we must lower our diaphragm creating a negative pressure in the chest cavity and therefore pulling air into our lungs. Once the lungs are filled with air the diaphragm can push up creating a positive pressure in our chest cavity. This positive pressure creates an airflow that is in turn used in the production of speech.

1.3.1.2 Larynx and vocal cords

The vocal cords within the larynx are not really cords but rather folds of tissue. These folds create sound by vibrating. The vibration cycle begins when the muscles within the larynx push the vocal folds together creating an air tight seal. Next the diaphragm pushes on the air-filled lungs creating air pressure below the vocal folds. This air pressure forces the cords to open to allow some air to escape. As the air rushes past the vocal folds it creates a decreased pressure (known as the Venturi effect) between

the vocal folds allowing the vocal folds to come together and seal again. Air pressure below the vocal folds then builds up causing the vocal folds to open and allow air to escape and the cycle continues as such. This cycle creates quasi-periodic pulses of air. The frequency of vibration depends on the pressure exercised by the muscles within the larynx that are pushing the vocal folds together. The amplitude of the voice signal is

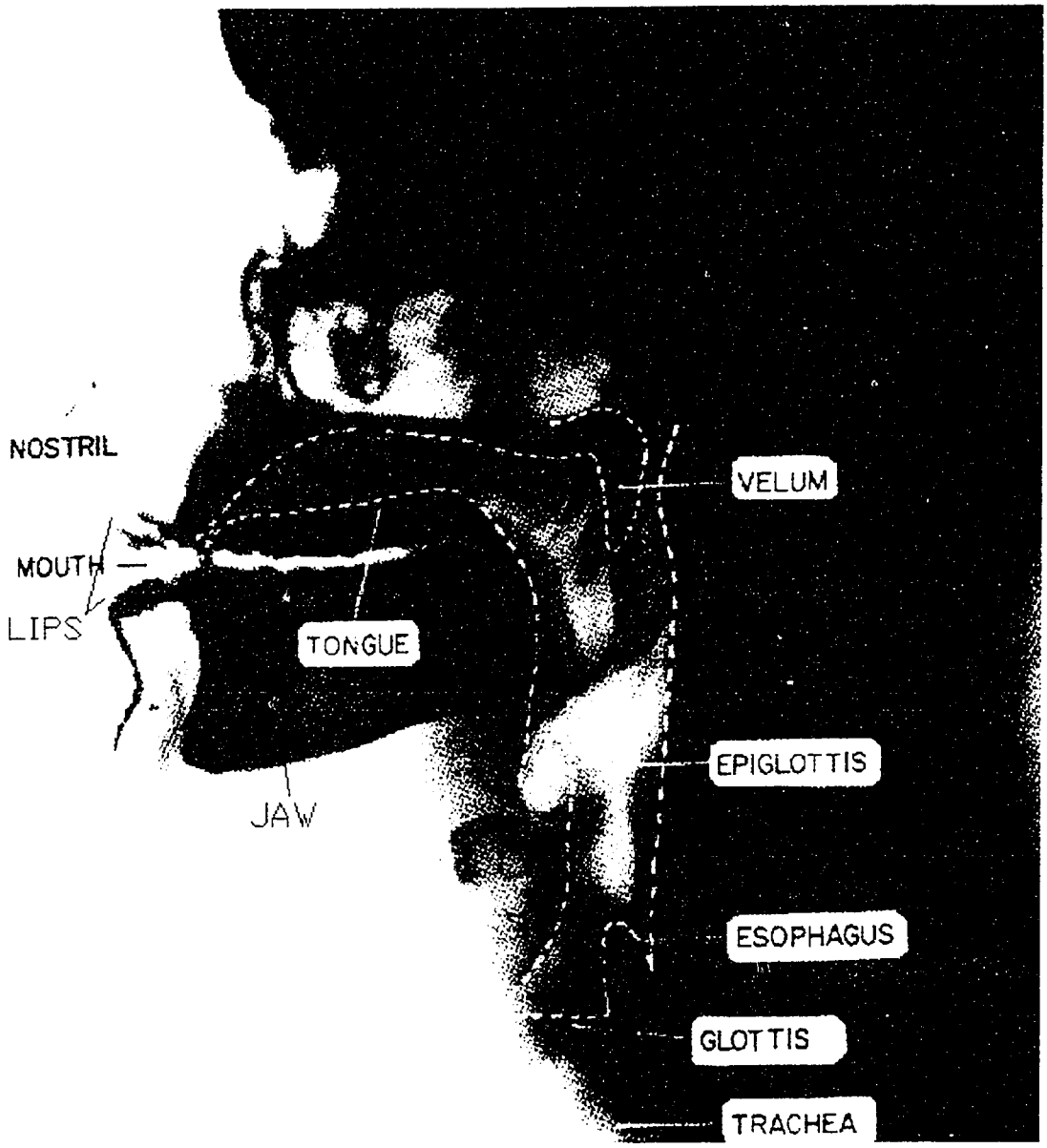

Figure 8 Human vocal tract X-ray adapted from [Flanagan70].

proportional to the air pressure applied to the vocal folds. The higher the air pressure the more loudly the vocal folds vibrate. The vocal folds do not vibrate for each type of sound used in speech. The vocal folds vibrate only for what are known as voiced

speech sounds. There also exists what are known as unvoiced speech sounds. When creating unvoiced sounds the air pushed up from the lungs simply flows through the larynx without creating any sound.

1.3.1.3 Vocal tract

The articulatory elements of the vocal tract above the vocal folds (throat, tongue, lips teeth, jaw, velum etc. see Figure 8) are used to modify the frequency spectrum of the pulses coming from larynx. The transfer function of the vocal tract is applied to the sound signal coming from the larynx. Changing the shape of the vocal tract modifies the value of the vocal tract transfer function. Coupling or uncoupling the nasal cavity from the oral cavity through the velum also modifies the transfer function of the vocal tract. As mentioned before the sound source for unvoiced sounds is not the larynx. For unvoiced sounds, sound is created by a turbulent flow of air through a constricted region in the vocal tract. An example of an unvoiced sound is the sound /f/ (as in the word fat) where the larynx produces no sound but instead the sound is produced by turbulent air created by a constriction at the lips.

1.4 Vocal tract shape and lip shape

There is an obvious cause-effect correlation between the vocal tract shape and properties of the speech signal. One of the basic assumptions of this thesis is that the shape of the lips is intimately related to the state of the vocal tract. This assumption is needed to be able to drive the lip shapes of a synthetic head from the properties of the voice stream.

1.4.1 Articulation

When pronouncing words the vocal tract and lips both articulate to transform the frequency spectrum of the created sound into intelligible speech sounds. Figure 9 shows typical vocal tract positions for some of the vowel sounds of the English language.

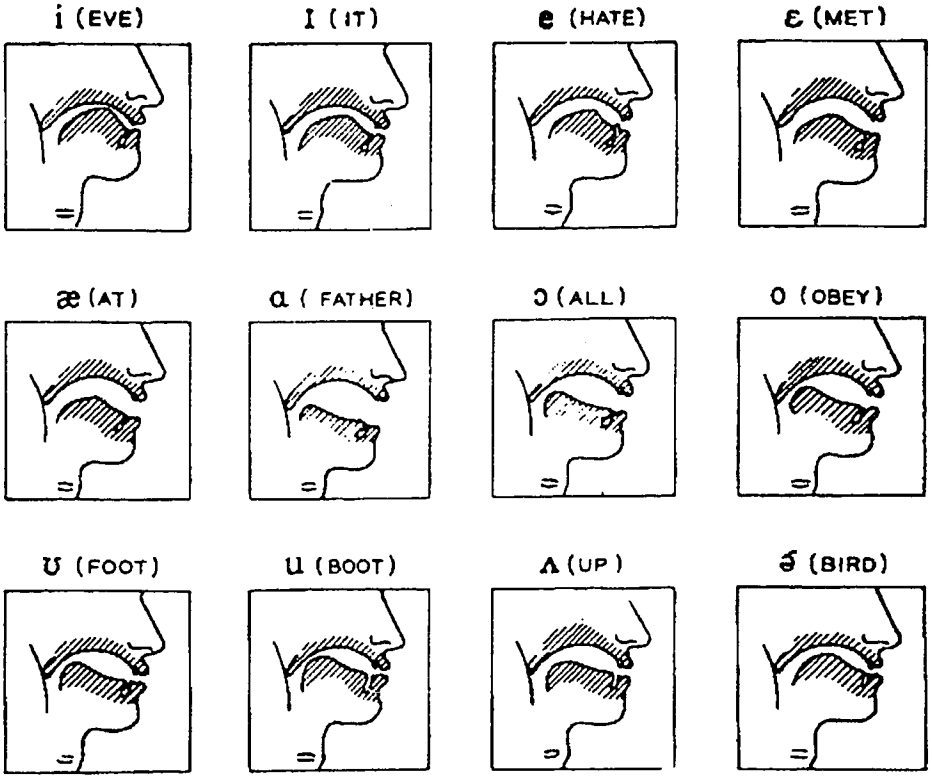

Figure 9 Typical vocal tract positions for some English vowels [Flanagan72].

1.4.2 Lip shape for vowels

Studies have been performed showing how the state of the vocal tract is closely related to the shape of the lips including [Badin94] and [Hani98]. Figure 10 contains example lip shapes for a speaker uttering the vowel sounds of Figure 9.

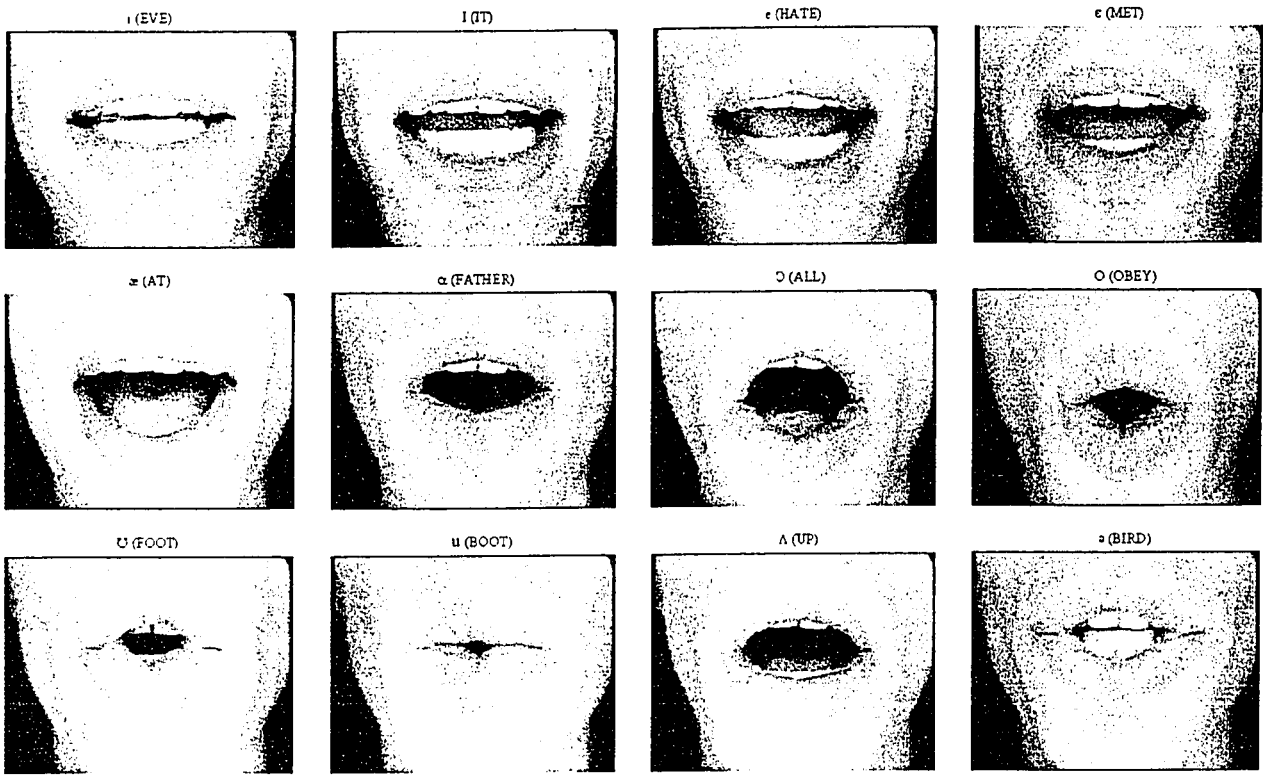

Figure 10 Typical lip positions for some English vowels.

The spectral envelope of a short duration speech segment (the envelope of the magnitude of the frequency transform of a speech segment) is related to the shape of the vocal tract [Rabiner93],[Bouabana98], [Fort98], [Fort96], [Lee96], [Pham98], [Lindblom71]. Therefore by measuring the spectral envelope of a short duration speech segment the speech sound contained within that speech segment can be identified. Having identified the speech sound we can then estimate the shape the speaker's lip took while uttering that sound.

1.4.3 Extendible beyond vowels

Although the 2 previous figures illustrated distinct lip positions for different vowel sounds the relations linking the spectral envelope and the shape of the lips is not limited to vowel sounds. All of the phonemes have their own speaker dependent, typical lip positions [Badin94] and [Hani98]. However this project passes from the description of the audio frame to the lip shape estimate without the intermediate step of explicitly identifying the phoneme contained within the audio frame.

1.5 References

[Adjoudani97], [Rabiner93], [Flanagan70], [Flanagan72], [Bouabana98], [Fort 98], [Fort96], [Lee96], [Pham98], [Badin94], [Hani98], [Assaf97],[Cordea99],[Lindblom71], [Kunt88], [Moore88].

CHAPTER 2 – Voice analysis

2.1 Voice analysis system structure

This section begins with a description of the structure of the voice analysis part of the lip shape estimating system. The steps taken in the audio analysis are common for both the training stage and the animation stage and are illustrated in Figure 11. The system input is an audio stream and outputs are the cepstral coefficients [Moore88] and an energy measurement of the voice frames.

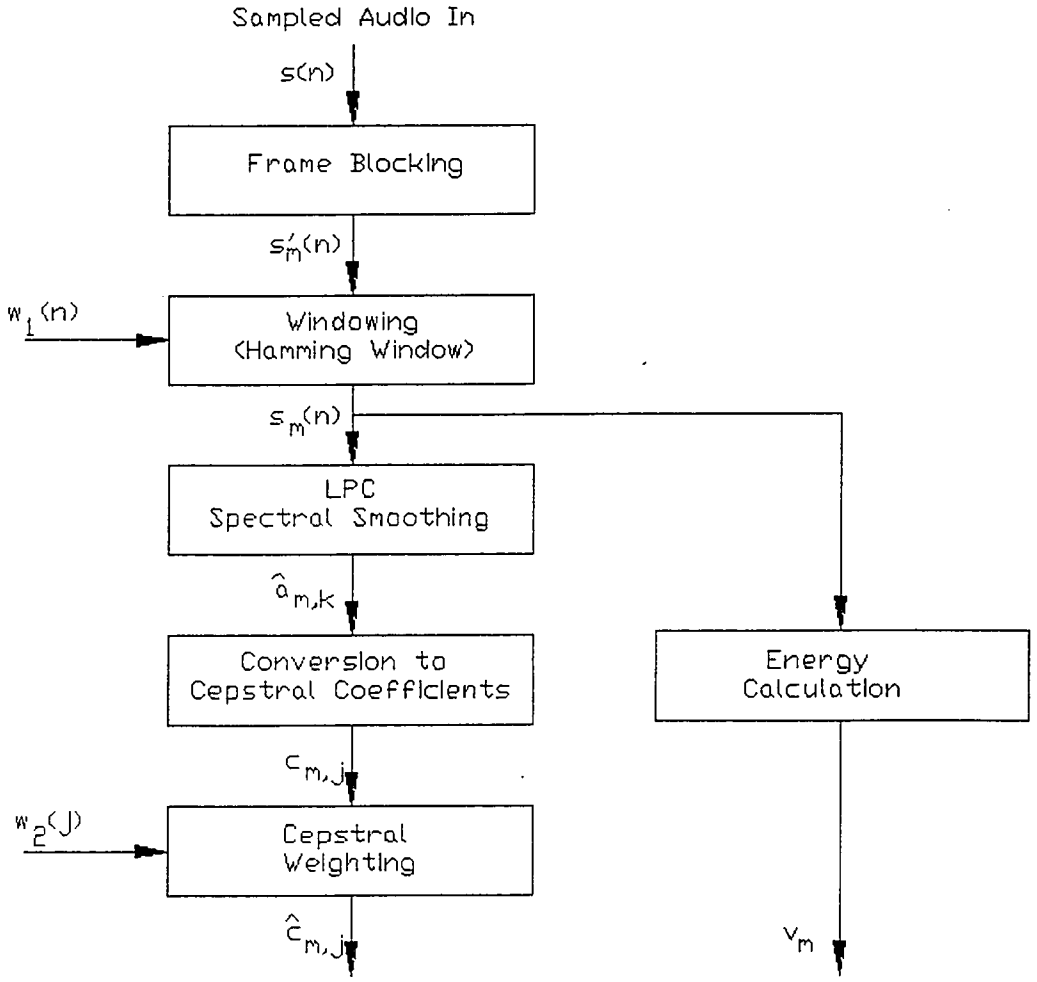

Figure 11 Block diagram of the steps taken when analyzing the audio stream.

2.1.1 Frame blocking

The first step taken in Figure 11 is to take the sampled audio stream s(n) and to cut the stream into overlapping frames $s'_m(n)$. The frames are N samples long. M samples separate the beginning of adjacent frames. M is smaller than N therefore the frames overlap. Figure 12 shows how the frames are cut.

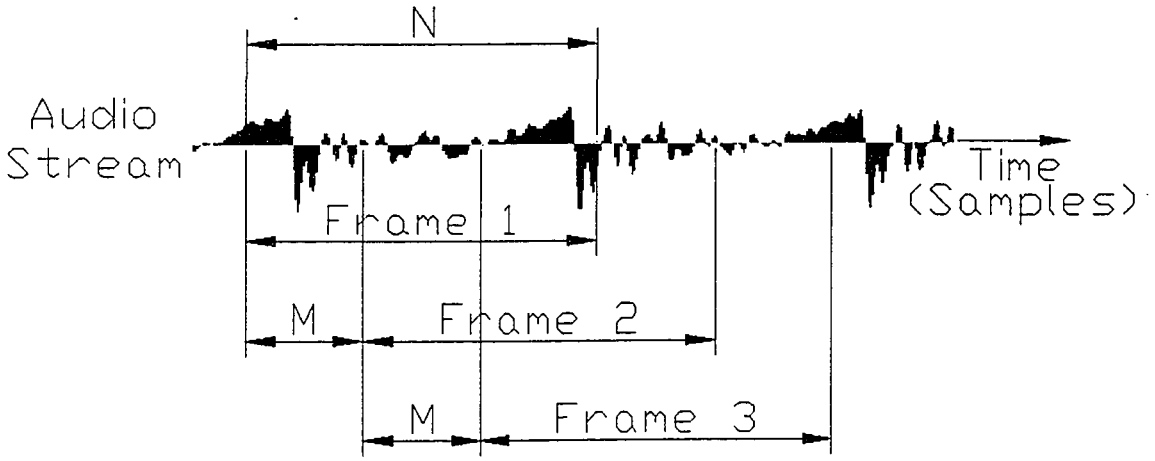

Figure 12 Overlapping audio frames are cut from audio stream.

Cutting the audio signal into frames of around 30 ms in length allows us to analyze the audio frame with the correct assumption that the waveform characteristics are similar throughout the analysis frame [Moore88]. The following defines $s_m'(n)$ which contains the audio samples for the m^{th} audio frame.

$$s'_m(n) = \begin{cases} s(n+mM) & 0 \le n \le N-1 \\ 0 & elsewhere \end{cases} \qquad m = 0,1,2...Y \qquad \text{Equation 1}$$

The next step in the audio analysis is to weight the samples in the newly cut audio frames with a window.

2.1.2 Windowing

After the audio stream is cut into frames, a Hamming window is applied to taper the samples near the edges of the audio frame down to zero. Reasons for applying the Hamming window will be discussed later. The Hamming window is applied as follows

$$s_m(n) = s'_m(n) \bullet \omega_1(n) \qquad \text{Equation 2}$$

Where $\omega_1(n)$ is the Hamming window and it is defined as

$$\omega_1(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad \text{Equation 3}$$

An example illustrating how applying a Hamming window to an audio frame tapers the edges of an audio frame can be found in Figure 13.

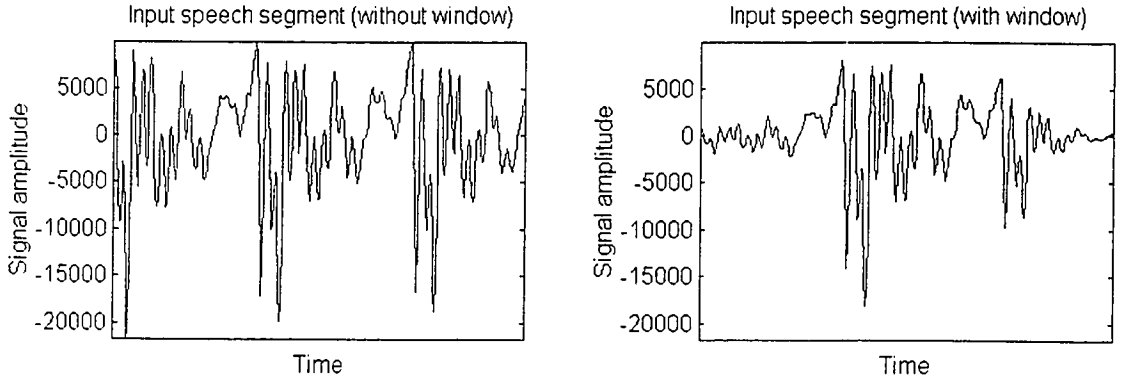

Figure 13 The Hamming window tapers the edges of the audio frames.

Because the waveform characteristics are similar throughout the short duration audio frame, tapering the edges of the frame does not change the general characteristics of the frame and allows the next step of the audio analysis, the LPC analysis, to obtain more accurate results [Rabiner93].

2.1.3 Linear predictive coding analysis

The third step taken in the audio analysis system illustrated in Figure 11 is the linear predictive coding spectral smoothing. The LPC analysis begins by attempting to predict a given audio sample from a weighted sum of the P previous samples. The weights on the previous P samples are the LPC coefficients as in the following equation

$$s_m(n) \approx a_{m,1}s_m(n-1) + a_{m,2}s_m(n-2) + \ldots + a_{m,P}s_m(n-P) \qquad \text{Equation 4}$$

The value P is known as the order of the LPC analysis. In this implementation P=10 [Rabiner93]. We can define $\tilde{s}_m(n)$ as an audio frame created by the linear combination of the LPC coefficients with the audio frame $s_m(n)$ as the following.

$$\tilde{s}_m(n) = \sum_{k=1}^{P} a_{m,k}s_m(n-k) \qquad \text{Equation 5}$$

2.1.3.1 Error signal equations

The difference between the predicted frame and the input frame is the error signal (sometimes known as the residual signal).

$$e_m(n) = s_m(n) - \tilde{s}_m(n) \qquad \text{Equation 6}$$

Substituting Equation 5 in place of $\tilde{s}_m(n)$ we obtain

$$e_m(n) = s_m(n) - \sum_{k=1}^{P} a_{m,k}s_m(n-k) \qquad \text{Equation 7}$$

The LPC coefficients are chosen so the sum of the square of the error signal is reduced to a minimum. In other words, the coefficients $a_{m,k}$ are chosen to minimize the following equation.

$$EER_m = \sum_{n=0}^{N-1+P} e_m^2(n) \qquad \text{Equation 8}$$

Substituting Equation 7 for $e_m(n)$ of the previous equation we obtain

$$ERR_m = \sum_{n=0}^{N-1+P} \left[s_m(n) - \sum_{k=1}^{P} a_{m,k} s_m(n-k) \right]^2 \qquad \text{Equation 9}$$

Now that the error signal is defined the reason for applying the Hamming window to the audio frame can be revealed.

2.1.3.2 Motivation for applying the Hamming window

At the beginning and end of each audio frame there is a potential for large prediction errors (see Equation 4). At the beginning of the frame the system is attempting to create a weighted sum of samples whose values are zero and arrive at a non-zero result. At the end of the analysis frame the system is taking the weighted sum of non-zero samples and trying to arrive at a sum arbitrarily set to zero. The solution to this problem is to taper the values of $s_m'(n)$ near the edges of the analysis frame to zero using a Hamming window.

With the samples near the edges of the analysis frame decreasing slowly to zero the sum squared prediction error is reduced. In Figure 14 we can see 2 error signals each derived from the same time domain speech frame. One frame was multiplied by the Hamming window and the other frame was not. We can see from Figure 14 that the frame that was not multiplied by the Hamming window has a large prediction error amplitude at the beginning and end of the frame. The frame that was multiplied by the

Hamming window has a very small prediction error amplitude at the edges of the frame and the error amplitude in the middle of the frame is smaller as well. A smaller error signal amplitude means that the sum squared prediction error is reduced and the chosen LPC coefficients more accurately represent the qualities of the audio frame in question.

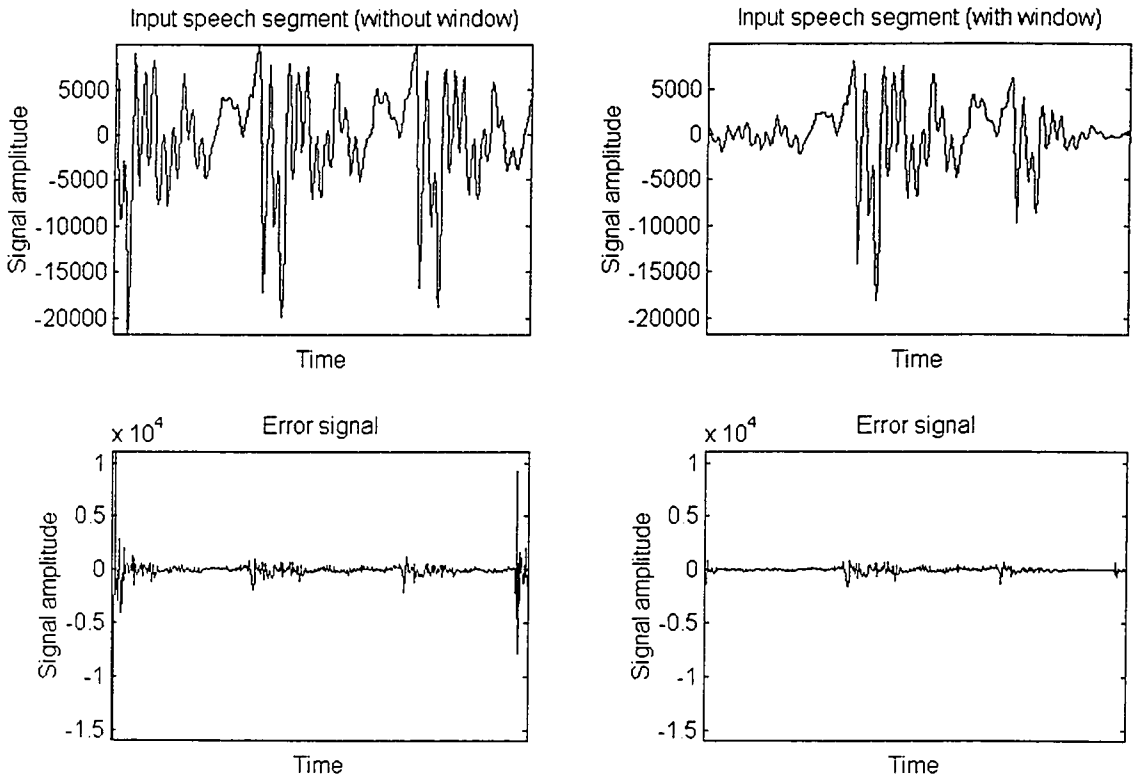

Figure 14 The use of a Hamming window to smooth edges of audio frames.

2.1.3.3 Choice of LPC coefficients

To find the optimum set of LPC coefficients ($a_{m,k}$) that will minimize Equation 9 we take the derivative of Equation 9 with respect to each LPC coefficient and set the derivative to zero.

$$\frac{\partial ERR_m}{\partial a_{m,k}} = 0 \qquad k = 1,2,3..P \qquad \text{Equation 10}$$

Using the chain rule and differentiating with respect to a specific LPC coefficient $a_{m,i}$, with $1 \le i \le P$, the previous equations becomes

$$0 = \sum_{n=0}^{N-1+P} \left\{ 2\left[s_m(n) - \sum_{k=1}^{P} \hat{a}_{m,k} s_m(n-k) \right] \frac{\partial}{\partial a_i} \left[s_m(n) - \sum_{k=1}^{P} \hat{a}_{m,k} s_m(n-k) \right] \right\} \quad \text{Equation 11}$$

Differentiating the second term leaves only

$$\frac{\partial}{\partial a_i}\left[s_m(n) - \hat{a}_{m,1} s_m(n-1) - \hat{a}_{m,2} s_m(n-2) - \ldots - \hat{a}_{m,i} s_m(n-i) - \ldots - \hat{a}_{m,P} s_m(n-P) \right] \quad \text{Equation 12}$$
$$= -s_m(n-i)$$

Resulting in the following equation

$$0 = 2 \sum_{n=0}^{N-1+P} \left\{ \left[s_m(n) - \sum_{k=1}^{P} \hat{a}_{m,k} s_m(n-k) \right] \left[-s_m(n-i) \right] \right\} \quad \text{Equation 13}$$

By re-arranging the terms we obtain

$$\sum_{k=1}^{P} \hat{a}_{m,k} \sum_{n=0}^{N-1+P} s_m(n-i) s_m(n-k) = \sum_{n=0}^{N-1+P} s_m(n) s_m(n-i) \quad \text{Equation 14}$$

Noticing that the previous equation is a function of the difference between i and k, we can rewrite Equation 14 as

$$\sum_{k=1}^{P} \hat{a}_{m,k} \sum_{n=0}^{N-1-(i-k)} s_m(n) s_m(n+(i-k)) = \sum_{n=0}^{N-1-i} s_m(n) s_m(n+i) \quad \text{Equation 15}$$

Remembering the definition of the autocorrelation function

$$r(x) = \sum_{n=0}^{N-1-x} s(n)s(n+x)$$ Equation 16

Equation 15 can be re-written with the autocorrelation function inserted.

$$\sum_{k=1}^{P} r_m(i-k)\hat{a}_{m,k} = r_m(i) \qquad 1 \le i \le P$$ Equation 17

The autocorrelation function is symmetric, r(x)=r(-x) therefore we can write r(|i-k|) in place of r(i-k). Giving the following equation.

$$\sum_{k=1}^{P} r_m(|i-k|)\hat{a}_{m,k} = r_m(i) \qquad 1 \le i \le P$$ Equation 18

The previous equation can be expressed in matrix form as

$$\begin{bmatrix} r_m(0) & r_m(1) & r_m(2) & \cdots & r_m(P-1) \\ r_m(1) & r_m(0) & r_m(1) & \cdots & r_m(P-2) \\ r_m(2) & r_m(1) & r_m(0) & \cdots & r_m(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_m(P-1) & r_m(P-2) & r_m(P-3) & \cdots & r_m(0) \end{bmatrix} \begin{bmatrix} \hat{a}_{m,1} \\ \hat{a}_{m,2} \\ \hat{a}_{m,3} \\ \vdots \\ \hat{a}_{m,P} \end{bmatrix} = \begin{bmatrix} r_m(1) \\ r_m(2) \\ r_m(3) \\ \vdots \\ r_m(P) \end{bmatrix}$$ Equation 19

The resulting matrix is symmetric with respect to the diagonal. All the diagonal values are the same. This kind of matrix is known as Toeplitz and can be solved using the Durbin algorithm.

2.1.3.4 Durbin algorithm

The Durbin algorithm [Rabiner93] is an iterative algorithm used to solve simultaneous equations that can be put in a Toeplitz matrix form. The algorithm is defined with the following equations.

$$ERR^{(0)} = r(0)$$ Equation 20

$$k_i = \frac{\left[r(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} r(i-j) \right]}{ERR^{(i-1)}} \qquad 1 \le i \le P$$ Equation 21

$$\alpha_i^{(i)} = k_i$$ Equation 22

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \qquad 1 \le j \le i-1$$ Equation 23

$$ERR^{(i)} = (1 - k_i^2) ERR^{(i-1)}$$ Equation 24

Equations 20 through 24 are solved one after the other while i takes values of i=1 through to i=P. The peak LPC coefficients $\hat{a}_{m,j}$ for the m[th] frame are found as in the following equation.

$$\hat{a}_j = \alpha_j^{(P)} \qquad 1 \le j \le P$$ Equation 25

2.1.3.5 LPC filter spectrum

Having found the optimum set of LPC coefficients we can now discuss the input-output relations of a filter built using the optimum LPC coefficients. Taking the z-transform of Equation 7 we can arrive at the following transfer function

$$A_m(z) = \frac{E_m(z)}{S_m(z)} = 1 - \sum_{k=1}^{P} a_{m,k} z^{-k}$$ Equation 26

The inverse system can be defined as the following

$$\frac{1}{A_m(z)} = \frac{S_m(z)}{E_m(z)} = \frac{1}{1 - \sum_{k=1}^{P} a_{m,k} z^{-k}} = H_m(z)$$ Equation 27

$H_m(z)$ is an all pole filter. The spectral shape of the filter depends on the choice of the LPC coefficients ($a_{m,k}$). Having found the optimal LPC coefficients by solving the matrix Equation 19 the spectrum of H(z) (known as the LPC spectrum) will follow the general shape of the spectrum of the original audio frame. The higher the order of the LPC analysis the more details of the spectrum of the original audio frame will be preserved in the spectrum of H(z). An example of a LPC spectrum is contained in Figure 15.

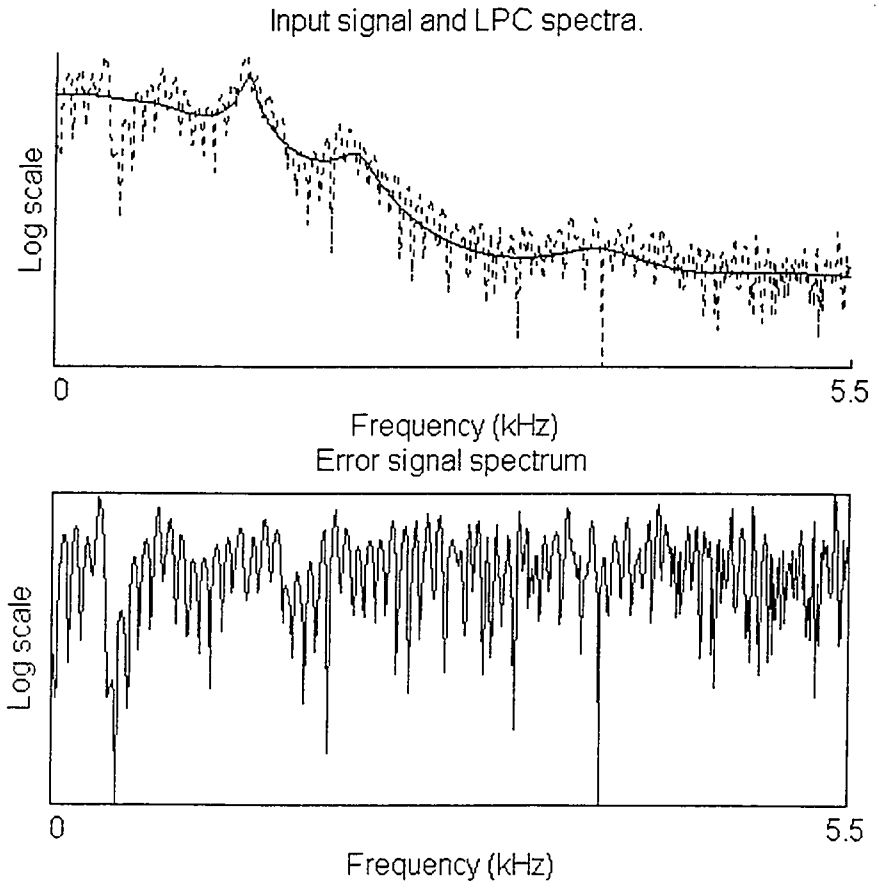

Figure 15 The FFT spectra with the LPC spectra overlapped and the error signal spectrum.

The LPC spectrum follows the overall trends of the spectrum of the input frame. These trends describe the state of the vocal tract of the speaker. Each vowel and consonant sound has a different spectral shape. The spectrum of the error signal is a whitened version of the spectrum of the input signal. Whitened because the overall spectral shape of the error signal is flattened across the spectrum. The error signal is further studied in section 2.2.

2.1.4 Conversion of LPC coefficients to Cepstral coefficients

The following step in the audio analysis system described in Figure 11 is the conversion of the LPC coefficients into the cepstral coefficients. The real cepstrum of a signal is the real part of the inverse Fourier transform of the log of the audio frame spectrum or in equation form as the following:

$$C(j) = \mathrm{Re}(IFFT(Log(abs(FFT(s_m(n))))))$$ Equation 28

These coefficients have been proven to be both reliable and robust when used as recognition features in a speech recognition system; more robust than the LPC coefficients or other methods of representing an audio frame like the PARCOR coefficient or log area ratio coefficients [Rabiner93][Lee96]. The cepstral coefficients derived from an LPC spectrum are a decaying sequence. Because the cepstral coefficients are a decaying sequence the number of coefficients needed to accurately represent a given LPC spectrum is relatively small. In general the number of cepstral coefficients used to represent an audio frame is at least Q=3/2 P [Rabiner93] where P is the order of the LPC analysis. In this particular implementation P is 10. The number of cepstral coefficients chosen to model the audio frames is 20. Having a vector containing only 20 elements modeling an entire block of audio samples makes the cepstral coefficient representation of an audio frame an efficient representation. Another reason illustrating the value of the cepstral coefficients is that there exists a practical distance measure that can be used to quickly determine the similarity or dissimilarity between 2 audio frames. The cepstral distance measure will be presented a little later. To convert the LPC coefficients into cepstral coefficients the following equation are solved iteratively for j = 1 to j=Q. (Q is chosen as 20).

$$1 \le j \le Q \quad Q \approx \frac{3}{2}P$$ Equation 29

$$c_j = \hat{a}_j + \sum_{k=1}^{j-1}\left(\frac{k}{j}\right)c_k \hat{a}_{j-k} \qquad 1 \le j \le P$$ Equation 30

$$c_j = \sum_{k=1}^{j-1}\left(\frac{k}{j}\right)c_k \hat{a}_{j-k} \qquad j > P$$ Equation 31

Now that the method for calculating the cepstrum coefficients is defined we can now continue and see how these coefficients can be applied to the rest of the system.

2.1.4.1 FFT spectrum distance and LPC spectrum distance

To measure how similar or dissimilar two audio frames are we can simply take the magnitude of the difference of the spectra over the range of applicable frequencies. We can take the FFT (Fast Fourier Transform) of the frames and calculate the magnitude of the difference between the spectra. We can also use the smoother LPC spectrum and calculate the magnitude of the difference between the spectra. Figure 16 shows 2 similar but not identical audio frame spectra calculated using both the FFT and the LPC analysis. These 2 frames are adjacent to one another in the audio stream, they are very similar acoustically, and they have the same lip positions. The sum of the difference magnitude between the 2 spectra calculated using the FFT is almost 4 times the sum of the difference magnitude calculated between the 2 spectra calculated using

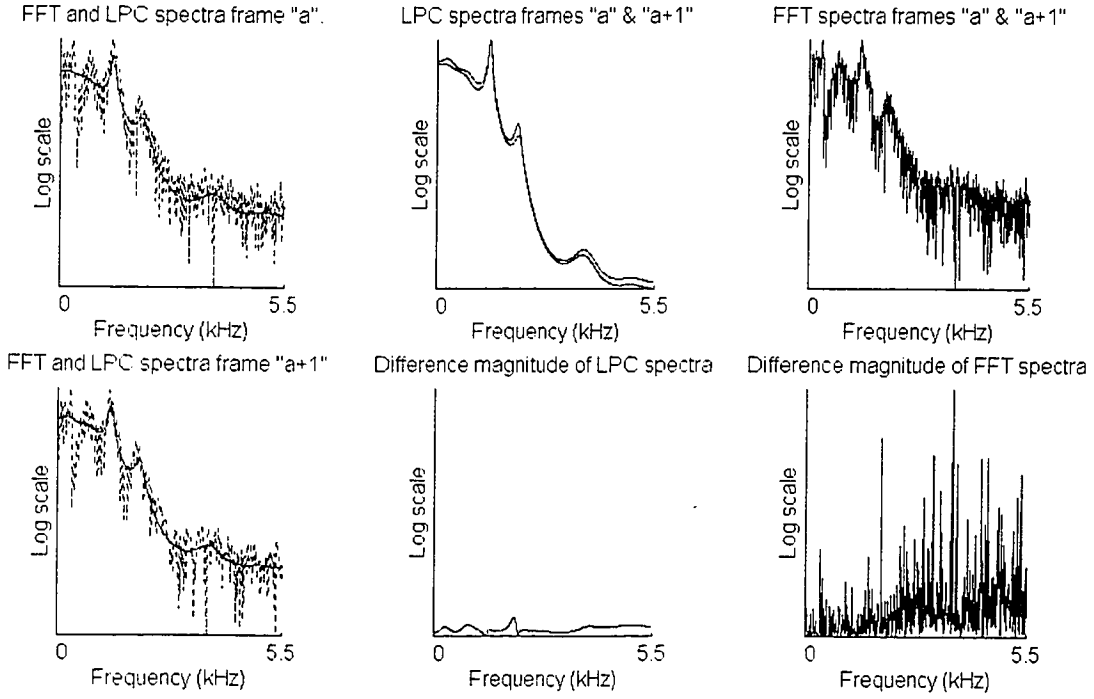

Figure 16 FFT spectral distance magnitude and LPC spectral distance magnitude for 2 similar audio frames.

the LPC spectra. The magnitude difference between the 2 spectra calculated from the FFT is too detailed; it contains much noise like variability. The speech signal is highly variable. Even the same person saying the same sound under the same conditions will have much variation in terms of the FFT difference magnitude. The LPC spectra more closely models the state of the vocal tract and because the LPC spectra is more smooth it discriminates between spectra without considering the fine variations that will exist between very similar sounds. Therefore the cepstral coefficients will be generated from the LPC model of the voice frame not the voice frame itself.

2.1.4.2 Cepstral distance measure

To define the cepstral distance we start by taking the sum of the difference squared between the spectrum of two audio frames.

$$D_{a,b} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\log S_a(\omega) - \log S_b(\omega)\right|^2 d\omega \qquad \text{Equation 32}$$

Where the spectrum of each audio frame is given by

$$\log S(\omega) = \sum_{n=-\infty}^{\infty} c_n e^{-j\omega n} \qquad \text{Equation 33}$$

It can be shown that following equation is equivalent to Equation 33 [Rabiner93].

$$D_{a,b} = \sum_{n=-\infty}^{\infty}\left(c_{a,n} - c_{b,n}\right)^2 \qquad \text{Equation 34}$$

When the cepstrum is calculated from a smooth LPC spectrum the cepstral coefficients form a decaying sequence. An infinite sum is therefore not needed and the sum of Equation 34 can be truncated into a finite sum of Q coefficients. Also $c_n = c_{-n}$ therefore

only the positive coefficients need be included in the sum. The result is known as the cepstral distance measure and is defined by the following equation.

$$D_{a,b} = \sum_{n=1}^{Q} \left(c_{a,n} - c_{b,n}\right)^2 \qquad \text{Equation 35}$$

The sum of Equation 35 begins with the coefficient c_1. The coefficient c_0 is a measure of the power within the audio frame. Some coefficients contain more information relevant to discriminating between speech sounds than others. Therefore a weighting function is applied to the cepstral coefficients to emphasize certain coefficients and de-emphasize others.

2.1.5 Cepstral coefficient weighting

The last step of the main branch of the system illustrated in Figure 11 is to weight the cepstral coefficients. It is a standard technique to weight the cepstral coefficients before using them in the cepstral distance measure [Rabiner93]. The cepstral coefficients are weighted so that the coefficients that contain more noise are de-emphasized while the coefficients that contain more useful information are emphasized. The cepstral coefficients weighting function is as follows:

$$\hat{c}(j) = c(j) \bullet \omega_2(j) \qquad \text{Equation 36}$$

$$\omega_2(j) = \left[1 + \frac{(Q-1)}{2}\sin\left(\frac{\pi(j-1)}{(Q-1)}\right)\right] \qquad 1 \le j \le Q \qquad \text{Equation 37}$$

2.1.5.1 Weighted cepstral distance measure

Now that the cepstral coefficients are weighted according to Equations 36 and 37 the cepstral distance measured can be applied to the coefficients. To measure how similar or how different 2 audio frames are one from the other a we can take the sum of the

distance squared between the corresponding weighted coefficients of the 2 audio frames producing a weighted distance measure as the following:

$$D_{a,b} = \sum_{i=1}^{Q} (\omega(i)c_{a,i} - \omega(i)c_{b,i})^2 \qquad \text{Equation 38}$$

Substituting Equation 36 into Equation 38 we obtain the weighted cepstral distance measure.

$$D_{a,b} = \sum_{i=1}^{Q} (\hat{c}_{a,i} - \hat{c}_{b,i})^2 \qquad \text{Equation 39}$$

Geometrically this comes down to measuring the squared Euclidean distance (in the weighted cepstral coefficient space) between the 2 sets of coefficients. As an example, consider a 2 dimensional weighted cepstral space (c1, c2) and a set of vectors $\bar{C}1$ to $\bar{C}11$. To find which vector of the set of vectors $\bar{C}1$ to $\bar{C}11$ is the most similar to a given input vector $\bar{C}in$ we find the distance separating $\bar{C}in$ with all other vectors. A graphical representation of such an example is given in Figure 17. From Figure 17 we see that

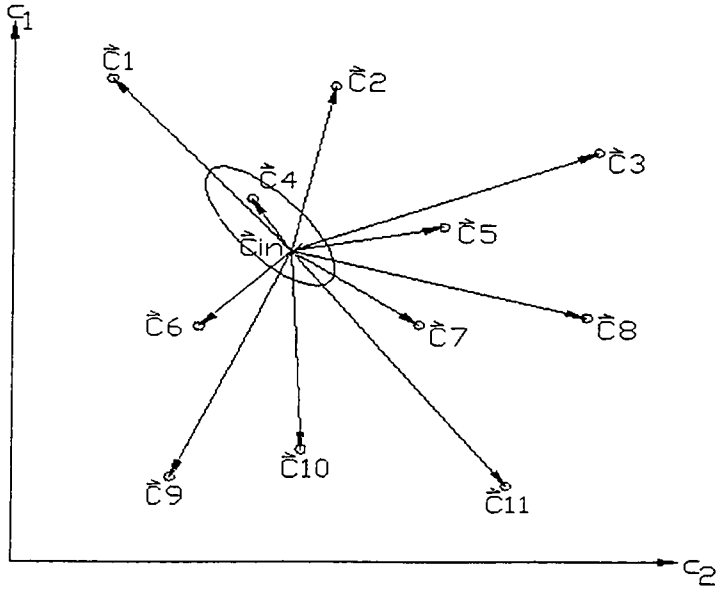

Figure 17 2D example using cepstral distance.

the vector $\bar{C}in$ is most similar to the vector $\bar{C}4$ because the distance between $\bar{C}in$ and $\bar{C}4$ is the shortest.

The cepstral vectors used to represent the audio frames have 20 coefficients creating a 20 dimensional space. The image in Figure 17 should therefore be extended to a 20 dimensional figure. This cannot be represented using ink and paper but the concept behind measuring the distance between vectors remains the same.

2.1.6 Energy calculation

The final box of the Figure 11 to be discussed is the audio frame energy calculation. The energy of an audio frame is simply the sum of the samples squared. Having calculated the autocorrelation function of the audio frame during the calculation of the optimum LPC coefficients, we have already calculated the energy value for that frame because the energy of a frame is also the 0^{th} term of the autocorrelation function. The energy value is used to establish a threshold value for background noise in the training stage. In the animation stage, the system uses the energy value to decide weather the speaker is producing a sound or weather there is only background noise within a particular frame. If the energy value of a given audio frame calculated during the animation stage is below the threshold calculated during the training stage, the system detects that the speaker is producing no sound (in between words for example). If the speaker is not speaking the shape of the lips remains undefined, the system cannot predict the shape of the lips from the voice signal if there is no voice signal.

2.2 LPC voice coding and decoding

LPC analysis is useful for voice recognition and also voice coding/decoding. This chapter briefly explains how the LPC analysis elaborated in the previous chapter can be applied to encoding and decoding the voice signal using a low bit rate for the transmission of the voice signal.

2.2.1 LPC voice production model

The LPC analysis benefits from a model that is useful for both the analysis of voice for use in a voice recognition system and also in the coding of voice signal. The LPC model can be used to reduce the bit rate needed to transmit and store voice because groups of samples (audio frames) are modeled using only the parameters of the LPC voice model. To synthesize the voice signal these parameters are used to animate the voice model. The voice synthesis LPC model is contained in Figure 18.

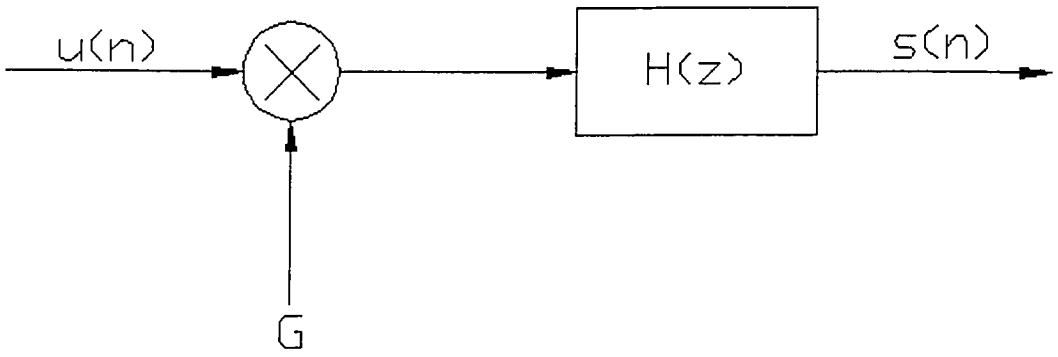

Figure 18 Block diagram of an LPC voice synthesizer [Rabiner93].

H(z) is the all pole filter defined from the optimal LPC coefficients (see section 2.1.3.5). The input to the system is the excitation source. G is the gain applied to the excitation source and S(n) is the output generated by the system.

2.2.2 Excitation source

The excitation source is a synthetic version of the vocal source. For voiced sounds the vocal source are the periodic pulses of air produced by the larynx. For unvoiced sounds the larynx does not create any sound instead the sound is created by turbulent airflow through some part of the vocal tract. Many methods of modeling the vocal source exist [Rabiner93], [Cranen96], [Childers95], [Flanagan70], [Fort96], [Woodward97], [Kunt88]. Perhaps the most basic method of modeling the vocal source is illustrated in Figure 19. This figure shows the voice source modeled using a switch that alternates between 2 voice source states. For voiced sounds the vocal source is an impulse train and for unvoiced sounds the vocal source is random noise.

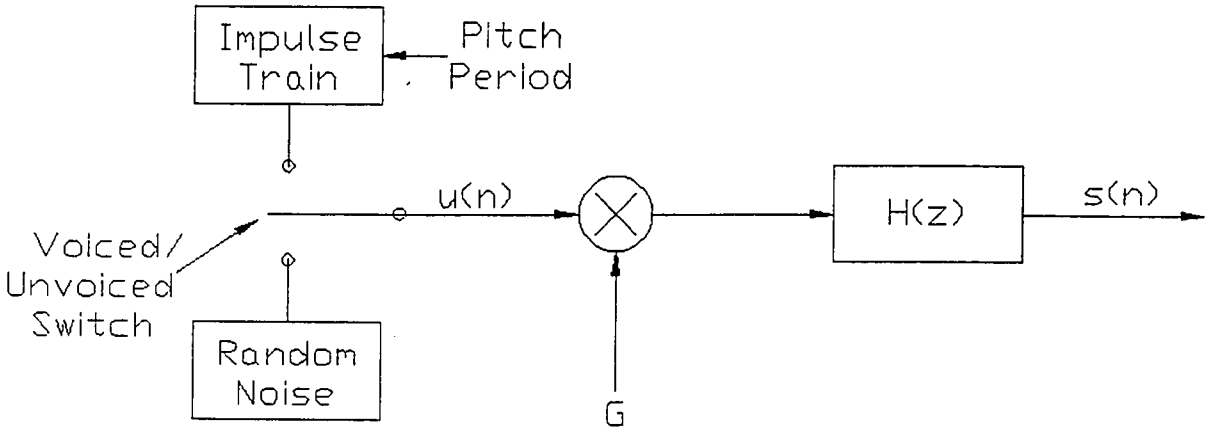

Figure 19 LPC voice synthesizer with vocal source model [Rabiner93].

The pitch period shown in Figure 19 is the fundamental frequency of vibration of the larynx. This value is calculated from the error signal (see Figure 15). There are many methods of extracting the pitch period including [Fort98], [Markel72], [Rouat97]. This vocal source model as presented in Figure 19 produces voice sounds that are not very realistic or conform to the speaker's voice. Other more complex methods of modeling the vocal source signal exist but they are beyond the scope of this document. The reader is referred to the reference section at the end of this chapter and to annex 1 for more information.

2.2.3 Bit rate savings

The bit rate needed to implement the general audio coding system outlined previously depends on the specific application. An uncompressed voiced signal captured at 11.025 kHz 16 bits/sample has a data rate of 176400 bits/sec. Some quality LPC audio coding systems can code voice at 8000, 4800 and 2400 bits/sec [Woodard97], [MELP algorithm in annex 1]. High compression voice coding systems can code voice (with varying quality) at a rate all the way down to 1000 bits/sec [Rabiner93]. See annex 1 for a description of the 2400 bit/sec MELP vocoder created by Atlanta Signal Processors Inc.

2.2.4 Motivation for using LPC analysis

The first steps of the LPC analysis allowing the prediction of the shape of the lips from the voice and the first steps of some LPC voice coder/decoders are identical. Therefore, the cepstral coefficients needed to predict the shape of the lips can be derived from the parameters used to code and synthesize the voice signal. An audio-video communication system that uses LPC coded voice could use the audio coding parameters to predict the shape of the lips. This would allow the shape of the lips to be extracted from the coded voice signal.

2.3 References

[Boubana98], [Choi00], [Fort98], [Fort96], [Painter96], [Pham98], [Rabiner81], [Rabiner77], [Rabiner95], [Schroeder85], [Fort98], [Fort96], [Painter97], [Woodard97], [Childers95], [Rouat97], [Moore88], [Kunt88], [Rabiner93], [Cranen96], [Flanagan70], [Markel72]

CHAPTER 3 – Image analysis

3.1 Sensors and shape capture

Audio capture has become a trivial task with the availability of today's computers. A computer with a regular sound card and microphone are about all that is needed to digitally capture sound (see audio capture conditions in section 5.3.2 for other constraints for quality audio capture). Lip shape capture however is not yet evolved to the point where standardized methods have been chosen and become commercially available. This section reviews some methods that can be used to digitally capture the shape of the lips jaw position or other elements relevant to voice production. Some of the strengths and weakness of each method are explained.

3.1.1 Mechanical measurement

Mechanical methods of measuring the shape and movement of the lips have been used in the past. [Abbs73] used a system as illustrated in Figure 20 to measure the vertical position of the lips and jaw by measuring the current in a circuit.

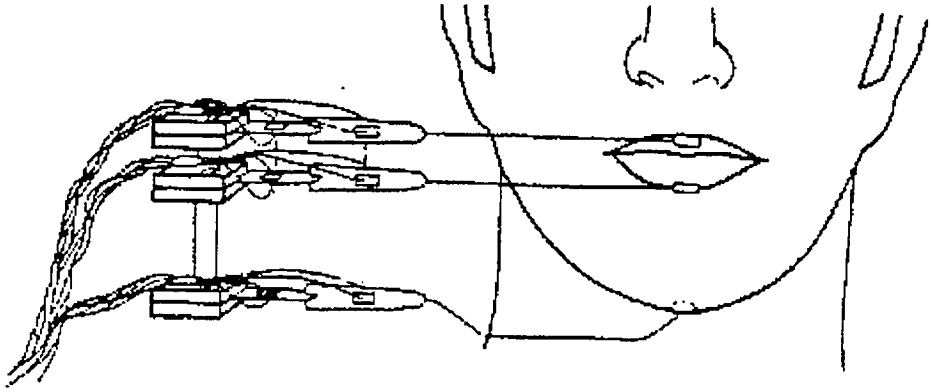

Figure 20 Mechanical lip shape capture by [Abbs73].

The ends of the sensors are glued to the surface of the skin above and below the lips and also on the chin. The up and down movement of the lips and chin made while speaking change the position of the ends of the sensors. This in turn causes the current in an electrical circuit to vary. By measuring the electrical current, the position of

the ends of the sensors can be determined. This method places a large restriction on the speaker because he cannot move his head without causing errors in the measurements. Other more user friendly and accurate methods of lip shape measurement have been developed since the publication of this paper.

3.1.2 Optical point tracking and electromagnetic vocal tract measurement

[Hani98] used a combination optical tracking as well as electromagnetic tracking. Infrared LEDs were glued on the speaker's face. Electromagnetic sensors [Perkell92] were used to capture the positions on the tongue that were hidden from view.

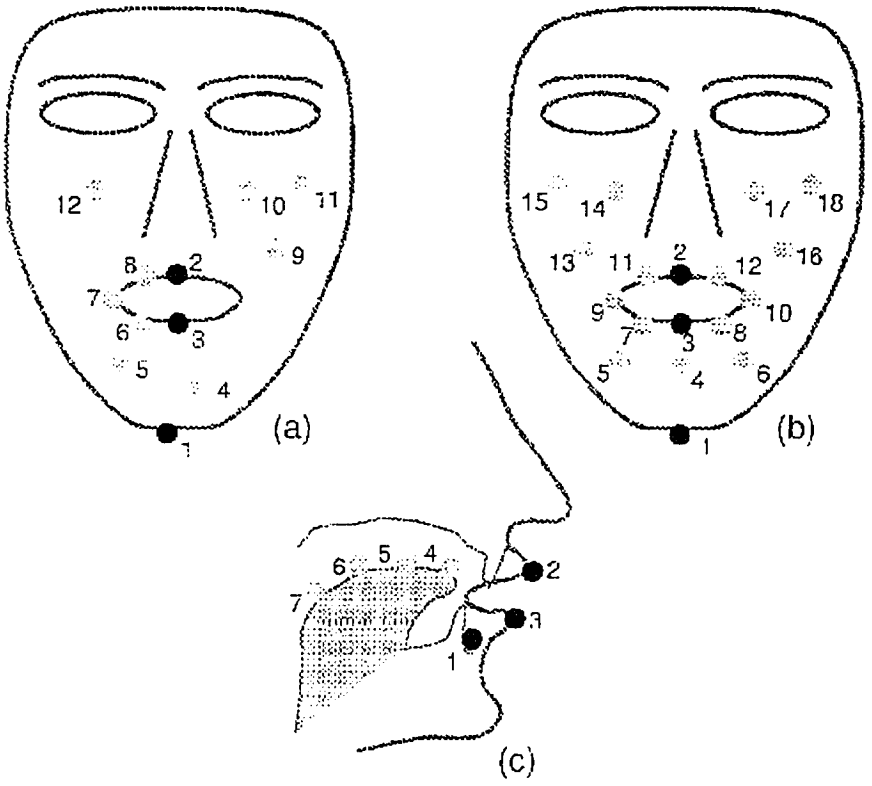

Figure 21 Combination optical and electromagnetic lip shape/vocal tact shape measurement[Hani98].

One advantage of using infrared LEDs is that the infra red light emitted, allows the LEDs to be easily isolated from all other objects in a camera's view. Because the LEDs are

small points of light the measurement can be precise. One disadvantage is that the LEDs cannot be used where they cannot be seen (e.g. inside the mouth).

The electromagnetic approach is very useful when trying to measure the position of objects that are hidden from view. The electromagnetic sensors are small coils of conductor. A magnetic field is generated at some known, constant position near the speaker. The current induced in the coil is proportional to the magnetic field at the coil, which is in turn proportional to the distance separating the coil from the electromagnetic source. Therefore, the measurement of the current induced into the coil can be transformed into a measurement of the coil's position is space. Placing several small magnetometers along the vocal tract is one of the few ways of safely and accurately measuring the movement of the vocal tract and tongue. The electromagnetic measurement method could also be used on the visible parts of the face, such as the jaw or lips, as well.

3.1.3 Image processing methods with makeup markers

Parts of the speaker's face marked with colored makeup can be isolated from the rest of an image using some image processing techniques. Once interesting parts of an image are isolated, the position of these spots can be tracked. An area of the face (e.g. lips) can be marked with a makeup and the shape of the marked surface can be measured. The shape can be measured using deformable templates (as in this implementation) or other measurements can be taken like, blob size, center point, axis of elongation, moments of inertia, contour signatures, density, number of holes, etc. Using makeup markers is a flexible method of measuring position and shape. Using makeup can speed the processing compared to image processing methods that do not use makeup markers. One drawback of using this method is that the makeup must be visible by the camera for this method to work. Another drawback of using makeup markers is that the user of the system has to carefully apply makeup before the system will work. The author of [Lallouache91] used makeup markers take measurements of the lip and chin position (see Figure 22).

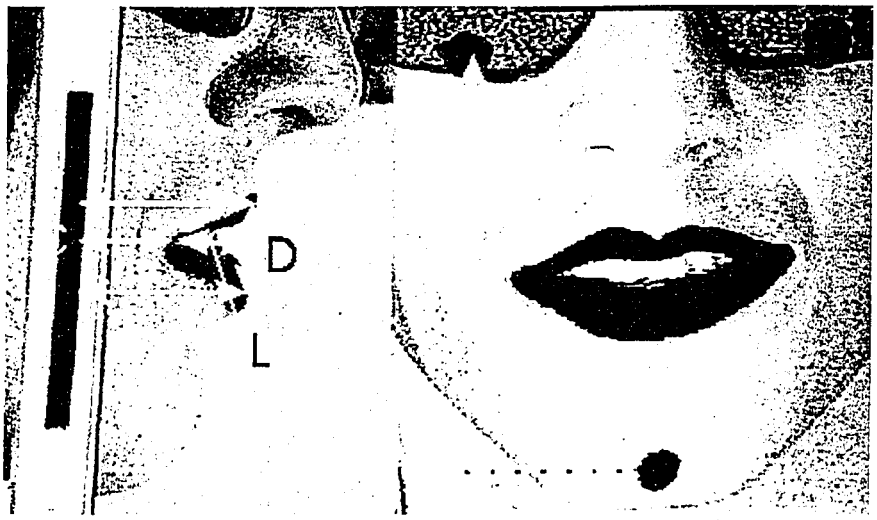

Figure 22 Lip and jaw measurement after [Lallouache91]

3.1.4 Image processing methods without makeup markers

There are many image-processing methods that do not use makeup markers to isolate the parts of interest within an image. Snakes or deformable templates define energy functions that deform their contours according to properties of the images. Some useful image properties are the structure in the image, image color, image texture, image brightness. Often the gradient of the image is a useful tool (for finding edges for example). Hough transforms are useful for finding lines or other shapes within images [Chan99] [Li95]. A face-texture model is used by [Dai96] to isolate the face in a cluttered color scene. The advantage of this approach is that it imposes fewer restrictions of the user. The use of makeup markers or other intrusive sensing devices is not necessary. A disadvantage is that these kinds of systems can be complex and accuracy can vary. The processing time required can also be large.

3.1.5 3D shape from 3D range finders

There exist some interesting methods for capturing of 3D data. Commercial 3D laser scanners exist and can capture a 3D range data and texture at the same time. The authors of [Morishima96] used a Cyberware 3D range scanner. This method acquires

precise data but the cost of such a machine can be high. This method of shape capture requires the subject to remain stationary for several seconds and is not applicable to real time capture of dynamic movements.

3.1.6 3D shape from multiple camera views (stereo vision)

Stereo vision using two cameras allows the capture of 3D position of objects [Franke00]. This is how the human vision system detects depth. What is needed is a method of finding identical points in both of the images captured from the cameras. Once these points are determined the 3D position of these points can be calculated. This three dimensional method of shape capture can be done in real time as long as there exists a quick method of finding corresponding points in the stereoscopic images [Adjoudani97].

3.1.7 3D shape from ultrasound imaging (ultrasonography)

An interesting method of 3D capture of soft deformable tissue is given in [Deng00]. An ultrasonic transducer is used to capture the shape of the lips in real time. To achieve the necessary coupling between the transducer and the face the transducer and the subject's face must be placed underwater. This method is not applicable for use in measuring facial shape while producing speech (speech production underwater would disturb the capturing process). However, this method can be advantageous in other contexts as the 3D information is captured not only about the surface but can penetrate and measure below the surface of the facial structures.

3.1.8 3D lip shape from 2D video

[Basu98] devised a system for the extraction of the 3D shape of the lips from 2D video. They used 3D mathematical models of the lips and combined them with a statistical model. The 3D rigid motion of the head was estimated from the input video and the head pose information was also used to extract the 3D lip shape from the 2D video.

The greatest advantage of this approach is that it frees the subject to move freely within the camera's view.

3.1.9 Chosen method of lip shape capture

In order to keep the image processing algorithms simple and because the lip shape capture is done off line and under laboratory conditions the method chosen for the lip shape capture is that of 2D image processing with makeup markers. The model-building stage needs to be done only once per speaker therefore the inconvenience of applying makeup and other constraints related to the capture process (see section 5.3) are not so important. Whereas, the complexity of the image processing is reduced and the accuracy of the system is increased when makeup markers are used. The complexities of the image processing algorithms are also reduced considerably when working only in 2D.

3.2 Image analysis system structure

This section contains a description of the structure of the image processing methods and algorithms used to determine the speaker's lip shape. A block diagram of the steps taken while performing the video analysis is contained in Figure 23. The input to the system is a single RGB video frame. The output is a parameter vector containing the parameters for the lip model that best fit the given video frame.

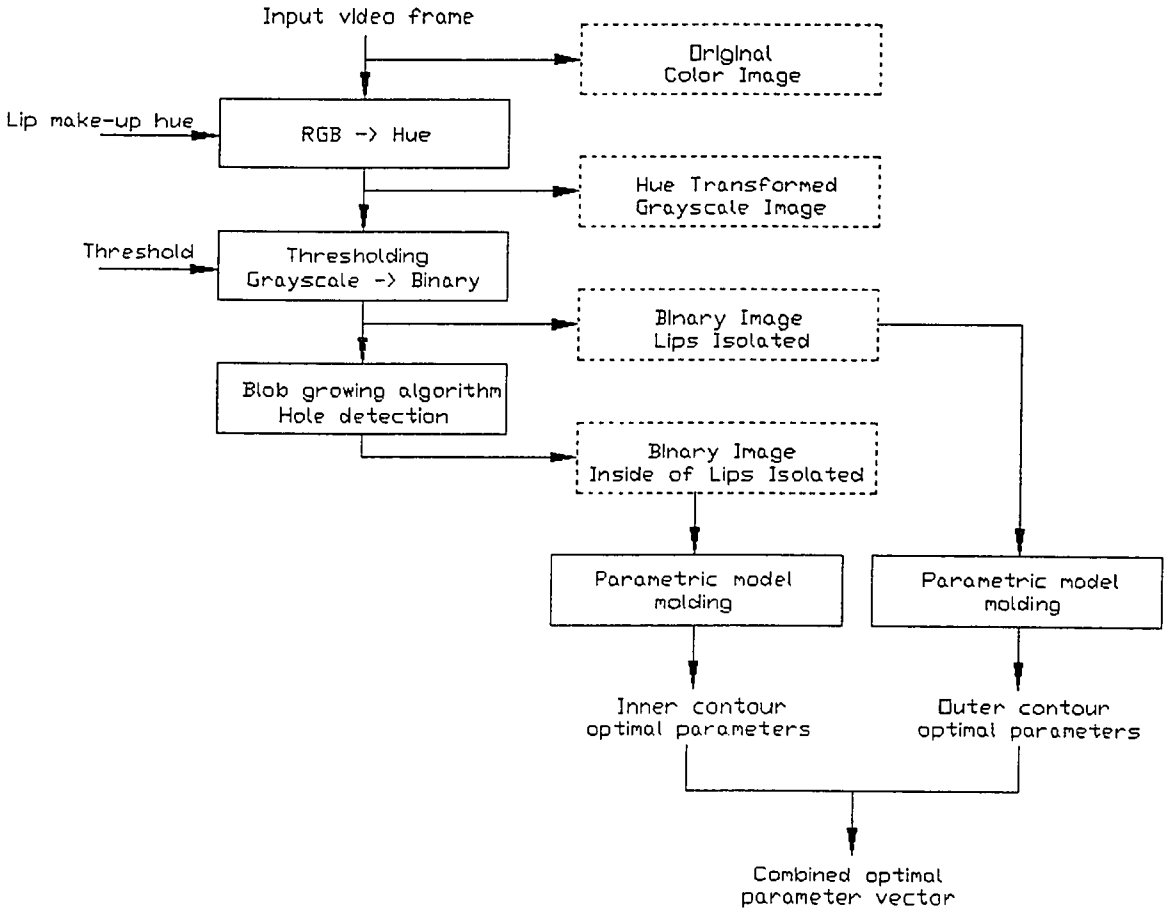

Figure 23 A block diagram of the video analysis system.

Examples of the original color image, the grayscale image and the 2 binary images shown in the block diagram are illustrated in the screen capture of the program AVEXTRACT.EXE in Figure 24. The first image on the upper left is the original input

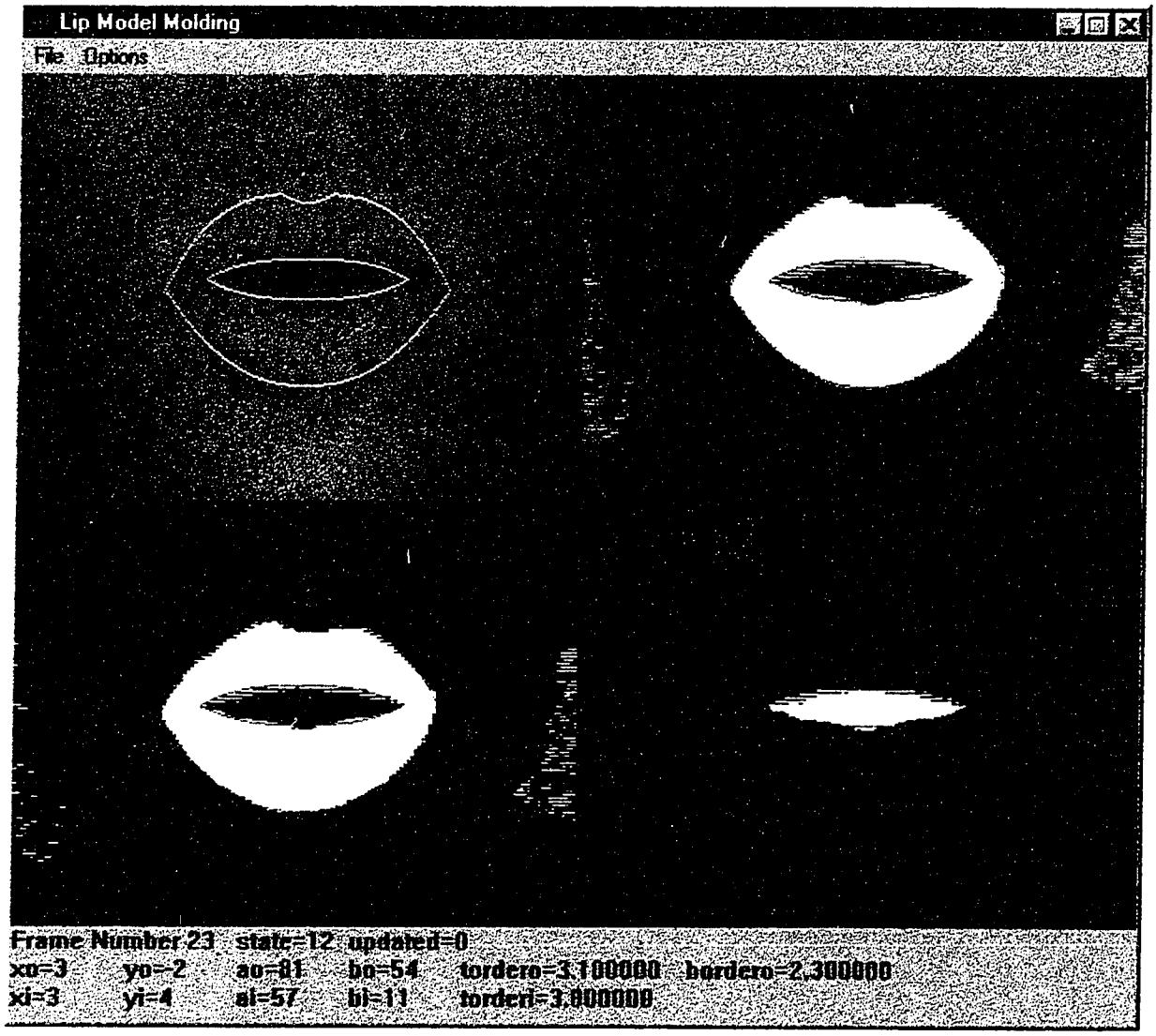

Figure 24 Screen capture of AVEXTRACT.exe

color image. The upper right image is the grayscale image created by transforming the input image according to the hue of each pixel with respect to the hue of the lip makeup. The image to the lower left is the binary image created by thresholding the grayscale image. The image to the lower right is another binary image but in this image the bright pixels are the holes detected using the blob-growing algorithm.

3.2.1 Transforming the input image according to hue

The first step of the image analysis outlined by Figure 23 is the transformation of the input image according to the hue of the pixels. Before going into how the transformation is accomplished some background on the representation of color in computer graphics is discussed.

3.2.1.1 RGB color space

One of the most commonly used methods to represent color in computer graphics is the RGB color space [Smith96], [Swenson98]. The RGB color space represents color using a vector with 3 elements. The 3 elements of the RGB vector represent the Red Green and Blue components of a pixel. This method of color representation is common because it fits well with the technology used in the capture and display of computer graphics. Cathode Ray Tubes (CRTs) used in computer monitors or in televisions use the RGB color model [Berns96]. In the back of the CRT there is an electron gun that fires electrons towards the screen. The screen is covered with a material that glows when struck by the electrons. How brightly the point glows is proportional to the voltage applied when the electrons are fired. The more voltage is applied the brighter the point on the screen. Each color pixel on a color CRT consists not of only one point but of 3 points. Each point is colored either red green or blue. The red, green, and blue components of the RGB vector tell the CRT how much voltage to apply to the corresponding part of the pixel enabling the CRT to display all the colors of the gamut.

For example, for 24 bit RGB, the vector (255,0,0) means that maximum voltage should be applied to the red component of the pixel and zero voltage should be applied to the other two components. This creates a bright pure red pixel on the CRT tube. When all 3 components of the RGB vector have the same value the color of the obtained pixel is a shade of gray. The RGB vector (255,255,255) is pure white, the vector (0,0,0) is black and (128,128,128) is a medium gray. Mixing colors using the RGB color model is

done by weighting the red green and blue components to create the desired color. For example mixing red (255,0,0) and green (0,255,0) creates the color yellow (255,255,0). Figure 25 is a graphical illustration of the RGB color space.

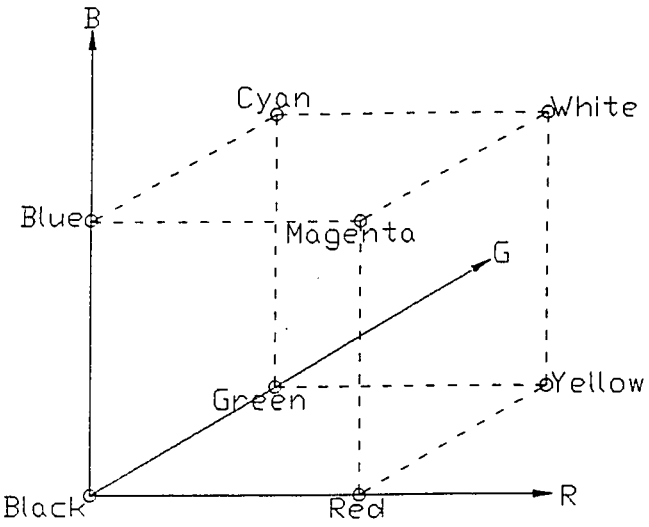

Figure 25 RGB colorspace. Adapted from [Adjoudani97].

The RGB color space can be difficult to visualize, and it is difficult to perform a thresholding operation in RGB color space that will allow us to isolate the pixels of the lips. A more intuitive method of representing color that allows for easy visualization of hue and enables simple thresholding is given in the following section.

3.2.1.2 HSL colorspace

A more intuitive method of representing color is described in the HSL (Hue Saturation Lightness) color model. The HSL model is considered to be more intuitive because it more closely resembles the way an artist uses color. To convert from RGB to HSL the first thing to do is to normalize the red green and blue components.

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B}$$ Equation 40

From the normalized components the saturation value can be calculated as the following.

$$S = 1 - \min(r, g, b)$$ Equation 41

The lightness component is calculated by simply adding the normalized red green and blue elements.

$$L = r + g + b$$ Equation 42

The hue is calculated with the following equation.

$$H = \cos^{-1}\left(\frac{(r-g)+(r-b)}{2\sqrt{(r-g)^2+(r-b)(g-b)}}\right) \quad if \quad b > g \quad then \quad H = 2\pi - H$$ Equation 43

From these equations a 3 dimensional representation of the HSL color space can be visualized. It is illustrated in Figure 26.

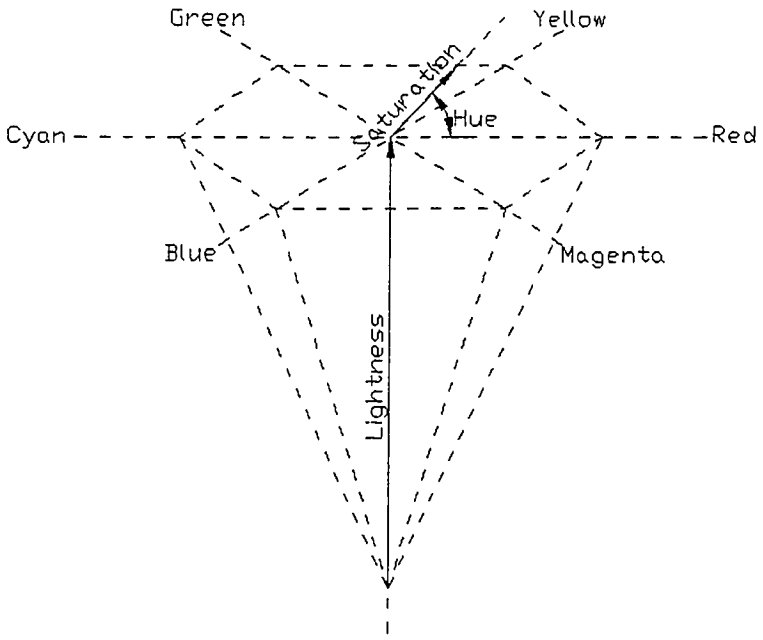

Figure 26 HSL colorspace. Adapted from [Adjoudani97].

The lightness component is a measure of the intensity of the pixel. The higher the lightness the more power illuminates the pixel. The saturation component is a measure

of the purity of a color. Low saturation values means that a color is more pastel and a completely unsaturated (saturation=0) colors are some shade of gray (and the hue is undefined). A high saturation value means that a color is more vibrant. For example a highly saturated red pixel is bright pure red (0,1,1). The most interesting part (as far as this project is concerned) of the RGB to HSL transform is the hue component. A hue color wheel is shown Figure 27. Figure 27 is simply the 3D object contained in Figure 26 viewed from the top. On the color wheel the red green and blue colors are as far away from each other as they can be and are therefore each separated by 120º. The colors in between are defined next with yellow

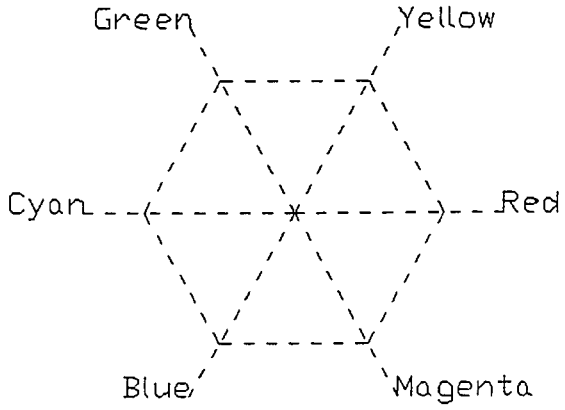

Figure 27 The hue color wheel.

at 60º cyan at 180º and magenta at 300º. Having the colors separated by hue enables us to easily isolate the blue pixels of the blue lip makeup from the other pixels in the image.

3.1.2.3 Input image transformation

The input image from the video camera is captured using the RGB color model. To find the pixels within the image that are part of the lips we have to calculate the hue component of the HSL color space for each pixel of the input image. The original image is transformed to a grayscale image. The transformation does not simply calculate the brightness of each pixel but rather uses the hue information to determine the brightness of the pixels. Pixels from the original image whose hue is near to the hue of the lip

makeup are bright and pixels whose hue is far from the hue of the lip makeup are dark. To determine how 'close' a given hue is to the hue of the lip makeup we measure the number of degrees on the color wheel separating the 2 hue values. See Figure 28 for an example of how the hue measurements are made. The hue of pixel A is much further away, (separated my more degrees) than the hue of pixel B with reference to the hue of the lip makeup. The pixel with the hue B will be brighter than the pixel with hue A in the new grayscale image. Completely unsaturated pixels have a hue that is undefined. In this implementation pixels whose hue is undefined are given a minimum brightness in the grayscale image. An example of the result of such an image transformation is illustrated in center image of Figure 29.

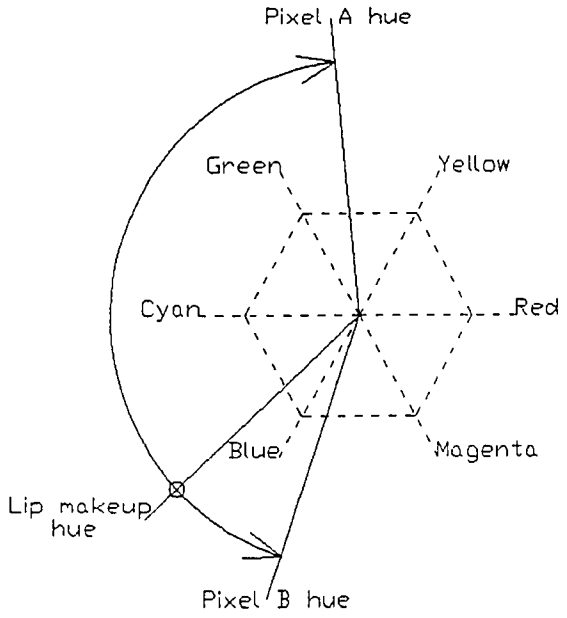

Figure 28 How a given hue is compared to the lip makeup hue.

3.2.2 Thresholding the grayscale image

Once the new grayscale image is created according to how close the hue of the pixels are to the hue of the lip makeup, a new binary image can be created. Performing a thresholding operation on the grayscale image creates the binary image. Each pixel of the grayscale image whose brightness is above a certain threshold is considered to be part of the lips and the corresponding pixel of the binary image is given maximum

brightness (white). Each pixel of the grayscale image whose brightness is below the same threshold is considered to be background and the corresponding pixel of the binary image is given minimum brightness (black). It is this binary image that is used in the model-molding algorithm (the algorithm that fits the parametric lip contour model to the image data).

Figure 29 Example input image, grayscale image and binary image

Figure 29 contains 3 example images. The first image is the input image. The input image is in RGB color. The second image is the grayscale image. The third image is a binary image.

3.2.3 Hole detection

The next step in the video analysis is to isolate the black space between the open lips. Blue makeup is used as a marker to identify the pixels of the lips but no makeup marker can be used to isolate the space between the open lips. This space can be identified because it is a background space completely surrounded by the foreground pixels. A blob-growing algorithm is used to create connected regions called blobs. Adjacent foreground pixels are connected together and adjacent background pixels are connected together. Holes are defined as a background blob completely surrounded by a foreground blob. By applying the blob-growing algorithm to the binary image, created by thresholding the input image, a second binary image is created. The only foreground (white) pixels contained within this new binary image are the holes detected in the previous binary image.

3.2.3.1 Blob growing algorithm description

Blob growing algorithms label adjacent pixels of the same type with the same label. Pixels with the same label are considered connected or part of a 'blob'. The chosen algorithm uses a 4-connected foreground and an 8-connected background. A 4-connected foreground means that foreground pixels will be considered connected only if they are immediately above, below, on the left or on the right of one another (see Figure 30). An 8-connected background means that the background pixels are considered connected if background pixels are immediately above, below, on the left, on the right or on each of the 4 diagonals from one another (see Figure 31).

Figure 30 4-connectivity.
Shaded pixels are 4-connected with the pixel marked with X.

Figure 31 8-connectivity
Shaded pixels are 8-connected with the pixel marked with X.

The algorithm works by passing a 2x2 window over each of the pixels of the input image (except the pixels on the edges of the image) and labeling the pixels according to the pattern observed within the window. The labeling procedure for each of the possible window patterns is contained in table 1. The pixel marked with an X in table 1 is the pixel the algorithm is currently labeling.

Observed Pattern	Labeling Procedure
	Connect to background blob
	Start new object blob
	Connect to background blob
	Connect to object blob
	Connect to background blob and merge background blobs
	Start a new object blob
	Connect to backgound blob
	Connect to object blob
	Connect to background blob
	Connect to object blob
	Connect to background blob
	Connect to object blob and merge object blobs
	Connect to background blob
	Connect to object blob
	Start a new background blob
	Connect to object blob

Table 1 Blob labelling procedure adapted from [McKerrow91].

The blob-growing algorithm has 3 principal labeling procedures, starting a new object/background, connecting object/background and merging objects/background. When starting a new object or background a new label (a counter is incremented and the new value) is assigned to the pixel in question. When connecting an object or

background, the pixel in question receives the same label as the adjacent pixel of the same type. Merging objects or backgrounds means that one of the previously assigned labels will be changed to show that 2 objects or background regions are now considered to be connected (both regions are assigned the same label).

3.2.3.2 Algorithm example

To illustrate how this algorithm can spot holes (background pixels surrounded by foreground pixels) in blobs, an example binary image contained in Figure 32 will be analyzed. The dark squares of Figure 32 represent foreground pixels and the white squares represent background pixels.

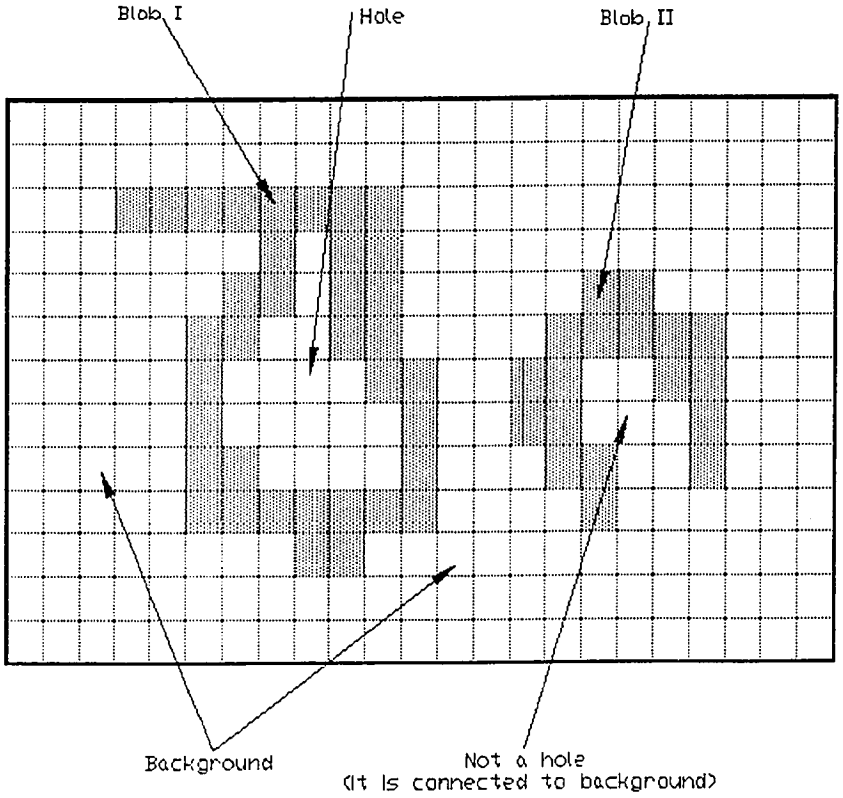

Figure 32 Example binary image containing blobs with holes.

Of the two blobs in Figure 32 blob I has a hole and blob II does not have a hole. Blob II does not have a hole because the background pixels inside the blob are not completely surrounded by foreground pixels. In other words the background pixels within blob II

are connected with the background of the image. Figure 33 shows a series of images of the example image at various stages during the labeling process.

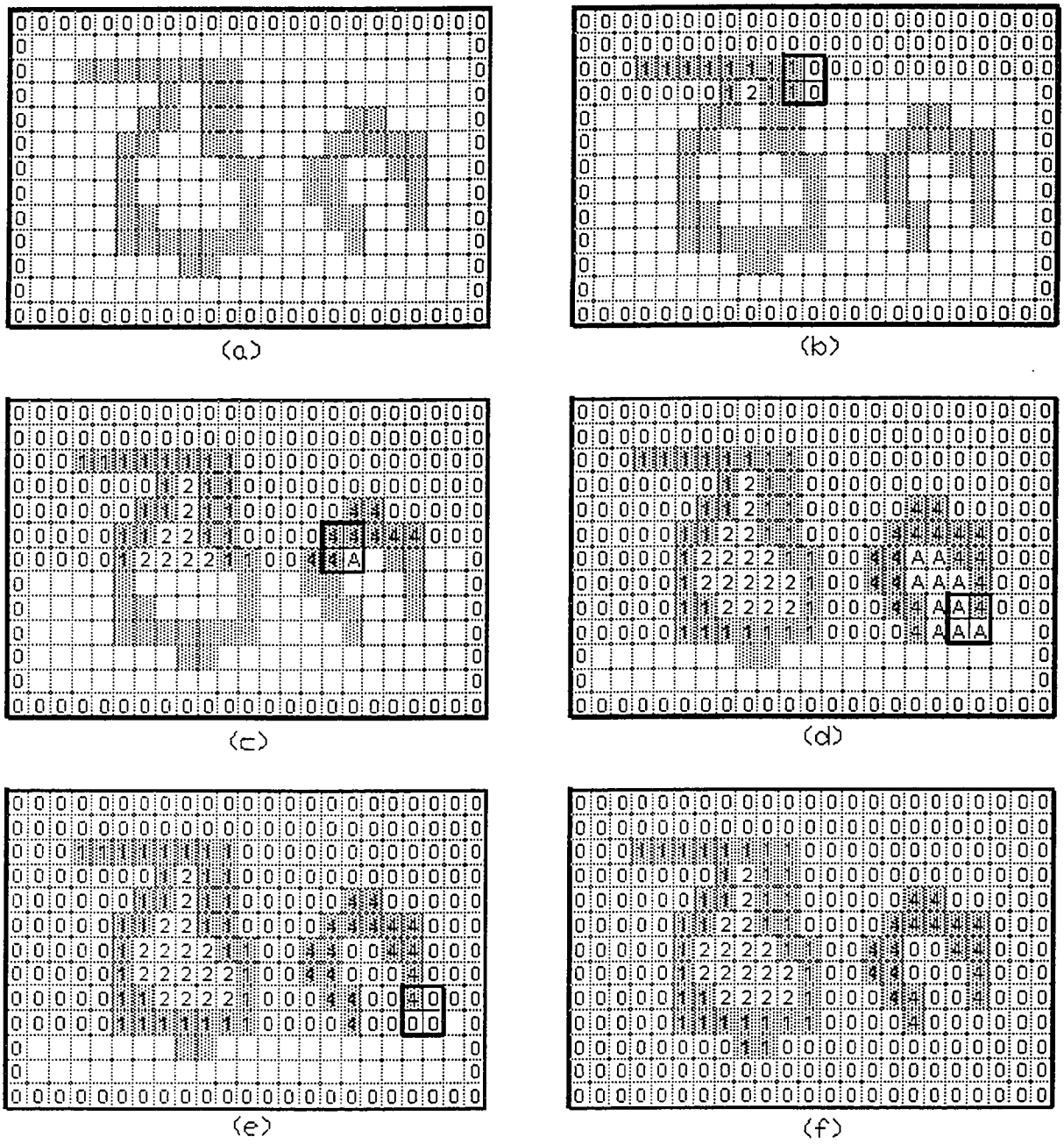

Figure 33 The example image at various stages during the labeling process.

The first step of the labeling process is to force a border of '0' around the edge of the input image. Because the window passed over this image is not passed over the edge

pixels, placing a frame of zeros around the frame of the image insures that '0' will be the label assigned to the background pixels.

As the 2x2 window, defined in table 1, is swept from left to right and top to bottom over the example binary image of Figure 32 we can see that the first foreground pixels that the testing window encounters are the pixels of blob I. The first pixel of blob I is therefore labeled with the label '1' (Figure 33.a). When the testing window encounters the hole within blob I it is labeled with the label '2' (Figure 33.b). By the time the testing window arrives at the blob II the labeling counter has been incremented to 4 and blob II is labeled with the label '4' (Figure 33.c). When the apparent hole in blob II is encountered it is labeled with the label 'A' (Figure 33.c). Further down Figure 33 we see that the apparent hole within blob II, labeled with the label 'A', is connected with the background (Figure 33.d). Therefore, the label 'A' is merged with the label '0' (Figure 33.e). The only background region that is not labeled with the label '0' is the hole within blob I. Using the blob-growing algorithm, holes can be detected as areas of background that are labeled with labels other than the label '0'. The completely labeled image is shown in Figure 33.f.

3.2.3.3 Algorithm limitations

If the background space between the open lips is connected to the background of the image the background space between the open lips cannot be isolated using the blob growing algorithm previously described. The lip blob lips must completely surround the background pixels if the space between the lips is going to be considered a hole in the lip blob. Figure 34 shows an example image with the background space between the open lips and the background of the image connected via a gap in the left edge of the lip blob. The lower left image in Figure 34 should have the space between the lips as foreground pixels and the rest of the image as background. In this case the whole image is background because the space between the open lips is not considered a hole.

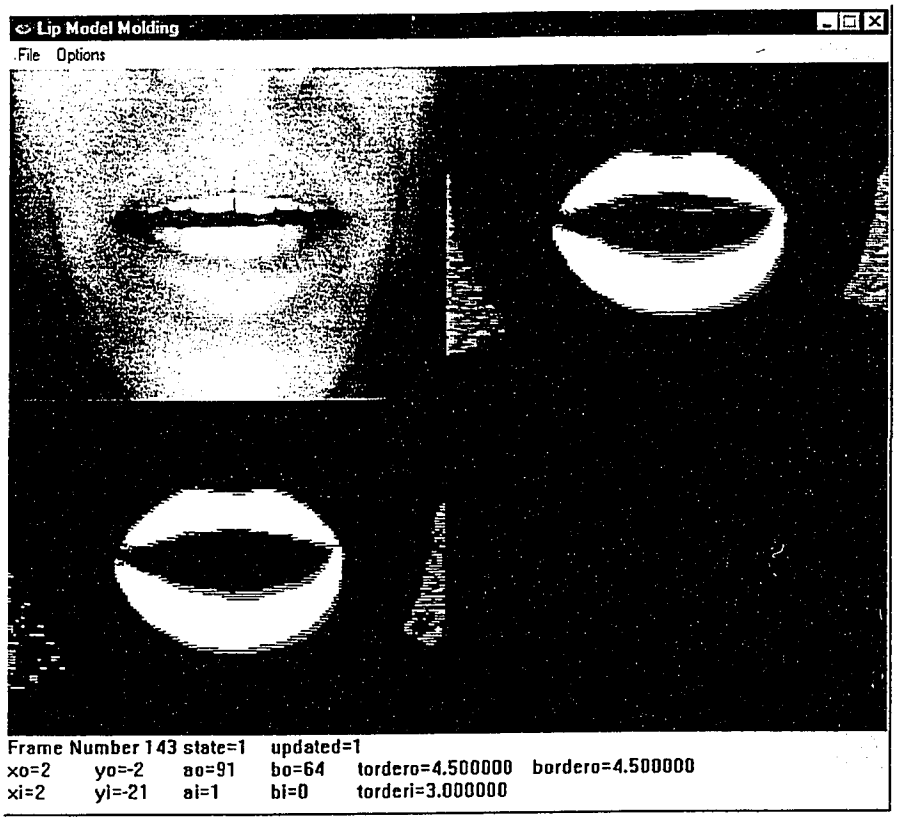

Figure 34 The space between the open lips is connected to the image background.

3.2.3.4 Blob gaps

Gaps in the lip blob may be caused by several reasons. Inappropriate or insufficient lighting conditions can cause shadows to be cast over the corners of the lips. When the image processing is performed on the input image the pixels that are too dark might not be considered to have the blue hue of the lip makeup and might therefore considered background pixels.

Gaps in the lip blob can also be created if the blue makeup marking the lips is not applied uniformly all the way around the lips. To avoid having gaps in the lip blob the color of the makeup must be chosen carefully. A dark blue color can sometimes be confused with black and the hue of black (or any shade of gray) is undetermined and will be assumed to belong to the background of the image.

3.2.3 Parametric model molding

The next section of Figure 23 to be discussed is how the parametric lip model is fitted to the image data. The two binary images are taken as input to a parametric lip contour model-molding algorithm. The external lip contour model is molded to the lip shape determined in the first binary image (the lower left image of Figure 24). The internal lip contour model is molded to the lip shape determined in the second binary image (the lower right of Figure 24). Before getting into the details of how the models are molded and the optimal model parameters are found some of the fine points surrounding the chosen lip model are introduced.

3.2.3.1 The lip model structure

The lips can take many shapes; they are not rigid however they still have a specific structure. To be able to represent the shape of the lips within a computer a method of measuring the shape of the lips is necessary. We need a model that can incorporate structure and allow for variability at the same time. To measure and illustrate the shape of the lips a 2D parametric lip contour model is defined. The lip model is separated in 2 parts the inner contour and the outer contour. The inner contour is based on 2 parabolas after [Lindblom71]. The outer contour consists of 2 parabolas and a cosine function after [Guiard-Marigny96]. Parabolas follow the inner and outer contour of the lips and the cosine function follows the 'dip' at the top of the upper lip. The dimensions of the parametric model used to measure and describe the shape of the lips are shown in Figure 35.

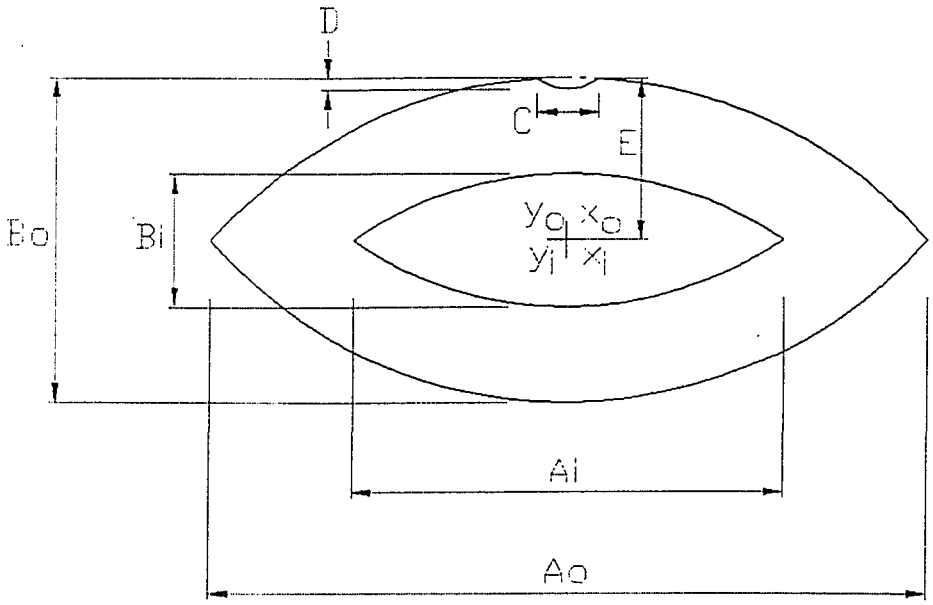

Figure 35 Dimensions of the parametric lip contour model.

The parameters of this model are as follows:

- xo = x value of the origin of the outside parabolas.
- yo = y value of the origin of the outside parabolas.
- xi = x value of the origin of the inside parabolas.
- yi = y value of the origin of the inside parabolas.
- Bo = Outer height
- Bi = Inner height
- Ao = Outer width
- Ai = Inner width
- D = Depth of 'dip'
- C = Width of 'dip'
- E = Internal parameter measuring the offset height of cosine function
- tordero = Top outside parabola order
- bordero = Bottom outside parabola order
- orderi = Inside parabola order (same on both top an bottom).

The parameters tordero, bordero, and orderi appear implicitly in Figure 35. These parameters control the slope of the parabolas. A low order value ($\cong 1$) makes the parabolas more triangular and a high order value ($\cong \infty$) makes the parabolas more square. The parameter E is an internal parameter and is used as an offset value for the

cosine function. Adding E to the cosine insures that there is no discontinuity between the parabolas and the cosine function.

3.2.3.2 Mathematical basis of the model

The Model is based on mathematical equations. The equations for each curve of the parametric lip model are given in the Figure 36. The dimensions that are found in Figure 36 are the parameters that are used in the mathematical equations below.

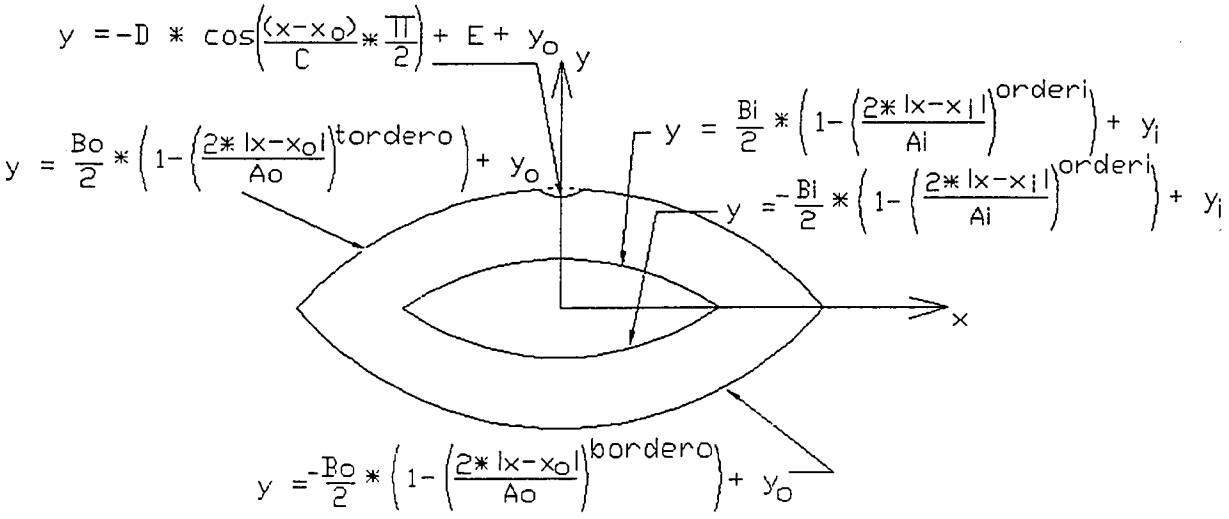

Figure 36 Mathematical equations associated to each part of the parametric lip contour model.

3.2.3.3 How the model can be transformed

The orders of the parabolas are variable. This allows for a better fit between the lip model and the image of the speaker's lip. Having a better fit between the lip model and the image of the speaker's lips in turn gives a better measurement of the height and width of the lips. The height and width of the cosine 'dip', parameters C and D are not changed during the optimization process. They are however adjusted once at the beginning of the training stage on a per individual basis. The center points of the outer and inner contours are independent. Not forcing the center point of the inner parabolas to be the same as the outer parabolas allows for a more precise measurement of the

space between the lips. The orders of the inner parabolas are the same for upper and lower parts. The orders of the outer parabolas are independent and can adjust for differences in the upper and lower lip shape.

3.2.3.4 Open mouth and closed mouth model

Other references have presented 2 different lip models used to model the lips. One for an open mouth another for a closed mouth [Zhang97a]. Here the same model is used for the open mouth and for the closed mouth. If the mouth is closed then the inner height Bi is zero and the inner part of the lip model collapses onto itself.

3.2.3.5 Lip model optimization

Now that the lip model is presented we can get down to the details of how the lip model is fitted to a specific image. The first step in the lip model fitting process is to transform the binary images into potential fields. The next step is to define an energy function. The energy function interacts with the potential field (derived from the binary images) to measure how well the current parameter values fit the image data. Finally a gradient ascent method is used to maximize the energy function. The optimum parameters (the parameters that give the best fit between the parametric model and the image) are the parameters that maximize the energy function.

3.2.3.6 Input data

The images used during the optimization process are the binary images of Figure 24. The outer contour is optimized using the binary image that has the lips isolated (as in the image in the lower left of Figure 24). The inner contour is optimized using the binary image that has the space between the lips isolated (as in the image in the lower right of Figure 24).

The binary images are transformed into potential fields as follows. The pixels that are considered, by the image processing steps described earlier, to be foreground pixels are given a weight of 10. The background pixels are given a weight of -1. A graphical representation of a potential field created by a given binary input image is in Figure 37.

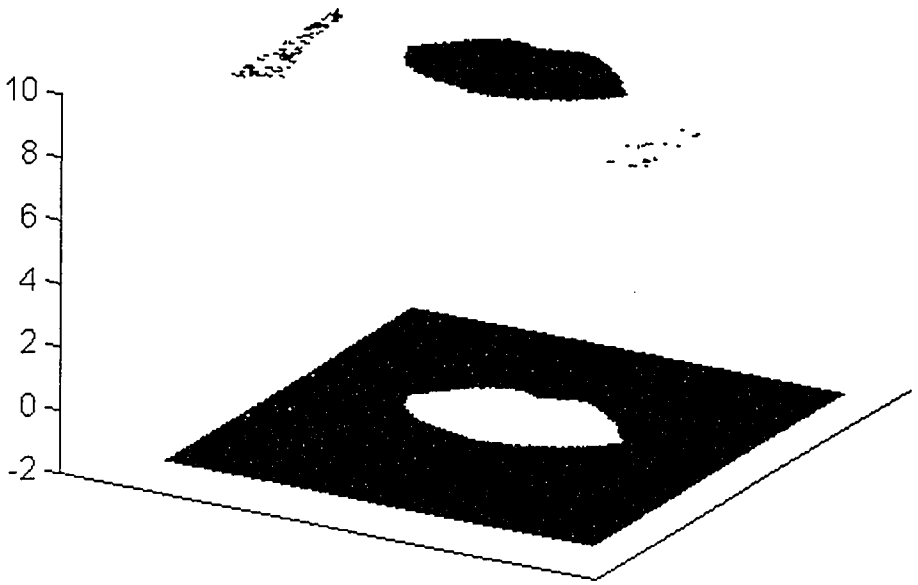

Figure 37 Binary lip image transformed into a potential field.

3.2.3.7 Energy function defined

The energy function, used to optimize the parameters of the parametric lip contour model, simply takes the sum of the values associated to the pixels of the input binary image (the values of the potential field) that are contained within the model.

$$E_{image}(x_o, y_o, a_o, b_o, tordero, bordero, x_i, y_i, a_i, b_i, orderi) = \sum_{Pixels\ inside\ contour} Pixel\ \ Weight \qquad \text{Equation 44}$$

For each foreground pixel that is included within the parametric lip contour model, the energy calculated by the energy function is incremented by 10. For each background pixel that is included within the parametric lip contour model, the energy calculated by the energy function is decremented by 1. The energy function is optimized when it

includes as many foreground pixels as possible and as few background pixels as possible. In other words the parameters of the lip contour model are optimum when the lip contour model follows the edges of the lips in the image.

3.2.3.8 Asymmetric weighting

An asymmetric weighting is used (+10 for foreground pixels –1 for background pixels) to give more importance to the foreground pixels. Experimentation with this system has shown that during the image processing the foreground pixels are more often incorrectly classified as background pixels than the background pixels are incorrectly classified as foreground pixels. It is therefore important that as many foreground pixels as possible be included within the contour of the parametric lip model. Even if that means a few extra background pixels become included. Updating the parameters of the lip contour model to include one extra foreground pixel within the contour of the lip model will increase the value of the energy function even if this parameter update causes up to 9 background pixels to also be included inside the contour of the lip model. Certain lighting conditions can cause shadows to be cast over the corners of the lips causing the corners of the lips to be considered part of the background of the image. The strategy of weighting foreground pixels more heavily than background pixels helps keep the corners of the lips included inside the parametric model. It is important that the pixels at the corners of the lips be included within the model to properly measure the width of the lips.

3.2.3.9 Energy function imaging

A 3-dimensional graphical representation of the shape of the 11-dimensional energy function can be created by calculating the value of the energy function while varying only 2 of the lip model parameters and keeping the values of the other parameters constant.

For example, by choosing to change the centre position of the outer lip model and leaving all other parameters with some constant value we can obtain a 3D graphic showing how the energy function changes as a function of the x_o and y_o parameters. Figure 38 shows the lip model being swept across an input video frame. As the centre point covers all possible pixels of the input image the energy function is evaluated

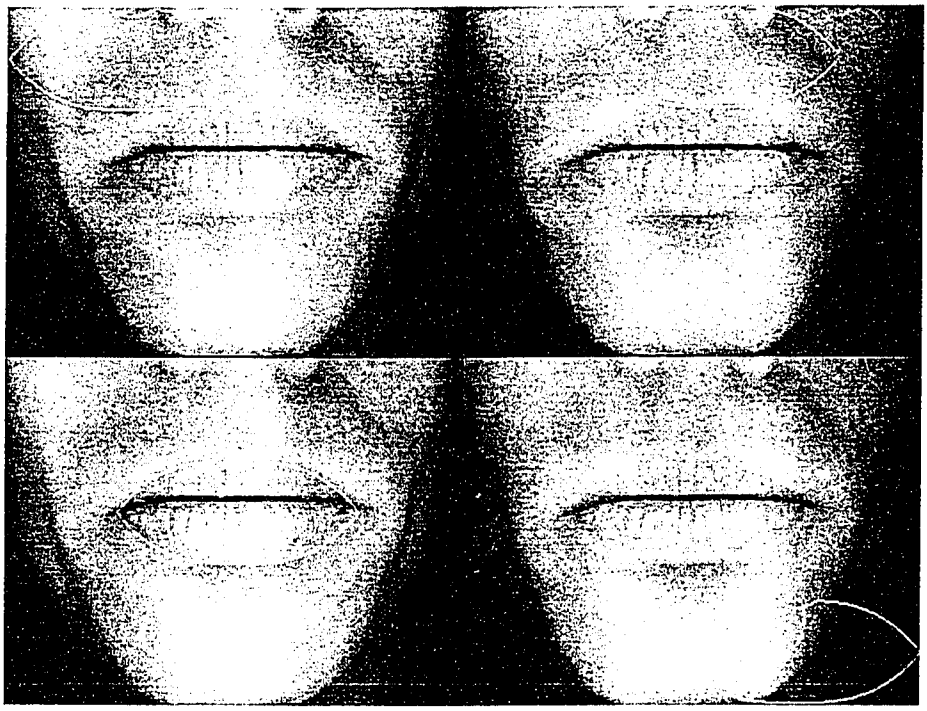

Figure 38 Lip model swept over an input video frame.

creating a function of the form $z=f(x_o,y_o)$. A graphic of the values of the energy function calculated at each centre point is contained in Figure 39. The energy function contains only one maximum point. The maximum point of the energy function is found when the parametric lip model is placed over the lips so a maximum number of foreground pixels and a minimum number of background pixels are contained within the contour of the parametric lip model. The optimal position of the parametric lip contour model is shown in the lower left image of Figure 38. The shape and peak value of the energy function graph depends not only on the values of x_o and y_o but it also depends on the constant values chosen for the other parameters of the lip model. For example changing the choice of constant values for the height or width of the lip model (parameters Ao or Bo)

will create an energy function graph that will have a different shape and peak value than the graph of Figure 39. The next section describes the gradient ascent method that is used to optimize the shape of the lips.

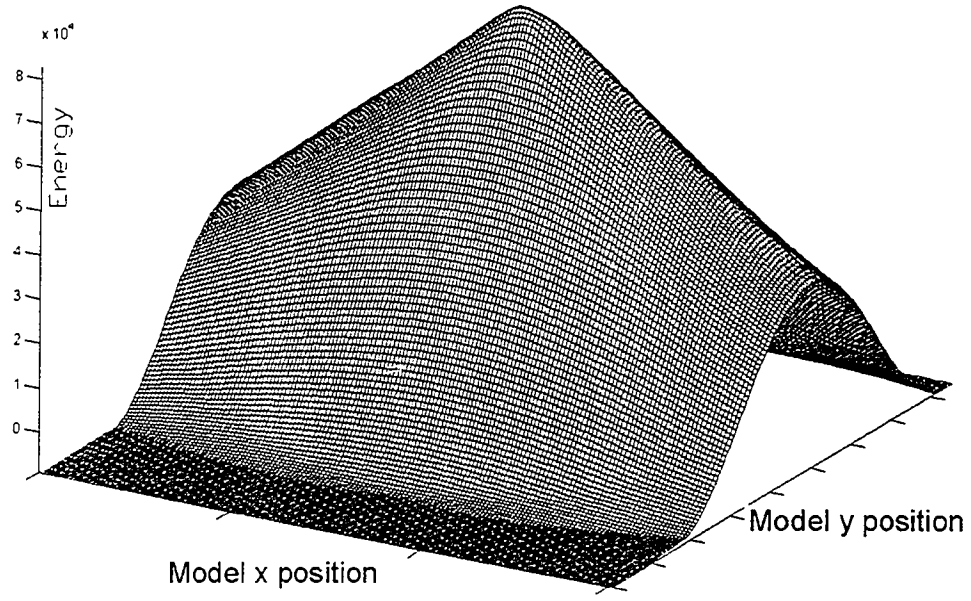

Figure 39 Energy function value as a function of lip model x_o-y_o position.

3.2.3.10 Gradient ascent

The gradient ascent method is used to find the maximum value of a function as well as the point in the parameter space where the energy function is maximised $\bar{X}_o$. The algorithm begins at some point $\bar{X}$ in the parameter space and steps in the direction of steepest ascent. We start by defining the starting point $\bar{X}$ as the following.

$$\bar{X} = (x_o \quad y_o \quad a_o \quad b_o \quad tordero \quad bordero \quad x_i \quad y_i \quad a_i \quad b_i \quad orderi) \qquad \text{Equation 45}$$

The direction of steepest ascent is given by the gradient of the energy function. The elements of the gradient vector are the partial derivatives of the energy function with

respect to each of the elements of the parameter vector. The gradient vector of the energy function is given by

$$\bar{\nabla}E_{image}(\bar{X})=\left(\frac{\partial E_{image}}{\partial x_o}\ \frac{\partial E_{image}}{\partial y_o}\ \frac{\partial E_{image}}{\partial a_o}\ \frac{\partial E_{image}}{\partial b_o}\ \frac{\partial E_{image}}{\partial tordero}\ \frac{\partial E_{image}}{\partial bordero}\ \frac{\partial E_{image}}{\partial x_i}\ \frac{\partial E_{image}}{\partial y_i}\ \frac{\partial E_{image}}{\partial a_i}\ \frac{\partial E_{image}}{\partial b_i}\ \frac{\partial E_{image}}{\partial orderi}\right)$$

Equation 46

In this implementation each element of the gradient vector is approximated using a difference equation. For example the first two elements of the gradient vector are:

$$\frac{\partial E_{image}}{\partial x_o}=\frac{E_{image}(x_o,\ldots)-E_{image}(x_o+\Delta,\ldots)}{\Delta}$$

Equation 47

$$\frac{\partial E_{image}}{\partial y_o}=\frac{E_{image}(y_o,\ldots)-E_{image}(y_o+\Delta,\ldots)}{\Delta}$$

Equation 48

The difference equation uses Δ=1 for the integer variables. The lip model parameters x_o y_o x_i y_i A_o B_o A_i B_i are all integers and therefore the smallest increment possible is an increment of 1. The parameters of the order of the parabolas are floating point values. For these parameters (tordero, bordero and orderi) an increment of Δ=1 is too large because it does not allow for fine adjustments to be made to these parameters. Too small a value of Δ is not good either. If the value of Δ is too small than the derivative calculated using the difference equation will always produce a value of zero because of the rounding that occurs within the energy function. The value used in this implementation is Δ=0.5.

Now that the starting point as well as the direction of steepest ascent is defined the next step is to determine how far along the direction of steepest ascent to go. An equation is defined that calculates the energy for a new point in the parameter space that is found by starting at the original point $\bar{X}$ and moving along the direction of steepest ascent given by $\bar{\nabla}E_{image}(\bar{X})$. This equation is as follows

$$Z(t) = E_{image}(\bar{X} + t\bar{\nabla}E_{image}(\bar{X}))$$ Equation 49

How far to go in the direction of steepest ascent is controlled by the value of the parameter t. The value of t that maximises Z(t) must be found. Experimentation has found that incrementing t over the interval from 1 to 10 give good results. The optimum value of t is given when Z(t) is maximum over the given interval. Once the optimum value of t is found the value of $\bar{X}$ is updated to the new approximation of $\bar{X}_o$.

$$\bar{X} = \bar{X} + t\bar{\nabla}E_{image}(\bar{X})$$ Equation 50

The optimization procedure restarts by calculating Equations 46 49 and 50 iteratively, each time adjusting the value of $\bar{X}$ closer and closer to the maximum point $\bar{X}_o$. The algorithm stops when adjusting any of the parameters cannot increase the value of the energy function. At that time the value of the energy function is maximised and $\bar{X}$ is taken as the best approximation of $\bar{X}_o$.

3.2.3.11 Model initialization

Care must be taken when initializing the lip model. The open lips are a ring of foreground pixels. There are background pixels both outside as well as inside the lips. If the lip model is initialised too small or too high or too low the model might collapse onto itself or onto only half of the lips (see Figure 40).

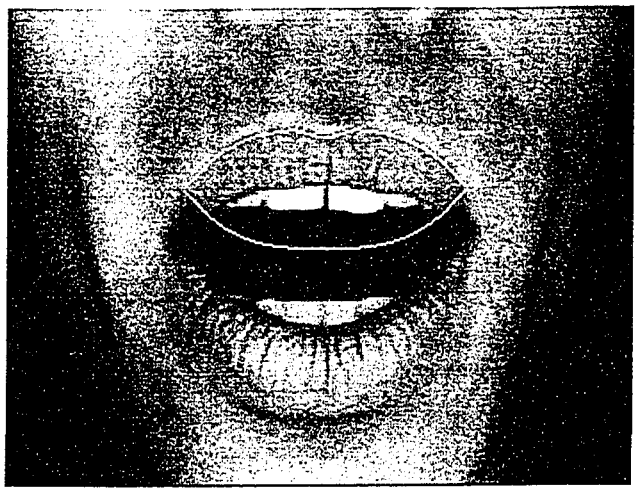

Figure 40 Incorrect lip model initialization.

In general the first few frames of the reference video have the mouth closed, and the lip model is initialised with parameters that make the model wide, tall, and centred. There is therefore no danger of missing the lips. At the chosen frame rate (30 frames per second) there are not very large differences in lip position and shape from one frame to the next. To initialise the lip model for the next frame, the model's centre point is chosen to be the same as in the previous frame and the model is set a little wider and a little taller. These precautions insure that the model does not fall into the local maximum found in Figure 40.

3.2.3.12 Lip height/width and centre point

To avoid local maximums during the optimization process not all parameters are updated at the same time. The parameters are not independent one from the other and the model can get caught in a local maximum of the energy function. For example if the model is not centred over the lips and the height and width are updated at the same time the centre point is updated the model might collapse on itself and miss the lips completely.

Figure 41 Updating the height and the center point at the same time.

When the gradient of the energy function is calculated in the first image on the left of Figure 41 the vector of steepest ascent required that the height of lip model be reduced as the model is moved up. Reducing the height of the model increased the value of the energy function by reducing the number of negatively weighted background pixels that are included within the model. Moving the model up increases the number of positively weighted foreground pixels contained within the model. The reduction in height of the model is more beneficial (created a higher increase in the energy function) than moving the model upwards. Therefore the model will reduce in height faster than it is moved up. This can cause the model to miss the lips completely and the height of the model is reduced to zero.

3.2.3.13 Lip height/width and parabola order

Other local maximum traps can occur if parameters are updated at the same time. If the order of the parabolas are updated before the height and width of the model is found, the model can get caught in a local maximum as is illustrated in Figure 42. The values of the width and height of the model are not large enough nor are the parameters tordero and bordero (the order of the top and bottom outer parabolas). In this particular case, increasing the order parameters increases the energy function more than increasing the height and width of the model. The order parameters are therefore increased faster than the height and width of the model causing the lip model to get caught in a local maximum.

Figure 42 Updating the height, width, and parabola order at the same time.

To avoid the local maximums, the parameters are not optimized all at the same time. Only small groups of parameters are updated at one time.

3.2.3.14 Direction limiting vector

Figures 40, 41 and 42 all show how it is important that the model be correctly initialized. It is important that the model be centered (or nearly centered) before the height and width parameters are updated. It is also important that the order parameters (tordero, bordero and orderi) be updated only when the height and width of the model is nearly optimum.

To limit the number of lip model parameters that are updated at once a direction-limiting vector multiplies the gradient vector before the gradient vector is used to find the direction of steepest ascent.

The values of the components of the direction-limiting vector are either 1 or 0. When the component of the direction limiting vector for a certain parameter of the parametric lip contour model is zero the contribution that that parameter has on the direction of steepest ascent is blocked.

$$SteepestAscentVector = GradientVector \bullet DirectionLimitingVector$$

$$\bar{S}_a = \nabla \bar{E}_{image} \bar{D}_l \qquad \text{Equation 51}$$

$$\bar{S}_a = \begin{pmatrix} \frac{\partial E_{image}}{\partial x_o} & \frac{\partial E_{image}}{\partial y_o} & \cdots & \frac{\partial E_{image}}{\partial orderi} \end{pmatrix} \begin{pmatrix} D_l 1 \\ D_l 2 \\ \vdots \\ D_l 11 \end{pmatrix}$$

3.2.3.15 Optimization stages

The optimization process is cut into 15 stages. Each stage optimizes a few parameters at a time. The first stage for example optimizes the center point of the model (both x_o and y_o) and the second stage optimizes the height and width of the model. Table 2 shows the value of the direction limiting vector for each of the 15 stages.

Parameter Name / Step #	xo	yo	ao	bo	tordero	bordero	xi	yi	ai	bi	orderi
0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0
4	If the parameter values have changed since step 0, go back to step 0. Else go on to step 5.										
5	0	0	0	0	1	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0
7	If the parameter values have changed since step 0, go back to step 0. Else go on to step 8.										
8	0	0	0	0	0	0	1	1	0	0	0
9	0	0	0	0	0	0	0	0	1	1	0
10	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	1	0
12	If the parameter values have changed since step 8, go back to step 8. Else go on to step 13.										
13	0	0	0	0	0	0	0	0	0	0	1
14	If the parameter values have changed since step 8, go back to step 8. Else the model parameters are optimal.										

Table 2 Direction limiting vector values for the 15 optimization stages.

3.2.3.16 Speed vs accuracy

The way the optimization stages presented in the previous table are organised is not unique. There are many other ways to decide which and when parameters are

optimised. Surely the number of stages can be reduced to increase processing speed. However, reducing the number of stages increases the risk that the model gets caught in a local maximum. Because the image analysis is done off-line during the training stage the choice was made to err on the side of caution and include perhaps a few extra stages (slowing down processing) to insure that the model does not get caught in local maximums.

3.2.3.17 Control stages

From table 2 we see that stages 4, 7, 12, and 14 are control stages. In a control stage the parameters are not updated but rather a choice about weather to restart certain stages is made. If any model parameters have changed since the last restart then the parameters updated in the previous steps may no longer be optimum. In other words, each time one parameter is changed all other parameters have to be checked to see if they are still optimum. If the algorithm passes from stage 0 to stage 14 without updating any of the parameters then the current parameters are assumed to give the best fit between the lip model and the lips in the image.

3.2.3.18 Immunity to noise

One interesting property of this type of image processing is that the lip model is (to a certain extent) immune to noise. The lip model is structured; it cannot follow just any outline. Adjusting one or more of the lip model parameters is the only way of changing its shape. This built in structure allows the model to be immune to noise. Figure 43 contains a screen shot of the program AVEXTRACT.exe. This program moulds the lip model to fit the image of the current video frame. In this case a noisy video frame of the lips is given as input. The program was able to see through the noise and capture the shape of the lips.

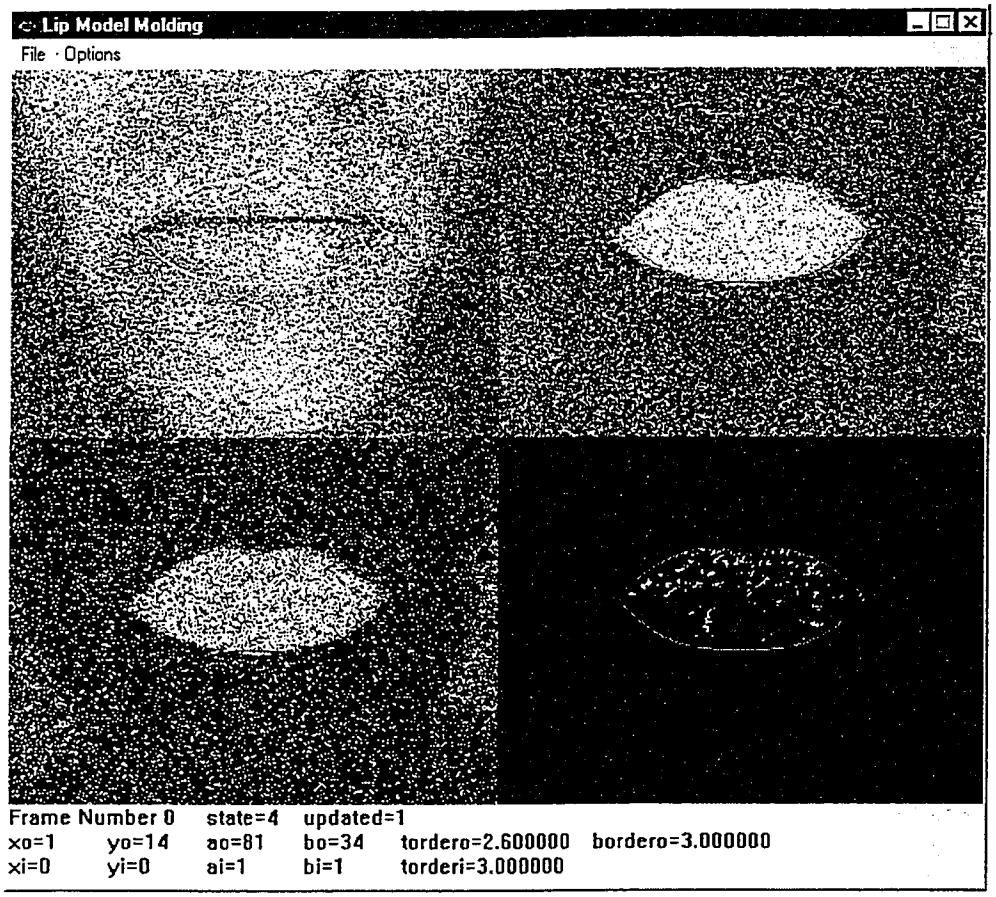

Figure 43 Lip model molding program can see through noise

3.3 References

[McKerrow91], [Berns96], [Smith96], [Abbs73], [Perkell92], [Hani98], [Lallouache91], [Chan99], [Li95], [Dai96], [Adjoudani97], [Deng00], [Basu98], [Lindblom71], [Guiard-Marigny96], [Zhang97a], [Yuille92], [Hennecke94], [Coughlan00], [Escolano97], [Esme96], [Figuiredo97], [Hennecke94], [Jain98], [Jain96], [Jyh-Yuan97], [Kober94], [Liu99], [Mirhosseini98], [Ngan96], [Rabi97], [Rao94], [Rao95], [Saji97], [Sakalli98], [Sungyun98], [Tawfik99], [Wang00], [Yuille92], [Xie94], [Xu98], [Zhang97a], [Axencott97], [Fabian97], [Tsallis96], [Bresinski99], [Franke00], [Smith96], [Swensen98].

CHAPTER 4 – Audio to video mapping

4.1 Audio video time alignment

The video frame rate is well below the audio sampling rate but a one to one comparison between the two media is necessary. The solution is to cut the audio stream into frames in order to facilitate a one-to-one comparison. This section will explain the choice of length and offset for the audio frames as well as how interpolation and averaging is used to time align the frames and allow a one-to-one comparison between the two media.

4.1.1 Audio and video frame rates

The video frame rate chosen for this project is 30 frames/sec. This frame rate corresponds to a video frame length of 1/30sec ≈ 33ms. The audio sampling rate is set to 11025 samples/sec. Typical values for an LPC analysis system as described in [Rabiner93] define the audio frame length as 30ms to 45ms with a frame to frame offset of 10ms to 15ms. To align the audio frames with the video frames an audio frame length of 1/30sec ≈ 33ms (corresponding to approximately 366 audio samples) was chosen. The offset from frame to frame was chosen as 1/3 the frame length. This length corresponds to 122 samples or approximately 11ms. Figure 44 shows the audio and video streams as a function of time. The audio stream is shown with overlapping hamming windows, numbered 0 to 12, superimposed over the audio stream.

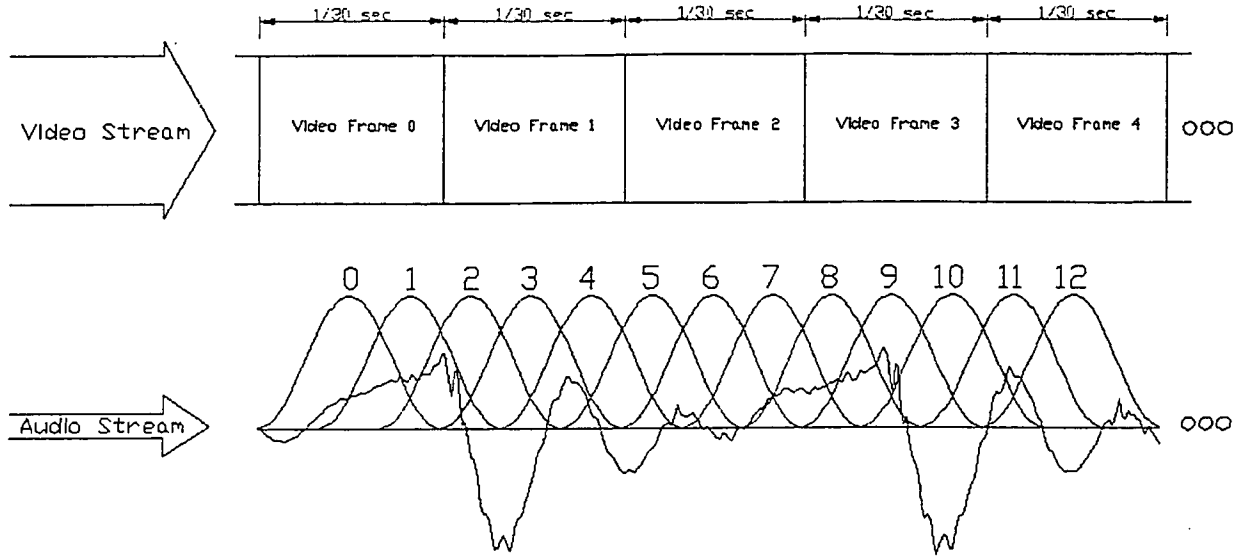

Figure 44 Video stream time aligned with audio stream under hamming windows.

To clearly show the timing relations (audio frame length as well as offset between audio frames) between the audio and video frames a second figure, Figure 45 shows the unique audio stream copied 3 times so the overlapping frames can separated one from another.

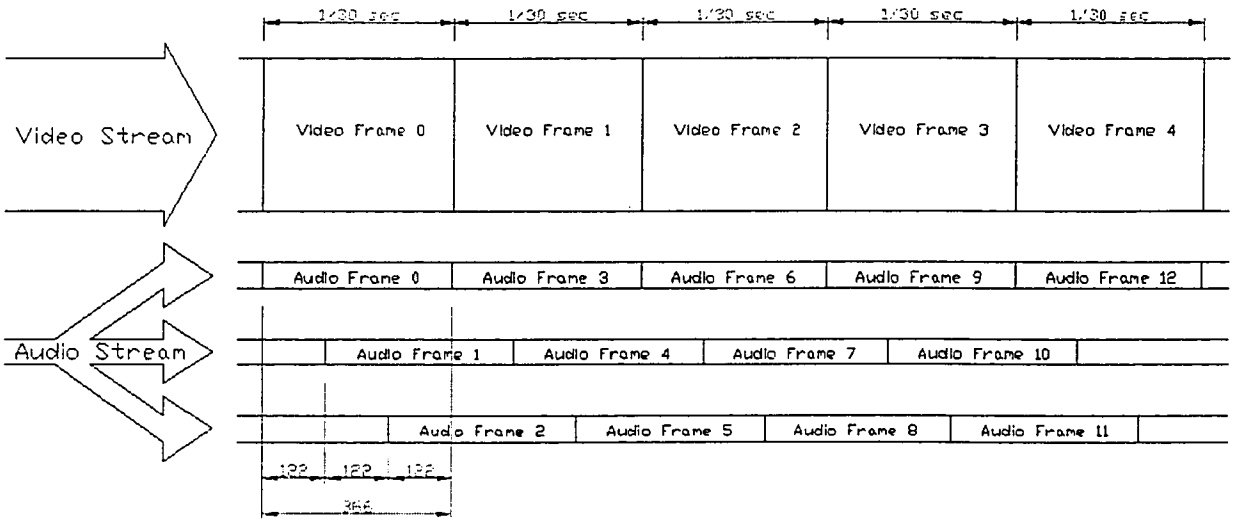

Figure 45 Video stream time aligned with audio stream copied 3 times.

Figure 45 shows an audio frame length of 366 samples and adjacent audio frames separated by 122 samples. This way every third audio frame lines up exactly with a video frame. The next sections describe how interpolation is used to create a set of lip model parameters for each audio frame.

4.1.2 Video parameter interpolation (training stage)

In the training stage a mapping between the qualities of the audio frames and the lip shape parameters of the video frames is created. It is therefore necessary to be able to compare the audio frames with the video frames on a one-to-one basis. The audio frame rate is three times the video frame rate. To make a one-to-one comparison between the voice properties, extracted from the audio frames, and the lip model values, extracted from the video frames, the lip model values found from the images in each of the video frames are linearly interpolated. Figure 46 shows the overlapping audio frames arranged end to end and how the values of the lip model parameters are calculated for the missing video frames.

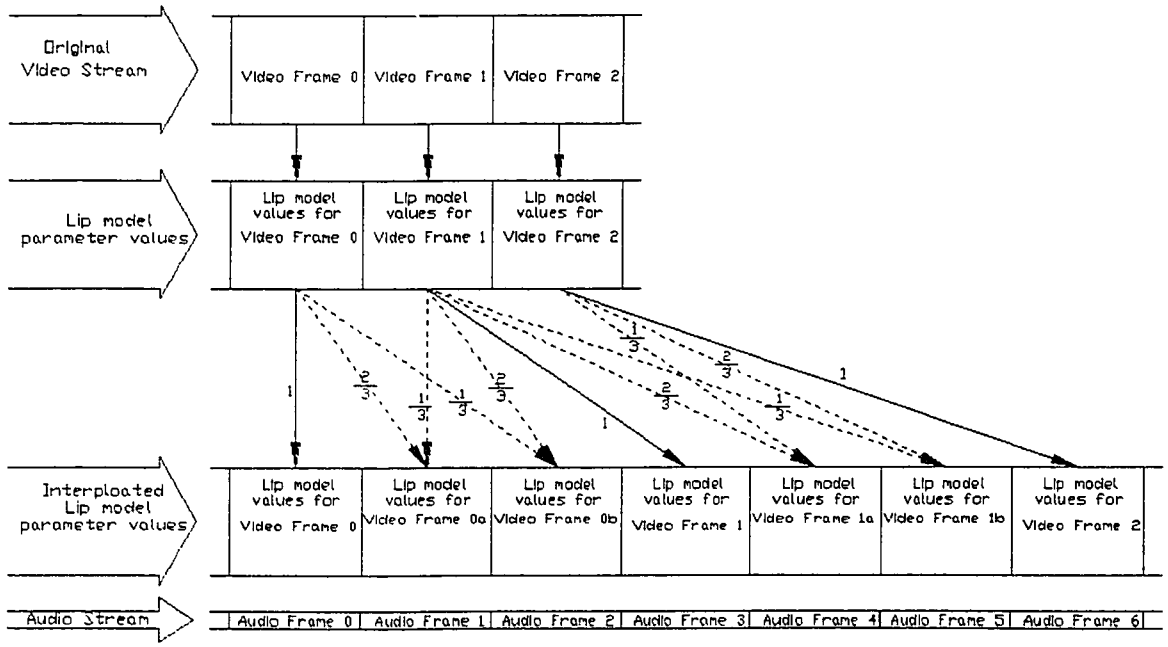

Figure 46 How the video parameters are interpolated.

The lip model parameters that will be associated with audio frame 1 for example are calculated by taking 2/3 the parameter values found in the video frame 0 and adding to that 1/3 the parameter values found in video frame 1. The weights 1/3 and 2/3 were chosen because their sum is 1 (to avoid scaling the values for the missing frames) and because the chosen interpolation function is linear. The mapping relating the properties of the audio signal to the parameters of the lip model can now be created with time aligned data.

4.1.3 Video parameter averaging (animation stage)

In the animation stage each audio frame is analyzed and the results of the analysis are used to estimate the parameters of the lip model. Because the audio frame rate is 3 times the video frame rate, the rate the lip model parameters are revealed will also be 3 times the video frame rate. Therefore a weighted average of the lip model parameters is calculated to reduce the lip model parameter rate so the animated lip model will be rendered synchronized with the audio signal. A weighted average is used because some audio frames are aligned with the video frames while others are shifted by 1/3 the audio frame length to the left or to the right (see Figure 45). The lip model parameters calculated from the audio frames that are not squarely aligned with a video frame are weighted 1/3 less than the lip model parameters calculated from audio frames that are squarely aligned with a video frame. For example audio frames 2 and 4 (see Figure 45) are not squarely aligned with any video frame. They are however only shifted by 1/3 their length to the right or left of video frame 1. Audio frame 3 is time aligned with video frame 1. Therefore when performing the averaging procedure the video parameters found from the analysis of audio frame 3 are weighted by 3/7 while the video parameters found from the analysis of frames 2 and 4 are weighted by 1/3 less than frame 3. (3/7 – (3/7*1/3) =2/7). These specific values were chosen because when performing the averaging the sum of the weights should be 1 to avoid scaling the average. Figure 47 shows how the weighted average is calculated.

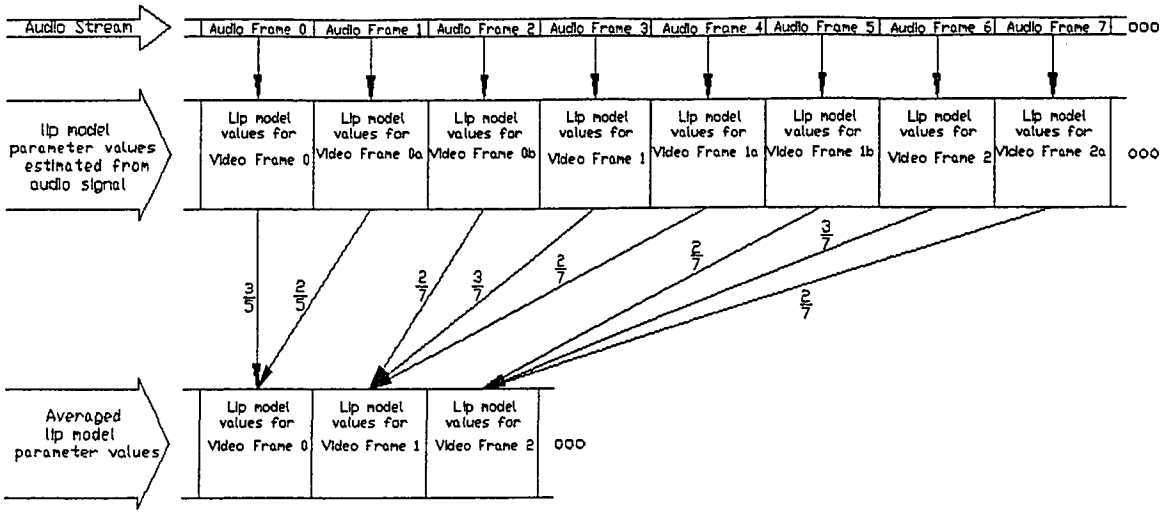

Figure 47 How the average of the lip model parameters is calculated.

4.2 Building the mapping

Human speech from the point of view of an acoustic signal and also from the point of view of lip shape is highly variable. When uttering a given speech sound several times, each time the frequency spectrum of the speech sound will be slightly different and the lip shape will vary slightly. A mechanism that can learn the *typical* lip shapes for *similar* sounds is required to effectively solve the problem of driving a lip shape model from the voice signal. The system implemented in this project processes the speech sound from a reference audio-video file with a vector quantization algorithm to group similar voice sounds together. Each voice sound from the reference audio-video file has a lip shape associated with it. The lip shapes associated to of each of the members of the small groups of speech sounds created by the vector quantization algorithm are averaged. This average lip shape is chosen as the output lip shape for the whole group of similar speech sounds. Using this method the variability of the speech frequency spectrum is accounted for by grouping similar voice sounds together. Also the variability of the lip shape for these similar sounds is accounted for by averaging these lip shapes. This method not only accounts for variability within the reference video but it also compresses the data by creating a compact mapping.

A block diagram describing the steps required to build the audio parameter to lip shape parameter mapping is contained within the following figure.

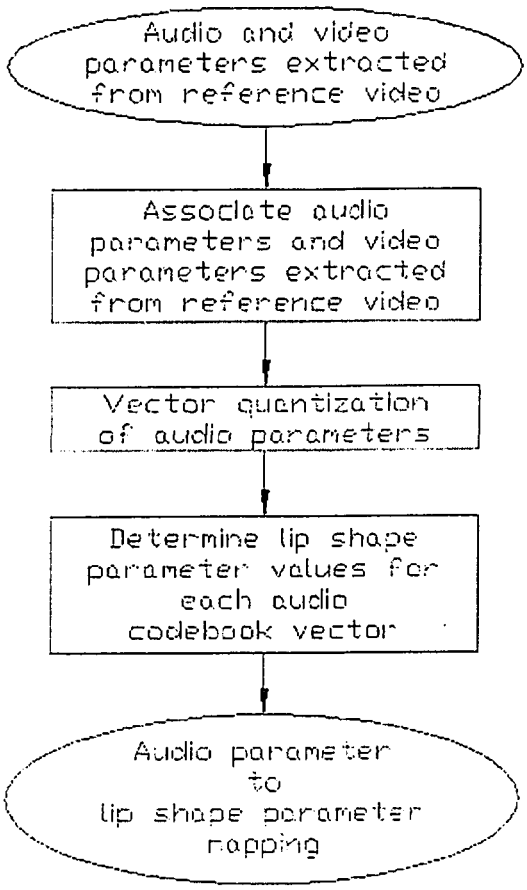

Figure 48 Building an audio to video parameter mapping from the reference video.

The input data to Figure 48 is the audio and video parameters extracted from the reference audio video file. The output data of Figure 48 is the audio parameter to lip shape parameter mapping. The following sections will describe each of the tasks outlined by the boxes of Figure 48.

4.2.1 Audio parameter and video parameter association

The audio/video file recorded for the training stage is analyzed to extract both the audio and lip shape parameters. The lip shape parameters and audio parameters are time aligned as in section 4.1.2. The audio parameters and lip model parameters are associated one to the other by assigning the same index concurrently to both sets of time aligned parameters. Figure 49 illustrates how the sets of cepstral coefficients are

associated with time aligned sets of lip model parameters. What has been created is a set of reference terms showing that *when the speaker produces this sound his/her lips make this shape.*

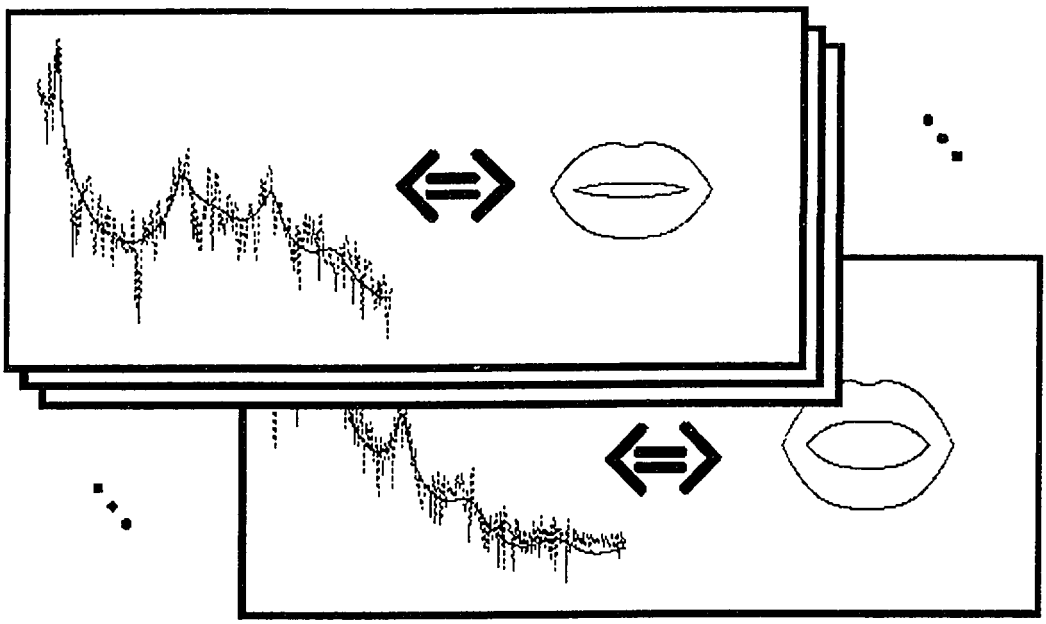

Figure 49 Time aligned audio parameters associated to lip shape parameters

4.2.2 Lip model parameters used in the mapping

Not all of the parameters of the parametric lip model are used in the mapping. The only parameters of the lip model that are estimated from the voice stream are the outer width A_o and the outer height B_o parameters. Values of the inner contour of the lip model can be inferred using the values of the outer contour of the lips (by giving the lips some constant thickness). Therefore, estimating the inner contour values from the audio signal would be redundant. The order parameters (tordero, borero, orderi) are highly variable and difficult to estimate from the voice. A speaker dependant constant value is chosen for the animation of the lip model. However, the order parameters are important in the training stage because they allow for precise measurement of the model width and height. Figure 50 shows the lip model parameters Ao and Bo that are the only parameters of the lip model predicted from the audio signal.

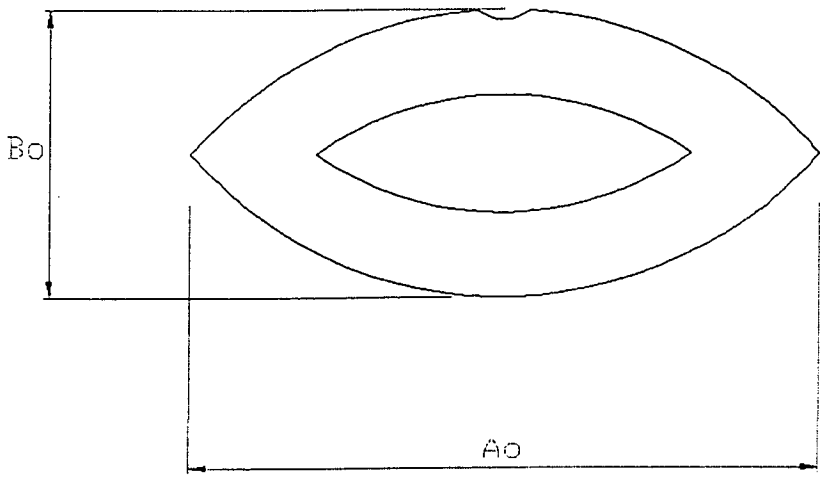

Figure 50 Lip model parameters used in the audio parameter to lip shape parameter mapping.

4.2.3 Vector quantization

Vector quantization is a method of compressing data. It has been used to compress voice, image and other data [Gray00]. The vector quantization method builds a codebook of vectors that are referred to by their index. Placing a codebook containing a list of codebook vectors on both ends of a communication channel, for example, allows a series of vectors to be transmitted from one end of the channel to the other by simply transmitting the index of the series of vectors. Transmitting only the index can significantly reduce the amount of data to transmit. A codebook containing 1024 vectors can refer to a vector using only 10 bits. One vector can contain many elements quantized to a given precision. For example if the codebook vectors each have 20 components and these are quantized to 16 bits each one single vector will require (20*16 =) 320 bits. When you compare the vector quantization method that uses only 10 bits per vector versus the transmission of the entire vector that requires 320 bit the savings are significant.

The vector quantization algorithm is used here not only as a method of compressing the data required to store the audio to lip shape mapping but also because of the grouping (averaging) effect the algorithm has. By grouping similar sounds together the typical lip

shapes for a small region of the cepstral space can be calculated, thereby seeing through the natural variation that exists in the audio/video reference file recorded in the training stage.

The codebook is built by distributing the codebook vectors over the interval spanned by the input data set in such a way as to minimize the distortion between each input vector and the nearest codebook vector (quantization error).

4.2.3.1 Distance measure

To be able to minimize the distortion (quantization error) between the codebook vectors that represents the input vectors and the input vectors themselves a distance measure must be defined. In this implementation the vectors that are quantized using the vector quantization method are the weighted cepstral coefficients obtained from the audio analysis. The weighted cepstral distance measure is used to determine how similar or dissimilar two given inputs are. It is simply the Euclidean distance squared between two weighted cepstral vectors and is defined as the following (copied here from section 2.1.5.1 for the reader's convenience).

$$D_{a,b} = \sum_{i=1}^{Q} (\hat{c}_{a,i} - \hat{c}_{b,i})^2$$

4.2.3.2 Algorithm description

To minimize the distortion created by quantizing the input vectors into the codebook vectors the codebook is built in stages. The algorithm begins by placing one codebook vector in the centroid of the input data set creating a 1-vector codebook. Then that one codebook vector is split into 2 and the positions of each of the codebook vectors are adjusted until the quantization error is minimized. Then the 2 codebook vectors are split into 4, and so on until the desired codebook vector size is achieved. Figure 51 contains a flow chart of this vector quantization algorithm.

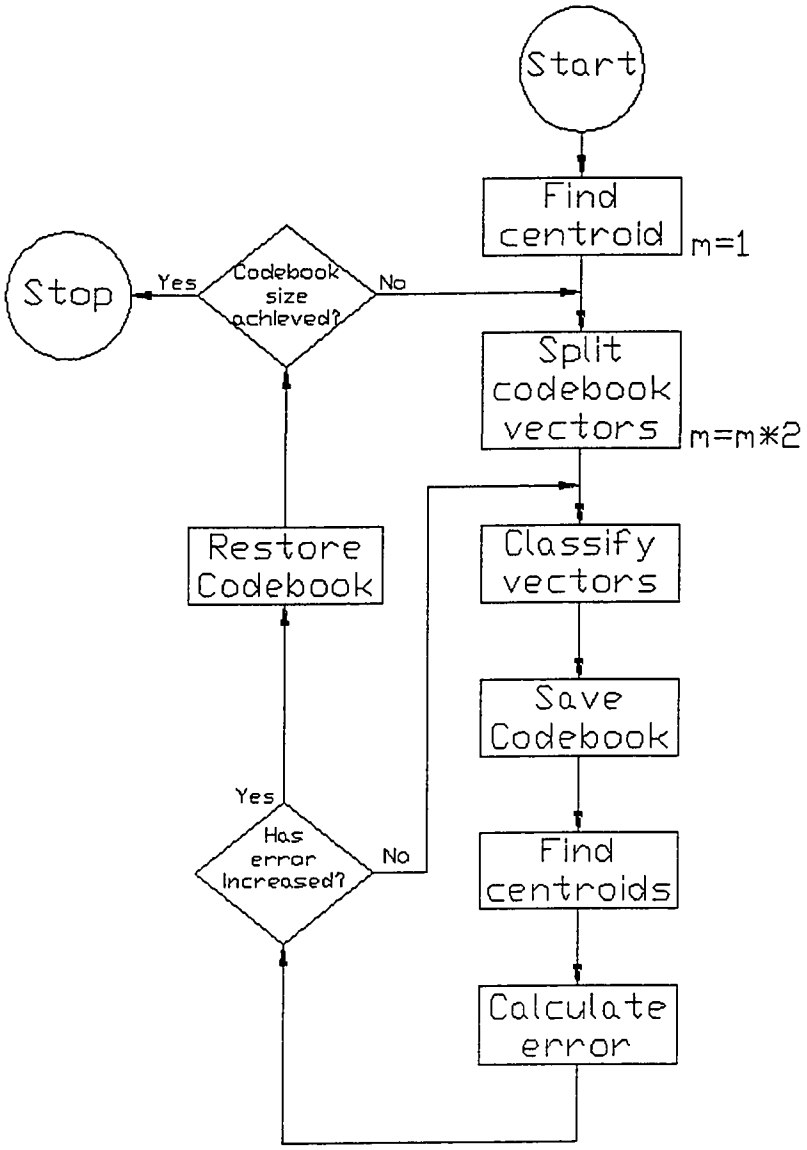

Figure 51 Binary split vector quantization algorithm adapted from [Rabiner93]

The next 2 figures show an example of the application of this vector quantization algorithm on a simulated 2D set of input vectors. The '+'s and 'x's are input vectors while the 'o's are the codebook vectors. The lines on the graphs show the classification borders. The input vectors contained within a given region are nearest to the codebook vector found within that region. The '+' change to 'x' and vice versa to help in the understanding of the classification of the input vectors. The codebook vectors are updated to the centroid of their input data region until such an update ceases to reduce the total distortion. At that point the codebook vectors are split and the new vectors are

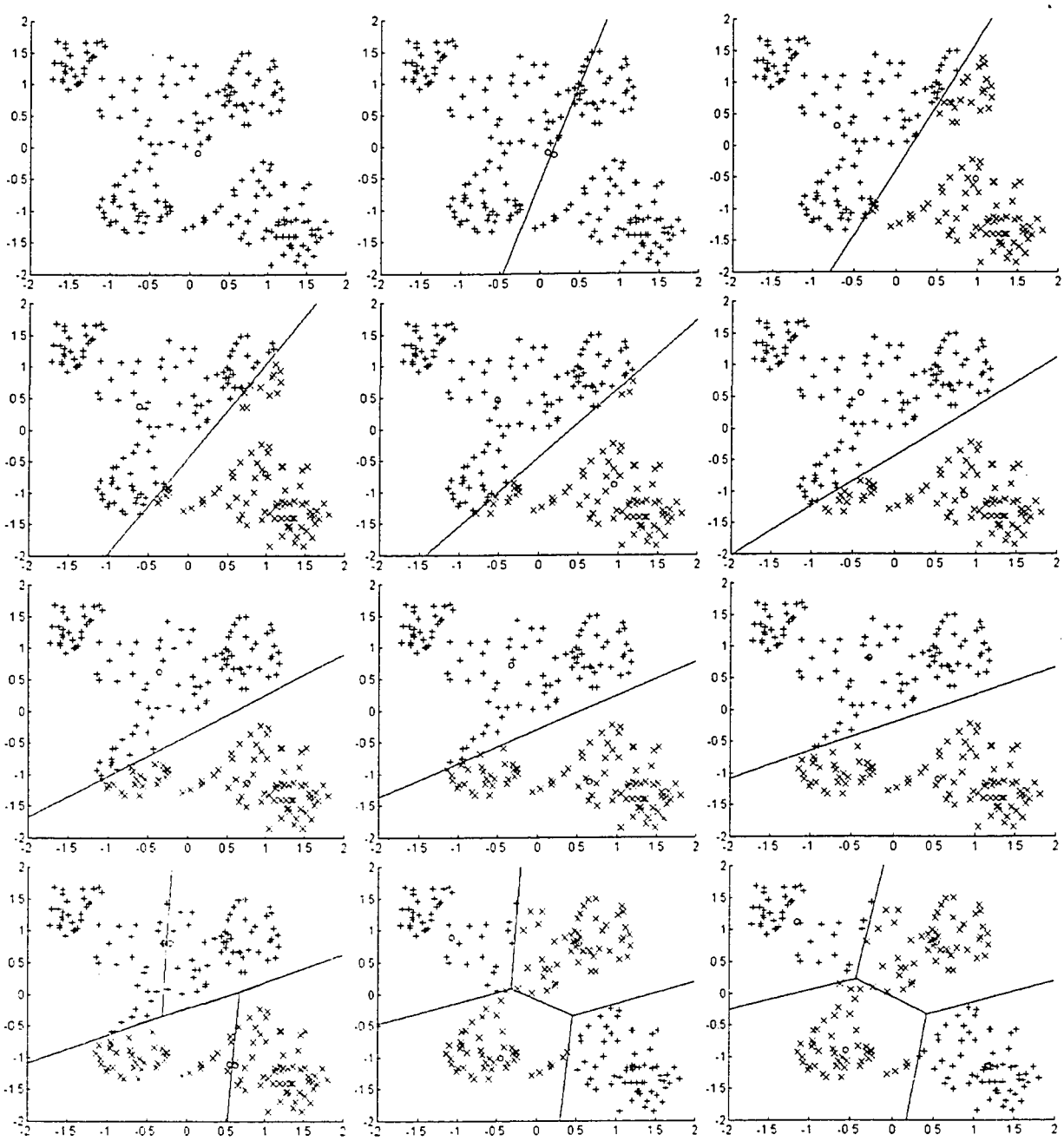

Figure 52 2D vector quantization example (continued in next figure)

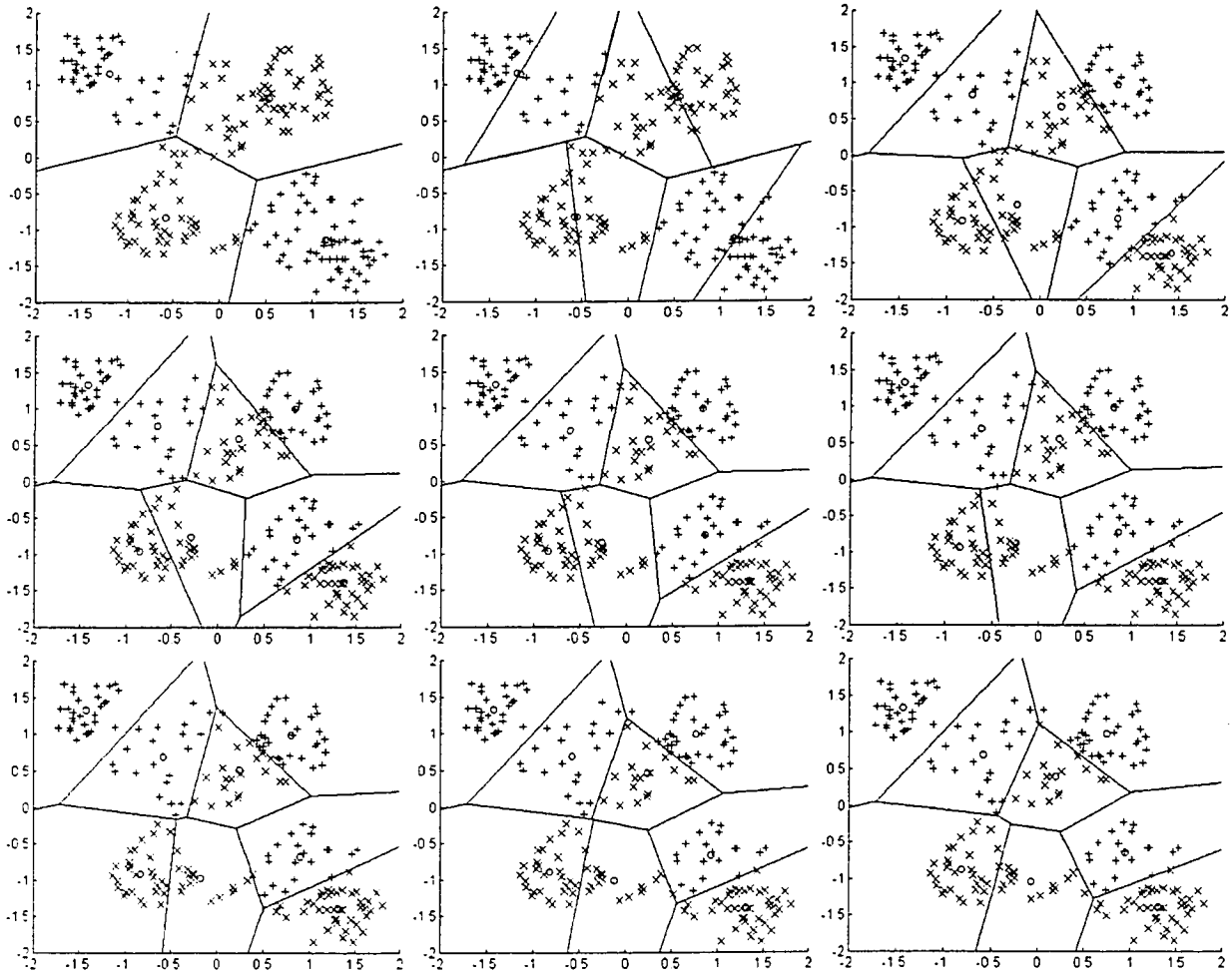

Figure 53 2D vector quantization example (continued from next figure)

updated again. A new graph is drawn each time the vector quantization algorithm either splits the vectors or updates the position of the centroid.

4.2.3.3 Quantization error

The total quantization error is defined as the sum of the distance squared between each input vector and the nearest codebook vector (see section 4.2.3.1). As the number of codebook vectors increases the quantization error decreases. The arrangement of the codebook vectors can also greatly affect the value of the total quantization error. The numbers 1 to 21 on the horizontal axis of Figure 54 illustrate the total quantization error for each of the 21 graphs in Figures 52 and 53.

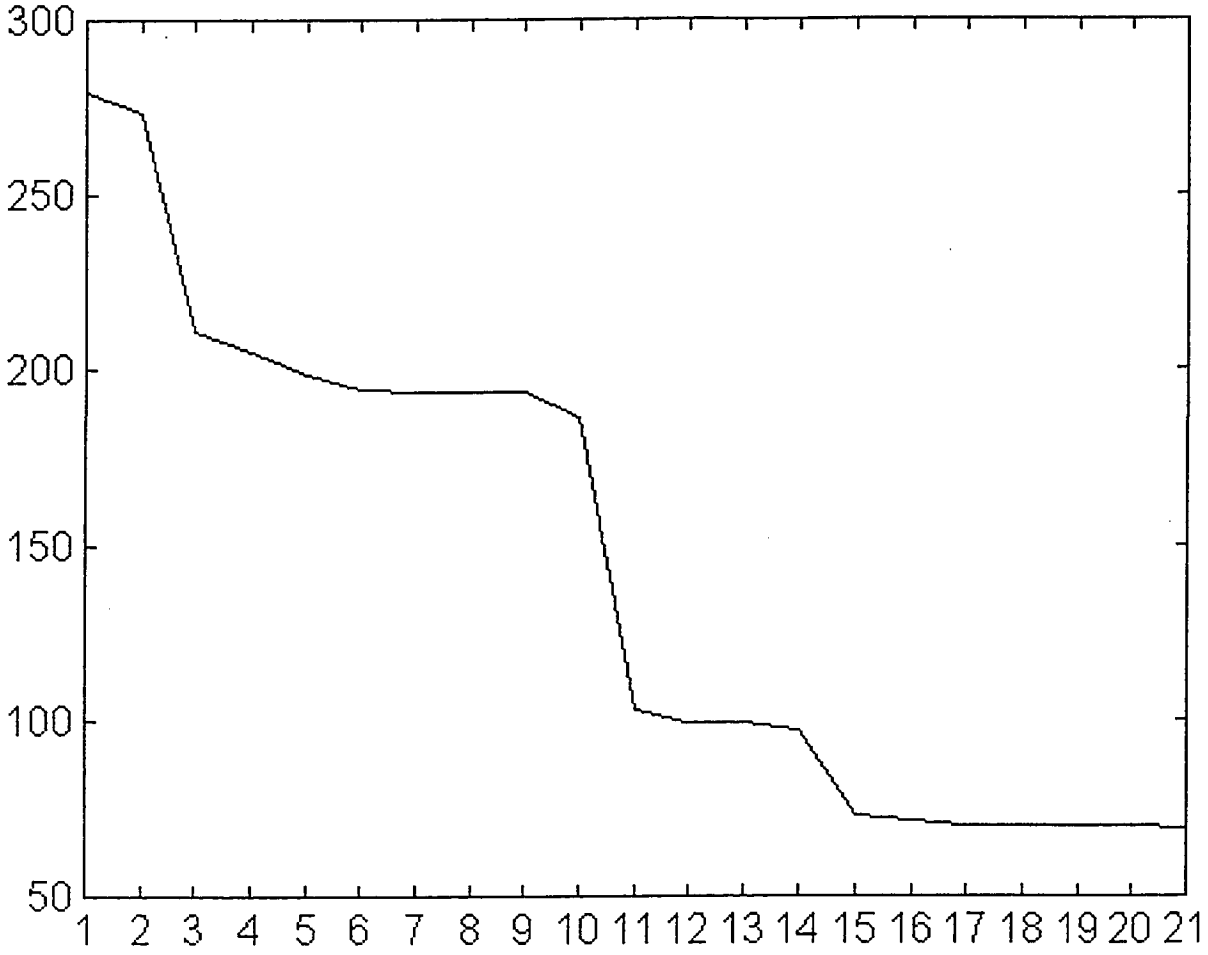

Figure 54 Quantization error for the example data

As the algorithm moved the newly split codebook vectors to the centroids of the newly created regions the quantization error dropped significantly (as in step 2 to 3, 10 to 11 and 14 to 15, in Figure 54). Adjusting the centroid of the existing codebook vectors also decreases the quantization error.

Although the vector quantization algorithm as illustrated is processing 2D data the algorithm is not limited to only 2 dimensions and is successfully applied to the 20 dimensional weighted cepstral vectors.

4.2.4 Assigning a lip shape to the cepstral codebook vectors

The vector quantization algorithm creates small groups of similar input vectors by assigning each input vector to the nearest codebook vector. Each cepstral input vector of these small groups are associated with a given lip shape. The best choice of lip shape for a given codebook vector is made by averaging the lip shape values associated to each of the input vectors belonging to that codebook vector [Yamamoto98].

Figure 55 shows an example of how the output lip shape is chosen based on a given set of input cepstral vectors and the associated lip shape vectors. The top left graph of Figure 55 shows a 2D projection of the 20D input cepstral vectors. The top right graph is a graph of the lip width versus lip height.

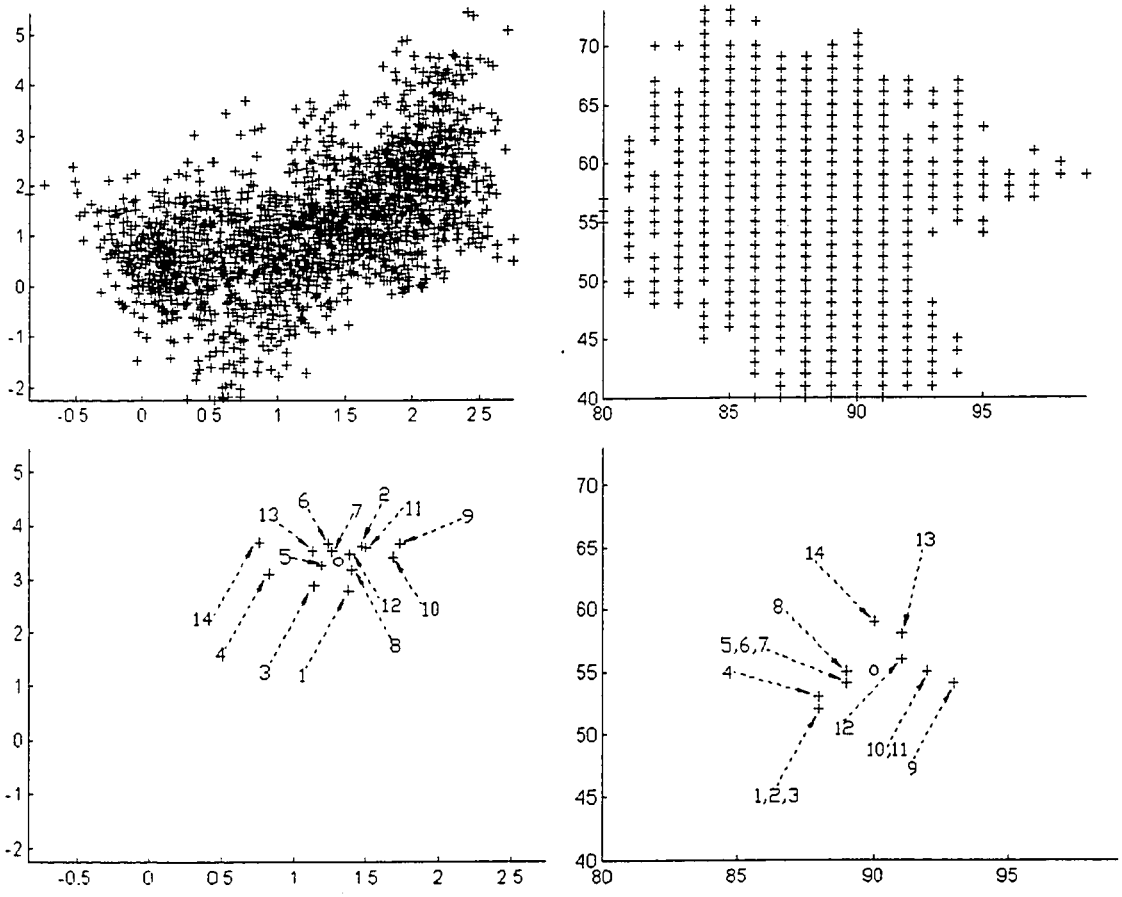

Figure 55 How the output lip model parameter values are chosen

The lower left graph shows a codebook vector (marked by 'o') and the input cepstral vectors nearest to it (marked by '+'). The lower right graph shows the average vector (marked by 'o') of the lip shape vectors (marked by '+'). Each input cepstral vector is marked with a number from 1 to 13. The lip shape vector associated to each cepstral vector is marked with the corresponding number. The single lip shape value chosen for the codebook cepstral vector is calculated by taking the average position of each of the lip shape vectors of the lower right graph of Figure 55.

The lip shape vectors are calculated for each of the codebook cepstral vectors and both sets of vectors are put into an array forming a one-to-one mapping. Table 3 shows a mapping of n cepstral vectors (20 elements each) associated to n lip model parameter vectors (2 elements each).

Cepstral Coefficients					One to One Association	Lip Model Parameters	
$C_0(0)$	$C_1(0)$	$C_2(0)$	...	$C_{20}(0)$	---->	ao(0)	bo(0)
$C_0(1)$	$C_1(1)$	$C_2(1)$	...	$C_{20}(1)$	---->	ao(1)	bo(1)
$C_0(2)$	$C_1(2)$	$C_2(2)$	...	$C_{20}(2)$	---->	ao(2)	bo(2)
$C_0(3)$	$C_1(3)$	$C_2(3)$	...	$C_{20}(3)$	---->	ao(3)	bo(3)
$C_0(4)$	$C_1(4)$	$C_2(4)$	...	$C_{20}(4)$	---->	ao(4)	bo(4)
$C_0(5)$	$C_1(5)$	$C_2(5)$	...	$C_{20}(5)$	---->	ao(5)	bo(5)
:	:	:	...	:	:	:	:
$C_0(n)$	$C_1(n)$	$C_2(n)$	...	$C_{20}(n)$	---->	ao(n)	bo(n)

Table 3 Cepstral coefficients to lip model parameter mapping.

4.3 Using the mapping

In the animation stage the mapping of reference cepstral coefficients associated to lip model parameters is used to estimate the lip positions from new audio input. The audio signal of the input video is analyzed and the cepstral coefficients of the video frames are calculated. The cepstral coefficients calculated from the frames of the input video are compared to the cepstral coefficients of the mapping. The set of cepstral coefficients of the mapping that are the closest match (in terms of the cepstral distance measure) to a given input frame $\bar{C}*$ is found. The lip model parameters associated with $\bar{C}*$ (from the reference mapping) are used to animate the lip model. Table 4 shows an abbreviated list of hypothetical distances between a given input cepstral vector $\bar{C}(in)$ and each of the codebook vectors. From the table the codebook vector $\bar{C}(19)$ is nearest to the input vector $\bar{C}(in)$.

	Input Vector	Codebook Vector	Distance
	C(in)	C(1)	25
	C(in)	C(2)	127
	C(in)	C(3)	55
	:	:	:
$\vec{C}* \rightarrow$	C(in)	C(19)	7
	:	:	:
	C(in)	C(n)	98

Table 4 Nearest cepstral vector.

Therefore $\bar{C}* = \bar{C}(19)$. Then using the mapping described in table 3 the best estimate for the lip width and lip height for the given input cepstral vector $\bar{C}(in)$ are the values stored in $A_o(19)$ and $B_o(19)$.

4.4 References

[Yamamoto98], [Gray00], [Rabiner93], [Kovesi99]

CHAPTER 5 – System implementation

5.1 Practical considerations

This section outlines practical considerations for the hardware and software components of the project implementation.

5.1.1 System hardware

The capture and display of the audio/video is done in real time so the computer system used to do the implementation must be fast enough to capture and display video in real time. This project is implemented on a PC with an Intel Pentium MMX™ processor running at 166MHz, with 32MB ram, and PCI data bus, under Windows 95™. Video capture was done using a Winnov Videum AV™ PCI capture card. One advantage of using this Winnov™ card is that the audio and video capture is done with a single card allowing for hardware synchronization of the two media. Audio was recorded at 11.025 kHz with a 16-bit precision. Video resolution was 320x240 pixels with 24-bit color. The video capture card, the system speed, as well as the hard disk transfer rate was sufficient to allow the uncompressed capture of an audio-video file with the previous specifications at 30 frames per second.

5.1.2 Captured file format

The captured audio-video files were stored using the AVI (Audio Video Interleaved) file type. AVI files can have none or more audio streams and none or more video streams and can also contain none or more text streams. The streams are independent one from the other but are stored on the disk with the streams interleaved, meaning that a block of video is written followed by a block of audio followed by a block of text (if these streams exist in the file). Interleaving the frames allows for a minimum of memory to be used to synchronize the audio and video during playback.

5.1.3 Programming languages

Input, editing, and output of the AVI files is programmed in C under the Win32 API™ using the Microsoft Visual C++ 5.0™ and the AVIFile™ library. The AVIFile™ library contains many functions specific to the input, decompression, compression, display, and output of AVI files [Petzold99].

The processing of the audio signal is done using MATLAB™ scripts. MATLAB™ stands for *matrix laboratory.* It is used to perform the audio processing and mapping building parts of this project because it offers advantages over programming in C. MATLAB™ is interactive and allows the user to easily access and process numerical data in the form of matrices. Mathematical functions are applied to symbolic data in MATLAB™ as you would write them mathematically. The language is interpreted allowing the user to experiment with different commands and to inquire about the value or values of the data after any command. Data values can be visualized using one of the versatile plot commands. There are several ready-made functions useful for speech processing included with Matlab. Solutions to numerical problems can be found in a fraction of the time that it would take to solve using C. MATLAB™ frees the programmer from having to worry about strict syntax and allows one to focus directly on the task at hand. Another virtue of programming under the MATLAB™ environment is that once the MATLAB™ script is working as desired this code can be compiled into fast machine code for real-time application. A review of the usefulness of MATLAB™ for signal processing can be found in [Painter96].

5.2 How data and programs relate one to the other

Figure 56 shows how the different programming environments work together to complete this project. Items in rectangles are data and items in parallelograms are programs. The .EXE programs were compiled using Microsoft Visual C++™ and the .M files are MATLAB™ scripts. The figure is divided into 2 stages. One for the training stage and another for the animation stage.

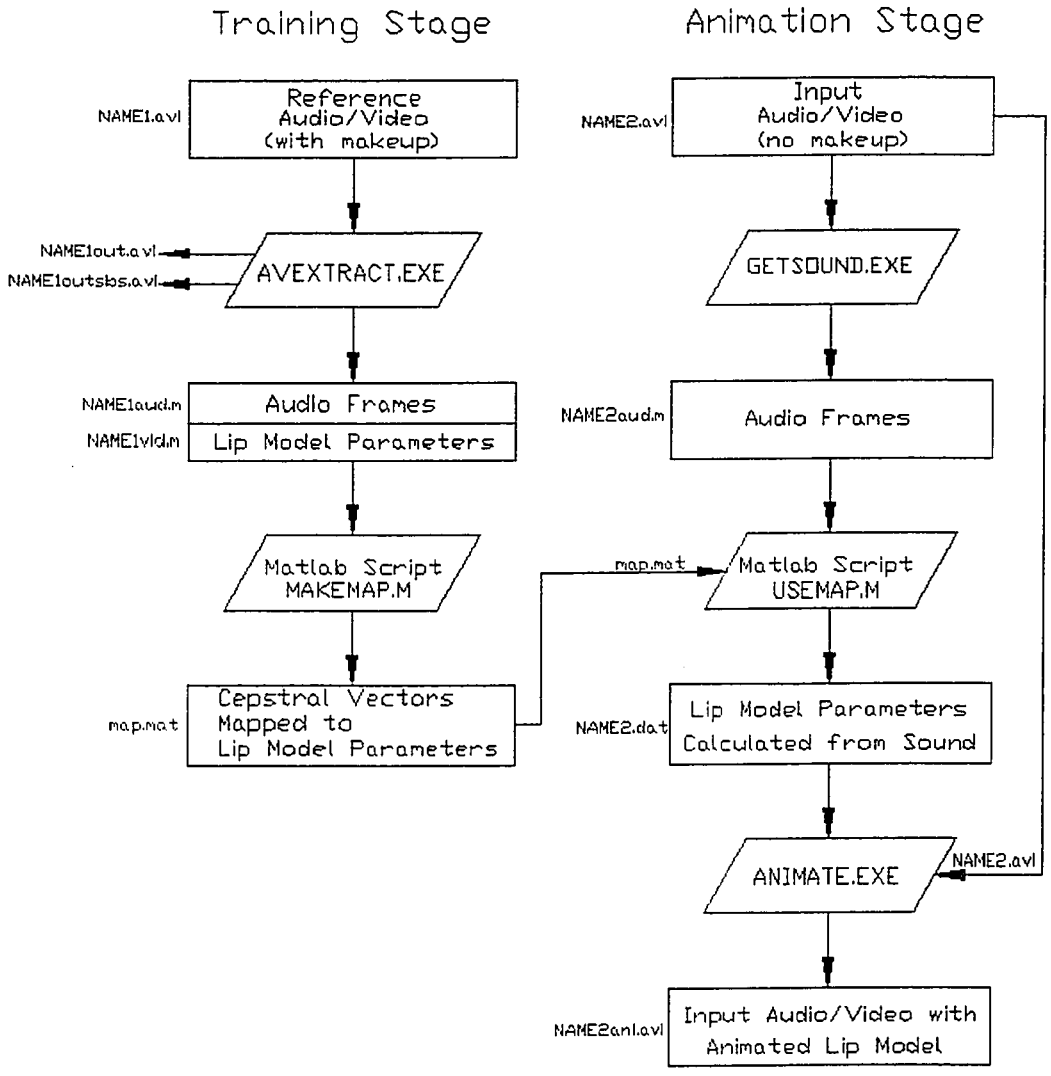

Figure 56 How the programs and data relate one to the other.

5.2.1 Training stage

The training stage is the section of the project where the speaker dependent audio parameter to lip model parameter mapping is created. The steps taken during the training stage of this project are described in the following sections.

5.2.1.1 Reference video

The first rectangle of Figure 56 represents the input reference video. The programs that analyze the input video assume that the input video file be recorded using the following specifications. The input video must be recorded at 30 frames per second. It must have 24-bit RGB color. It must be 320x240 pixels in size. It must have 16-bit audio recorded at 11025 audio samples per second. Any compression method for the video is allowed but no compression of the audio is allowed. The reference video is recorded under laboratory conditions (see section 5.3). The lighting is carefully adjusted so no shadows appear on the lips. The background noise level is kept low – the computer's cooling fan noise must be minimized. The speaking person's head position is adjusted so that the lips are set squarely in front, and are placed at a constant distance from the camera. Blue makeup is applied to the speaker's lips. All these restrictions (aside from low background noise) are needed only for the training stage of the project because in the animation stage the video portion of the input AVI file is not considered only the audio signal is examined.

5.2.1.2 AVEXTRACT.EXE

From Figure 56 we see that the program avextract.exe is responsible for taking the reference AVI file, named NAME1.avi, ('NAME1' can be replaced with any given name) and determining the optimum lip model parameters for each video frame of the file. The AVEXTRACT.EXE program also takes the audio stream and cuts it into frames and saves it in a file type MATLAB™ can read. The program AVEXTRACT.EXE also outputs 2 AVI files useful for demonstration purposes. The first demonstration file

NAME1out.avi shows the images of the speaker's lips taken from the reference file NAME1.avi with the synchronized audio stream. Overlapped on the image is a tracing of the lip contour model generated from the optimum lip shape parameters. To the left of this image is a display of all the current parameters of the lip contour model and below this image is a graphical representation of the audio signal. An example frame from a NAME1out.avi file is displayed in Figure 57.

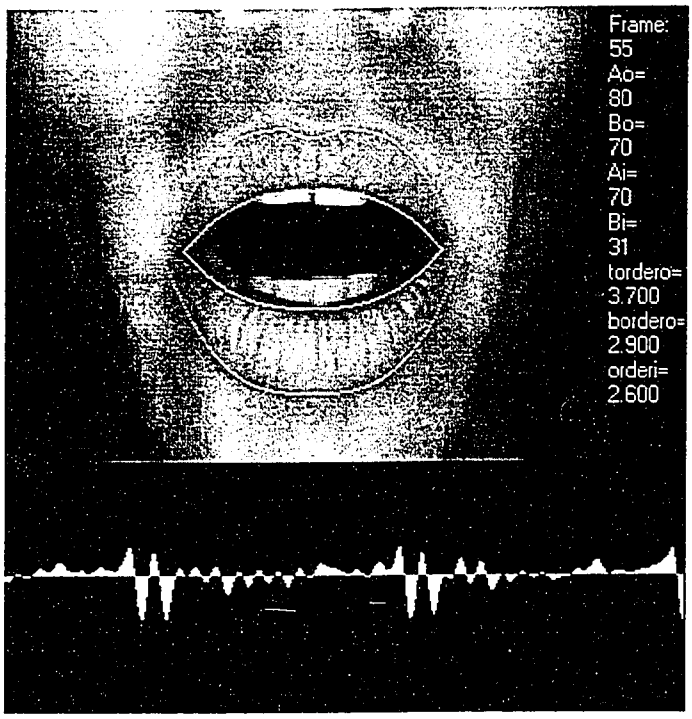

Figure 57 Frame taken from demonstration file NAME1out.avi

The second demonstration file is named NAME1outsbs.avi. This file only has a video stream, no audio stream, and it shows how the lip contour model was deformed in order to find the optimal position and shape for each of the original video frames. An example of a series of frames showing how the lip model deformed itself to find the optimal set of model parameters for a given input video frame is displayed in Figure 58.

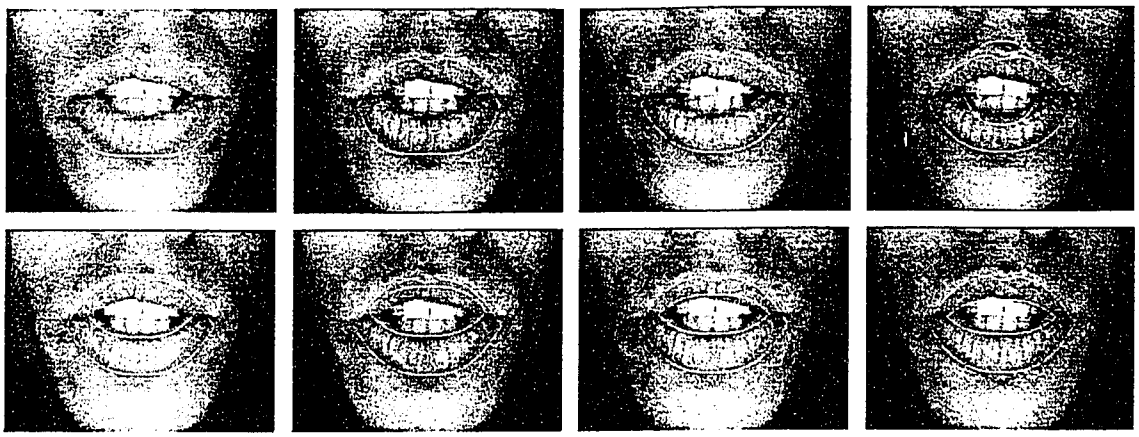

Figure 58 Series of frames showing the lip model being molded to the shape of the speaker lips.

5.2.1.3 MAKEMAP.M

The next program of the training stage, as illustrated by Figure 56, is the MATLAB™ script, MAKEMAP.M. MAKEMAP.M loads the audio frames that were saved to a file by AVEXTRACT.EXE and processes them using a Linear Predictive Coding (LPC) analysis then transforms the LPC coefficients into weighted cepstral coefficients. These cepstral coefficients are represented by a set of codebook vectors using a vector quantization algorithm. The vector quantization algorithm creates small groups of similar input vectors by assigning each input vector to the nearest codebook vector. Each cepstral input vector of these small groups have already been associated with a given lip shape. The best choice of lip shape for a given codebook vector is made by averaging the lip shape values associated to each of the input vectors belonging to that codebook vector (see section 4.2.4). The complete mapping linking the audio parameters to the lip shape parameters consists of the cepstral vector codebook associated to the averaged lip shape parameter values. These values are stored in the mapping file map.mat. It is this mapping that is used when estimating the shape of the lips from the voice stream in the animation stage.

5.2.2 Animation stage

During the training stage, the audio parameter to lip shape parameter mapping tailored to a specific user was created. In the animation stage the mapping is used to animate the lip model from the voice signal. The steps taken during the animation stage are outlined in the following section.

5.2.2.1 Animation stage input video

In the animation stage an input video is recorded and saved as NAME2.avi. As was the case for the input video to the training stage the video to the animation stage must be recorded at 30 frames per second. It must have 24-bit RGB color. It must be 320x240 pixels in size. It must have 16-bit audio recorded at 11025 audio samples per second. Any compression method for the video is allowed but no compression of the audio is allowed. These specific settings are required because the programs that follow assume these specifications. This video file must also have low background noise but it need not have any specific lighting, it need not have the speaker's head placed in a specific position, nor have the lips painted blue with blue makeup. These restrictions are not needed because the video stream is not analyzed; the audio stream is the only part of the input AVI file that is analyzed in the animation stage.

5.2.2.2 GETSOUND.EXE

GETSOUND.EXE loads the file NAME2.avi and cuts the audio stream into frames and saves the audio frames in a format MATLAB™ can read. The audio frames from GETSOUND.EXE and the mapping created in the training stage are read into the MATLAB™ script USEMAP.M.

5.2.2.3 USEMAP.M

USEMAP.M loads the audio frames saved by GETSOUND.EXE and transforms these audio frames into vectors of cepstral coefficients. Then a cepstral distance measure is used to find the vector of the audio parameter codebook C* that is nearest to the cepstral vectors calculated from each of the input audio frames. Next, the lip model parameters describing the shape of the lips are calculated using the weighting functions, the lip model codebook, and C*. The calculated lip model parameters are saved in a file called NAME2.dat and are used as input to the program ANIMATE.EXE to animate the lip contour model.

5.2.2.4 ANIMATE.EXE

ANIMATE.EXE takes the predicted lip model parameters stored in NAME2.dat as well as the input AVI file NAME2.avi and creates a composite AVI file with the name NAME2ani.avi. An example frame from an output AVI file is contained in Figure 59. In the left half of the window there is a video frame from the file NAME2.avi and on the right half of the window there is an image of the lip model animated from the parameters calculated by USEMAP.M. This program gives a visual measure of the systems performance.

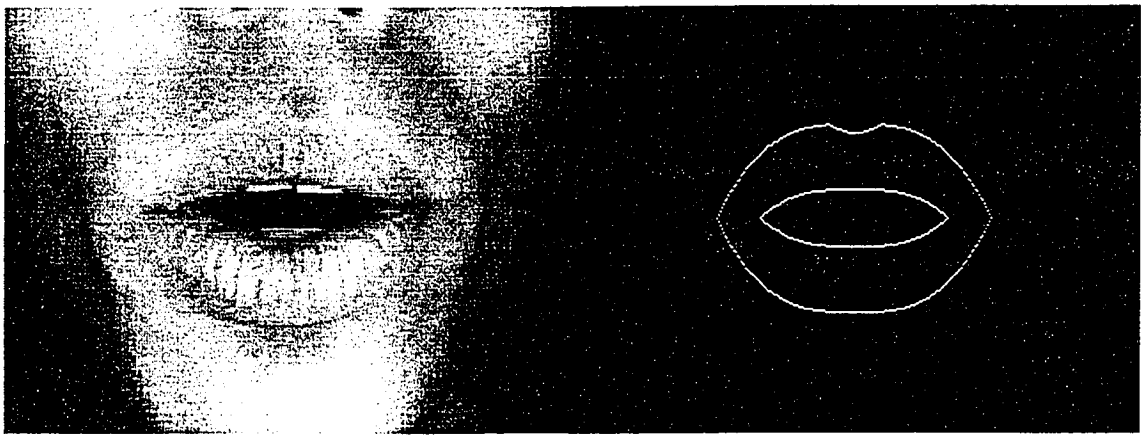

Figure 59 Output file showing an input frame and the animated lip model.

5.3 Laboratory Audio/Video Capture conditions

The audio and video conditions necessary for the capture of a clean reference audio/video file are outlined here. The steps taken here remedy the problems of gaps in the blobs outlined in section 3.2.3.4. We will also introduce other limitations and remedies associated with the capture of the audio/video files.

5.3.1 Avoiding gaps

The proper choice of makeup color and lighting conditions are important. If the makeup color is too dark or if shadows are created on the lips some of the blue pixels can become too saturated (black or some shade of gray). The video processing procedure can mistake the dark pixels as black pixels. As was mentioned earlier the hue of a completely saturated (black, white, or some shade of gray) pixel is undetermined. If the hue of any given pixel in the input image is undefined then the corresponding pixel of the grayscale image is black. Therefore if the chosen makeup is too dark or if shadows are cast on the lips, some of the pixels of the lips can be incorrectly classified as background pixels in the binary image. Figure 60 shows an example of how dark makeup and shadow can cause some of the lip pixels to be incorrectly classified as background pixels.

Figure 60 Makeup is too dark and lighting is insufficient.

Because a point light source was placed above the speaker's lips a shadow from the upper lip was cast over the corners of the lips. This shadow caused the pixels at the corner of the lips in binary image (the image on the right of Figure 60) to be classified as background pixels. To include more pixels as foreground we could lower the threshold used to transform the grayscale image into the binary image. By lowering the threshold more of the pixels of the lips would be classified as foreground, but more pixels that are not part of the lips will be classified as foreground as well. The best solution is to improve the quality of the captured image.

5.3.1.1 Choice of makeup color

A light hue of blue reflects more light and is not as easily confused with black. This simple step will increase the contrast between the pixels of the lips and background pixels in the grayscale image (the image that is the result of the original image transformed by hue). Allowing more of the lip pixels to be classified as foreground.

5.3.1.2 Lighting conditions

To avoid the creation of shadows a point light that was originally used (see Figure 60) was replaced with a fluorescent light ring. This ring is placed around the camera lens and projects light on the speaker's lips from all angles. Because the light is coming from all angles shadows (especially at the corners of the lips) are reduced to a minimum. Figure 61 shows how the fluorescent ring was placed on the camera.

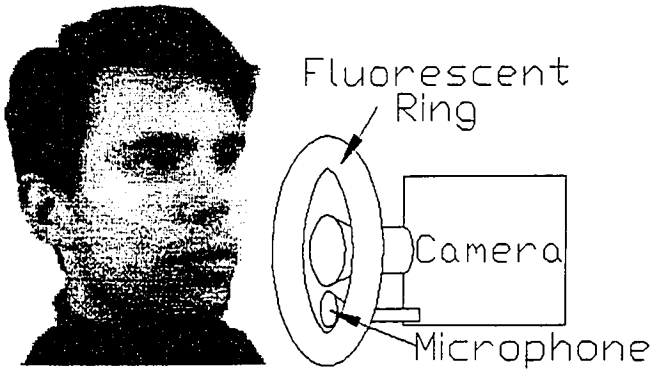

Figure 61 Lighting used during audio/video capture.

5.3.1.3 Improved image quality

By placing the fluorescent ring around the camera lens and changing the color of the lip makeup, a large improvement in the quality of the captured images is observed. Figure 62 shows an example of an input image, a grayscale image (the original image transformed according to hue) and a binary image (the grayscale image thresholded). It is easy to see that the pixels of the lips in Figure 62 are much more clearly separated from the pixels of the background than they were in Figure 60.

Figure 62 Lighter colored makeup and improved lighting.

5.3.2 Audio capture

The microphone is placed near the speaker's lips (see Figure 61) so that the voice signal can be captured with as little background noise as possible. Care must be taken to minimize background noise including the noise produced by the computer's cooling fan. An isolation box was created to encase the computer and to stop the noise produced by the computer's fan from being picked up by the microphone. The noise from the fluorescent light transformer was also isolated from the microphone.

5.3.3 Camera alignment

The model used to measure the shape of the lips is a 2D model. This model cannot accurately measure the shape of the lips if the head is not set squarely in front of the camera. Also the distance between the camera and the subject's lips must remain

constant. Changing the lip to camera distance changes the measured height and width of the lips. Proper camera alignment is important only in the training stage. When the system is in the animation stage only the audio portion of the audio/video file is processed. Because the video is not processed the speaker's head can be in any position. It does not even have to be in the camera's view, because the lip shape will be found using the voice.

5.3.3.1 3D head rotation

Because the simpler 2D model was chosen the speaker's head must be aligned squarely with the camera. The model cannot account for rotations in 3 dimensions. Figure 63 defines the 3 axis around which the human head can rotate.

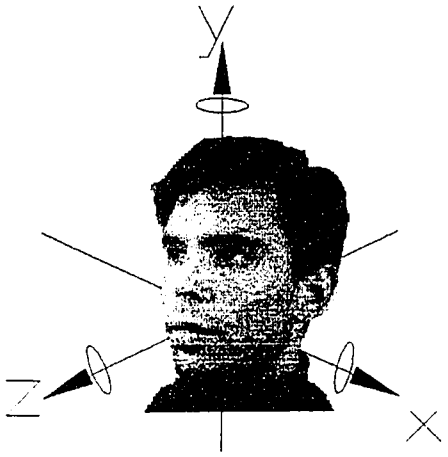

Figure 63 Rotation axis definition.

Incorrectly aligning the camera will cause the lip model to incorrectly measure the lip shape as illustrated in Figure 64.

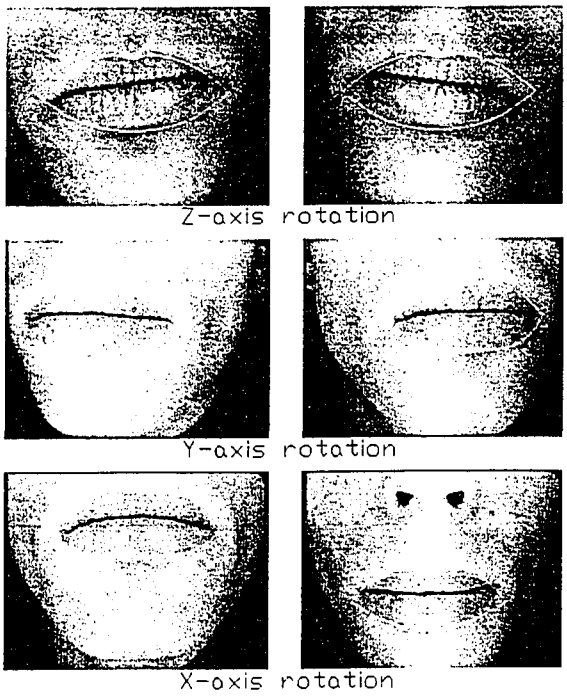

Figure 64 Lip model shape measurement when camera not aligned with lips.

5.3.3.2 Lip to camera distance

Scale is also important. The lips can seem bigger if they are closer to the camera's lens. Care must be taken to record the entire training video at the same scale (with the lips at a constant distance from the camera lens). The same lip position recorded with the lips at different distances from the camera lens will give different measurements. Figure 65 shows the same lip shape measured with 2 different camera to lip distances.

Figure 65 Lips in same position but at different distances from camera lens.

5.4 References

[Petzold99], [Simon99], [Painter96], [Rabiner93]

CHAPTER 6 – Experimental Results

6.1 Measuring the system performance

This chapter contains test results of the performance of the voice driven lip shape animation system. To show how this system can be trained to recognize the shape of the lips of a wide range of speakers, the system is tested on two speakers one male one female. The female speaker spoke the ten digits 0 to 9 in French (zero, un, deux, trois...neuf). While the male speaker spoke the ten digits 0 to 9 in English (zero, one, two, three...nine).

6.1.1 Testing procedure

The test subjects are recorded speaking their test phrase several times. Each time being careful to remain squarely aligned to the camera and to maintain a constant distance from the camera lens. Other optimum recording conditions outlined in section 5.3 are also observed.

All but one of the audio/video reference files are used as input to the training stage. The final training video will be used to test the system performance. In the training stage the optimum lip model parameters for each of the video frames are measured and the cepstral coefficients for each of the audio frames are calculated. All the time aligned data from these reference files is used to create the cepstral coefficient to lip shape parameter mapping as outlined in section 4.2.

The remaining reference audio/video file is also analyzed to extract the lip model parameters, but these parameters are not included in the reference mapping. This remaining audio/video file is used as input to the animation stage. The voice stream of the video is used to estimate the speaker's lip shape. The lip model parameters estimated from the audio stream are compared to the lip model parameters extracted from the video stream, giving a measure of the performance of the system.

Audio frames whose energy is below a pre-determined threshold (e.g. audio frames taken between words) are excluded from the comparison. The system is not expected to be able to determine the shape of the speaker's lips if he/she is not speaking.

6.1.2 Female speaker test results

Figure 66 contains the results obtained from the female speaker. The results are

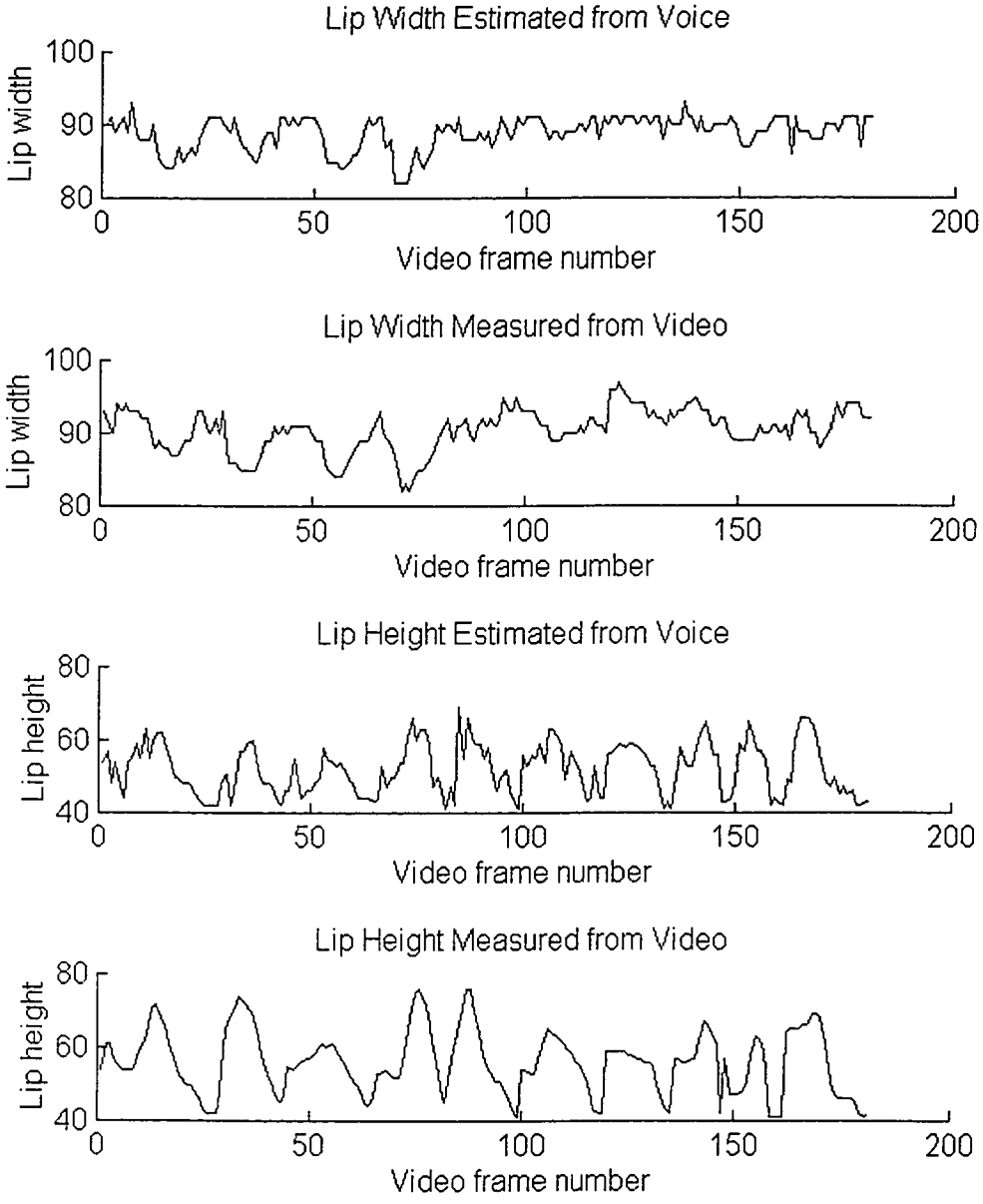

Figure 66 The lip model width and height estimated and measured for the female speaker.

displayed using 4 graphs. The first graph is of the lip model width estimated from the voice as a function of frame number. The second graph is of the lip model width measured from the video as a function of frame number. The third and fourth graphs are of the lip model height estimated from the voice and the lip model height measured from the video respectively.

To more clearly show the similarities and differences between the height and width of the lip model predicted from the voice and measured from the video, Figure 67 contains both graphs overlapped in one figure.

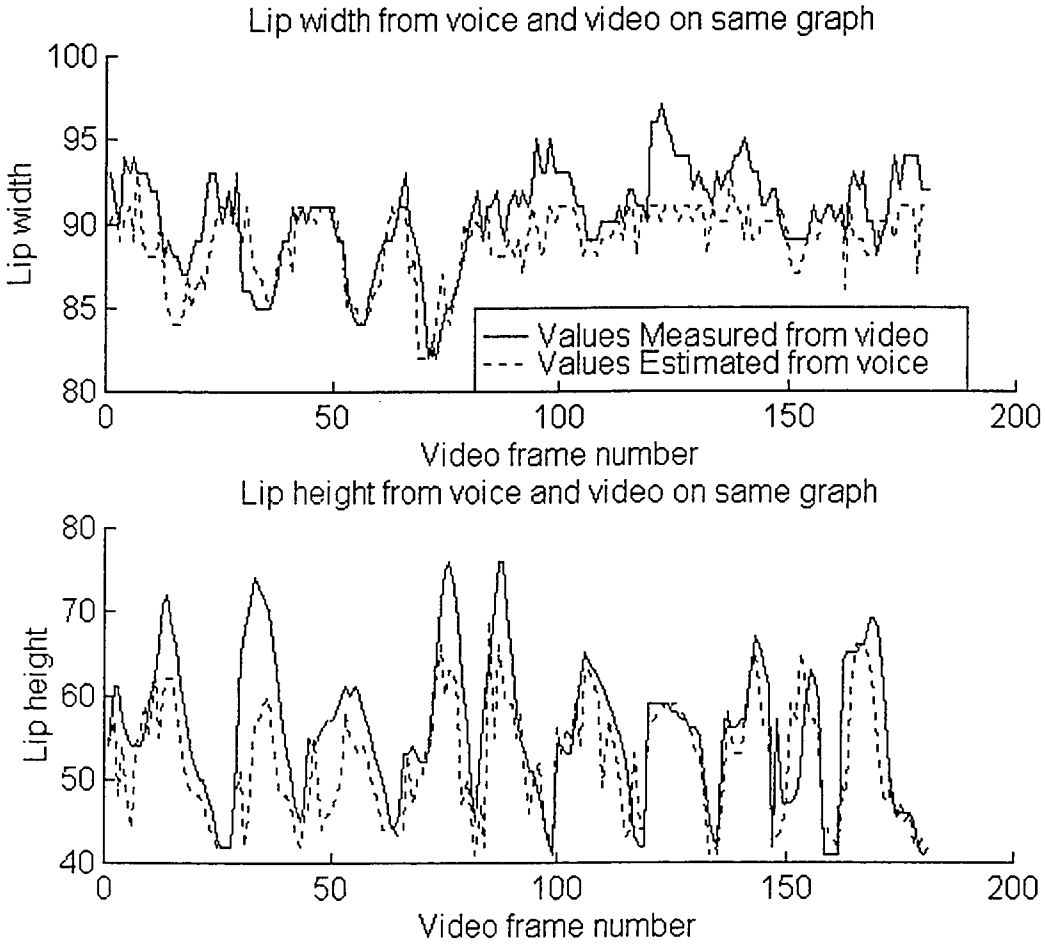

Figure 67 Lip width and height estimated and measured for female speaker on one graph.

The lip model parameter values estimated from the sound are similar to the values measured from the video. Taking the difference between the two width graphs (the

width estimated from voice and the width measured from video) and between the two height graphs (the height estimated from voice and the height measured from video) we can create a histogram of the errors. The histogram of the difference between the two width graphs and the histogram of the difference between the two height graphs are contained in Figure 68.

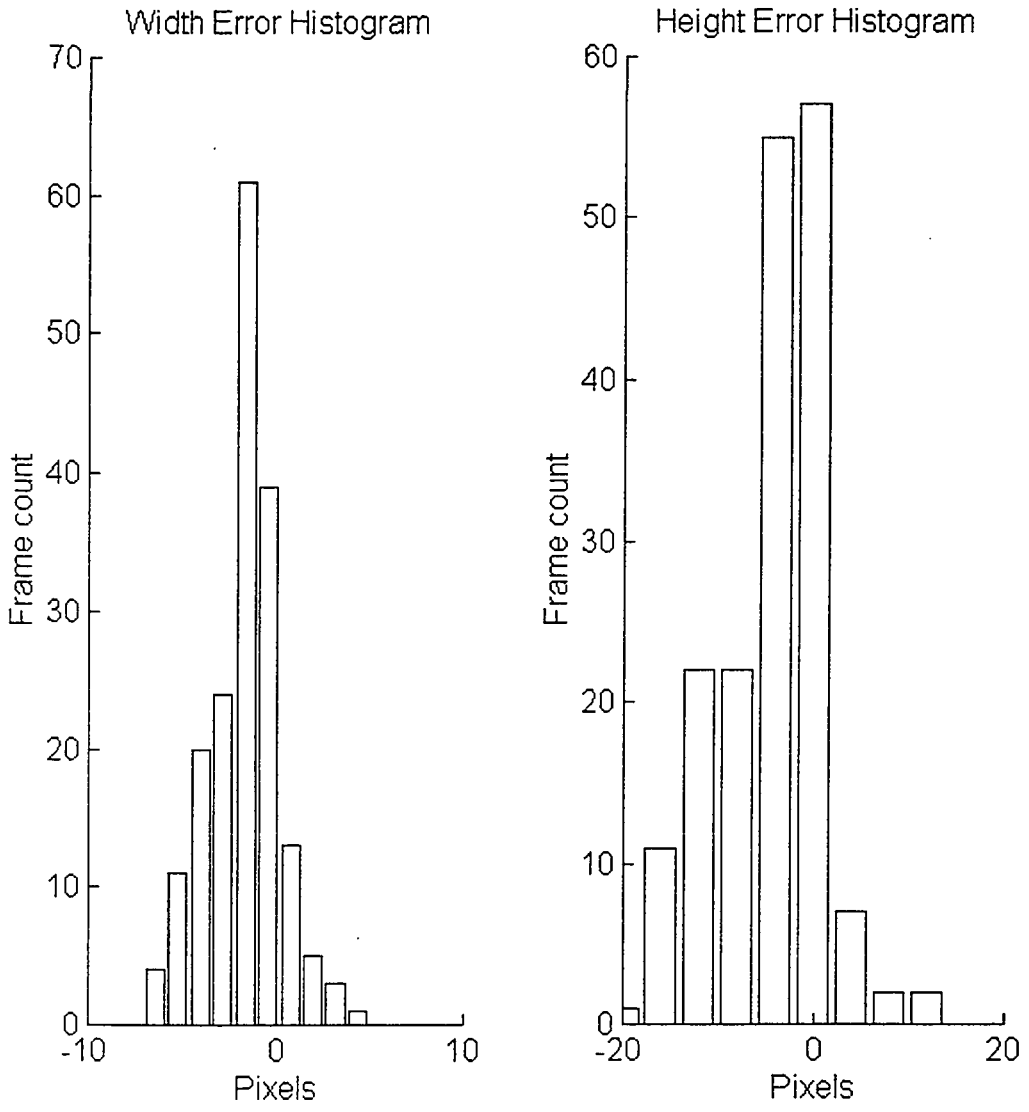

Figure 68 Histogram of the difference between the width and height graphs for female speaker.

The mean value of the difference between the lip width estimated from the voice and the width measured from the video is -1.54 pixels and the standard deviation is 2.05

pixels. The mean value of the difference between the lip height estimated from the voice and the height measured from the video is –4.08 pixels and the standard deviation is 5.98.

6.1.3 Male speaker test results

The lip shape predicted from the audio signal of an AVI file is compared with the lip shape measured from the video portion of the same AVI file as the male speaker said the ten English digits (zero, one, two,...nine). The graphs of the lip model width estimated from the voice, lip model width measured from the video, lip model height estimated from the voice, and the lip model height measured from the video for the male test subject are given in figure 69.

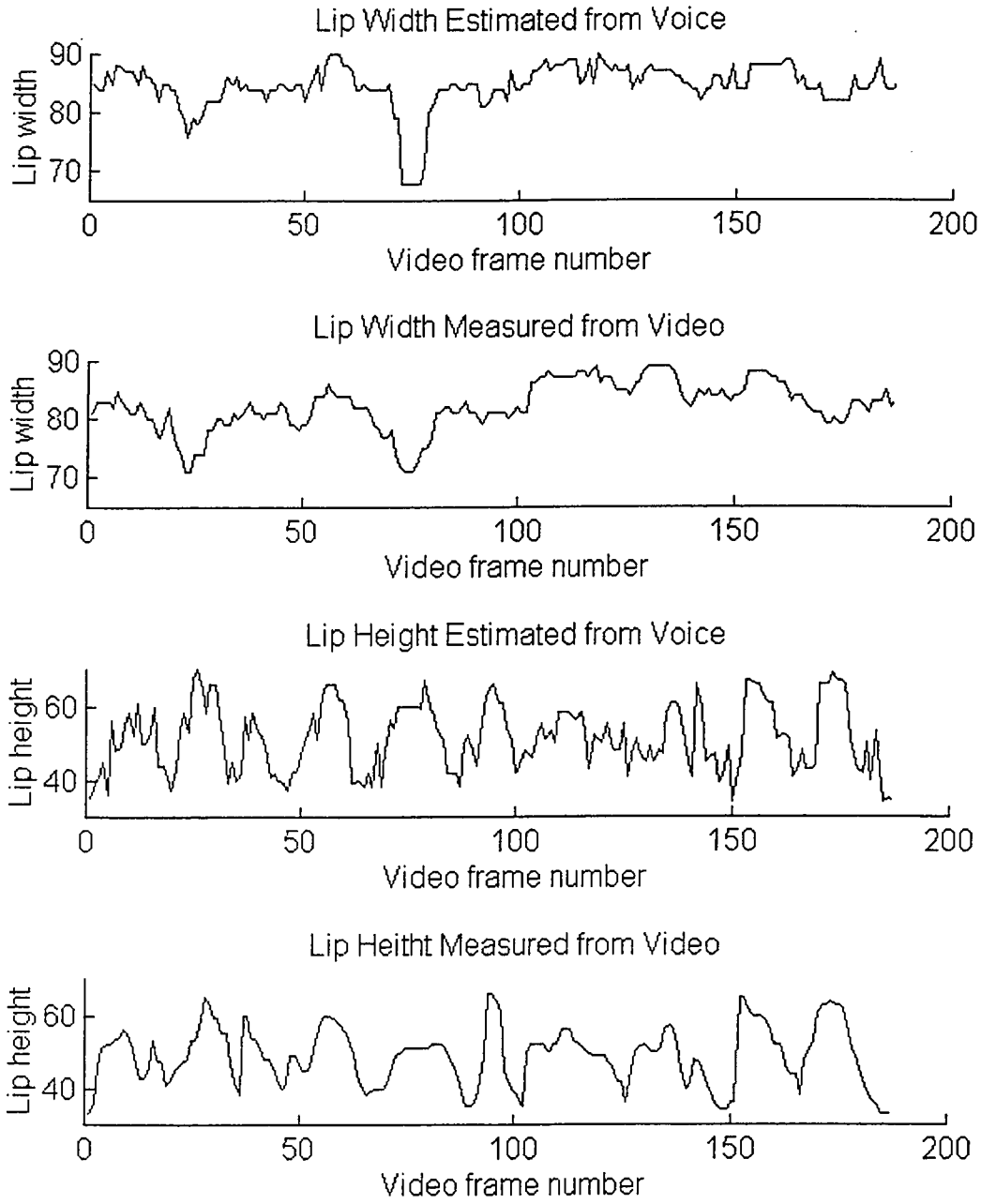

Figure 69 The lip model width and height estimated and measured for the male speaker.

The next figure shows both the estimated and measured tracings on the same graph. More clearly illustrating how similar or dissimilar the estimated lip model height and width are from the measured lip model height and width for the male speaker.

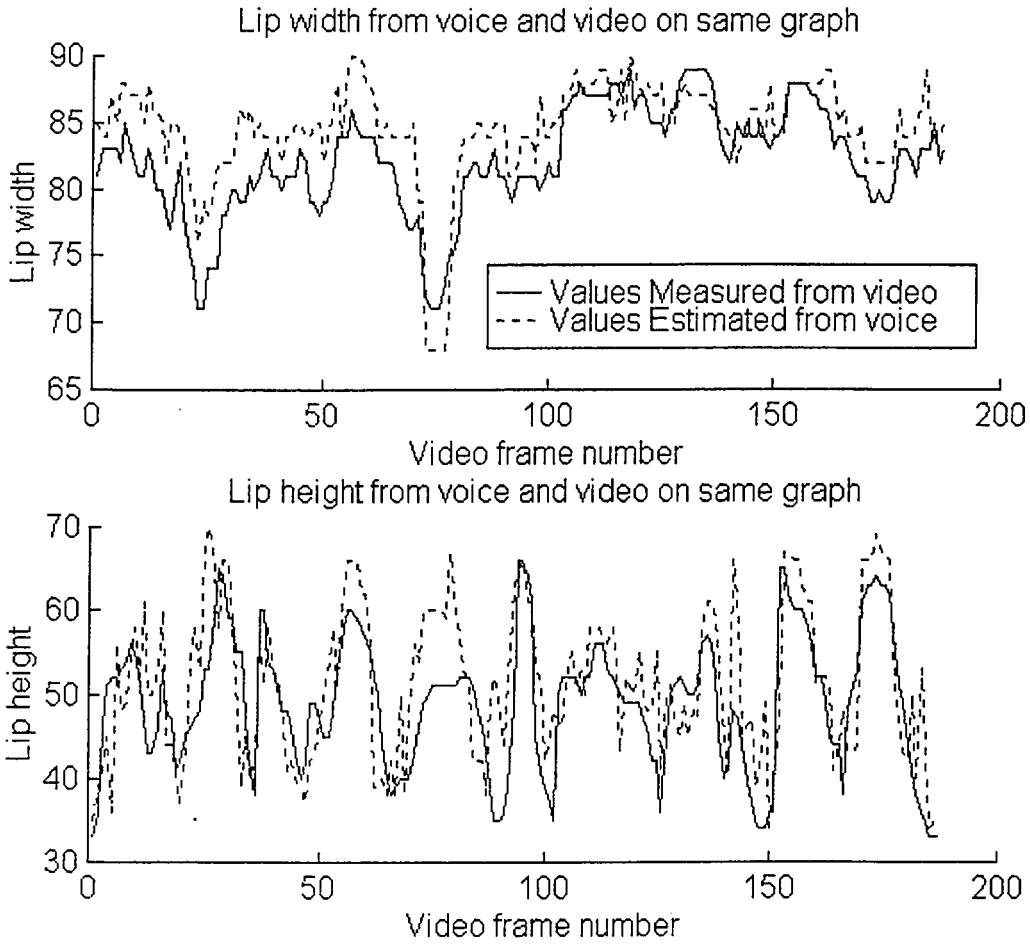

Figure 70 Lip width and height estimated and measured for male speaker overlapped on one graph.

Taking the difference between the two width graphs (the width estimated from voice and the width measured from video) and between the two height graphs (the height estimated from voice and the height measured from video) we can create a histogram of the errors for the male speaker too. The histograms are contained in Figure 71.

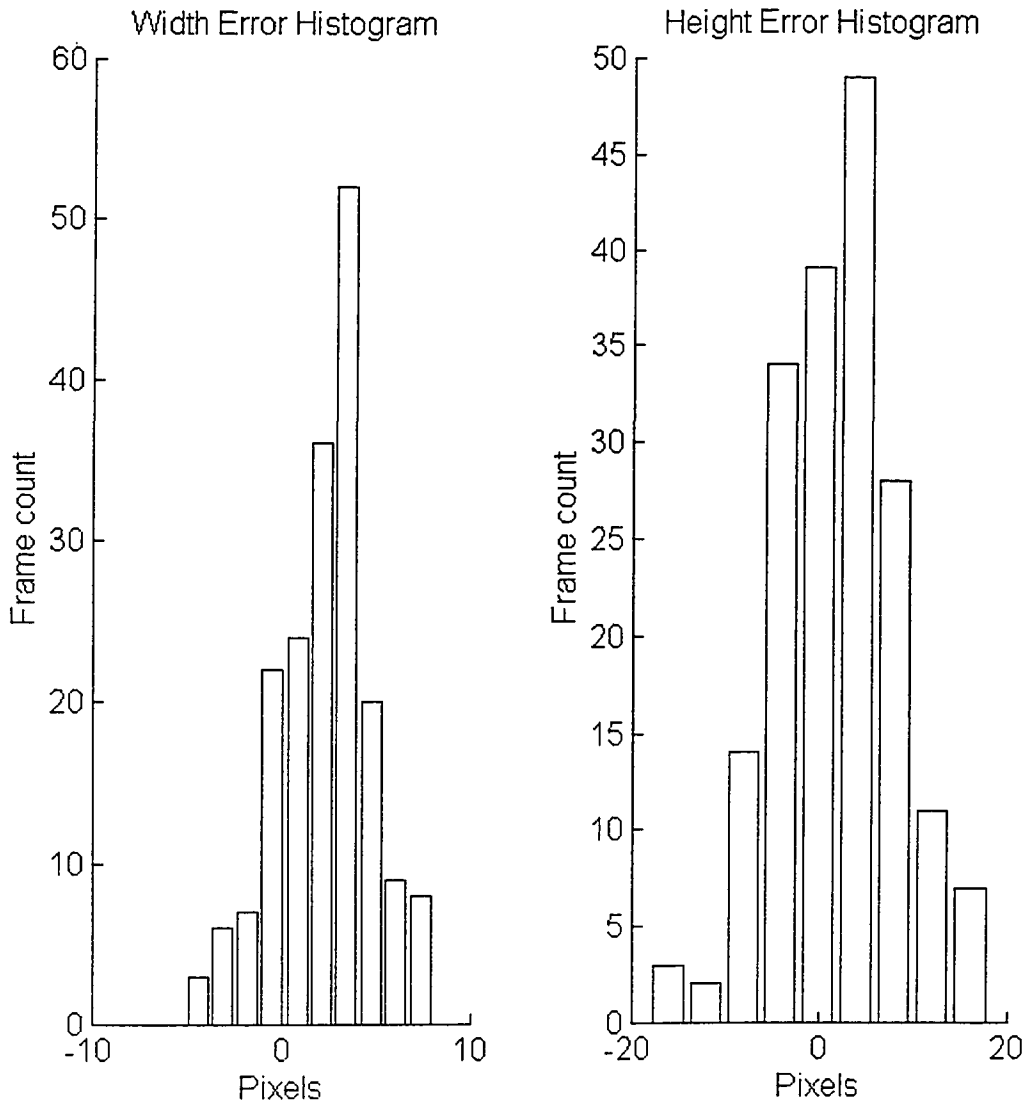

Figure 71 Histogram of the difference between the width and height graphs for male speaker.

The mean value of the difference between the lip width estimated from the voice and the width measured from the video is 2.34 pixels and the standard deviation is 2.54 pixels. The mean value of the difference between the lip height estimated from the voice and the height measured from the video is 2.35 pixels and the standard deviation is 6.56 pixels. See the next section discusses these results.

6.2 Discussion of the results

The results for both the male and the female speaker show that the system described in this thesis can effectively determine the shape of the lips from the voice signal to a satisfactory degree of accuracy. This accuracy depends largely on the quality of the information contained within the audio parameter to lips shape model mapping.

6.2.1 The ideal voice parameter to lips shape parameter mapping

Ideally the mapping should contain a wide variety of audio parameters associated to the *typical* lip shape parameter values. The difficulty is in deciding what lip shapes are typical for a given speech sound for a given speaker. A large sample of articulatory and audio information should create a more precise mapping by averaging variation that naturally occurs when speaking. However, inconsistent or contradictory articulatory and audio information should be excluded from the mapping (e.g. two very similar sounds giving two very different lip shapes).

6.2.2 Limitations of this method

The frame by frame approach used in this system is cannot model contextual effects that exist when people speak. Two simple cases of contextual effects that cannot be modeled with a frame by frame approach are articulatory anticipation and articulatory retention [Jourlin96], [Abry94].

6.2.2.1 Articulatory anticipation phenomenon

The articulatory anticipation is when a speaker articulates prior to producing sound. Sometimes at the beginning of words or new sound structures a speaker will first form a shape with their lips then begin to produce sound. This event cannot be modeled with the frame by frame approach to driving an animated lip model from the voice stream. A

system that accounts for context is needed to accurately drive the lip shape model when there is no sound present at the current moment. An example of articulatory anticipation is illustrated in Figure 72. This figure shows a series of 24 video frames with a graphical representation of the audio track shown below each video frame. The speaker's mouth opened 21 frames (21 * 33 = 700 ms) before any sound was produced.

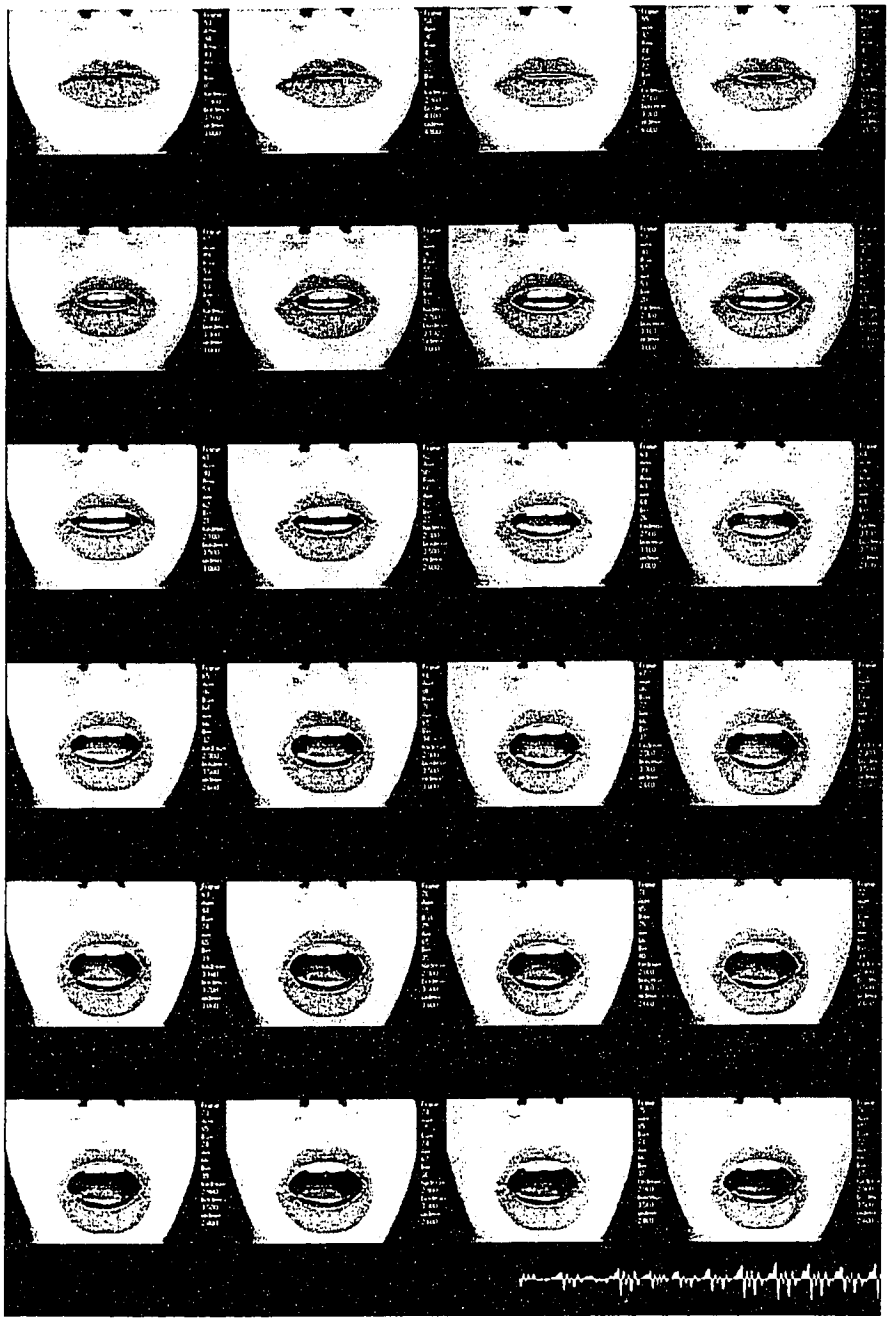

Figure 72 Example showing the articulatory anticipation phenomenon.

The articulatory anticipation phenomena was slightly more apparent for the female speaker who spoke the ten French digits (zero, un, deux...neuf) than for the male speaker who spoke the English digits (zero, one, two...nine). This can be an artifact of the language spoken or it can also be attributed to the articulating behavior of the individual speakers.

6.2.2.2 Articulatory retention phenomenon

The retention phenomenon is when the speaker stops creating sound and the lips continue to articulate. This phenomenon occurred less frequently than the anticipation phenomenon and had a shorter duration for the speakers considered in this analysis. Figure 73 shows an example of the retention phenomenon. The duration of this phenomenon in the example below is only about 5 frames (5 * 33 = 165 ms).

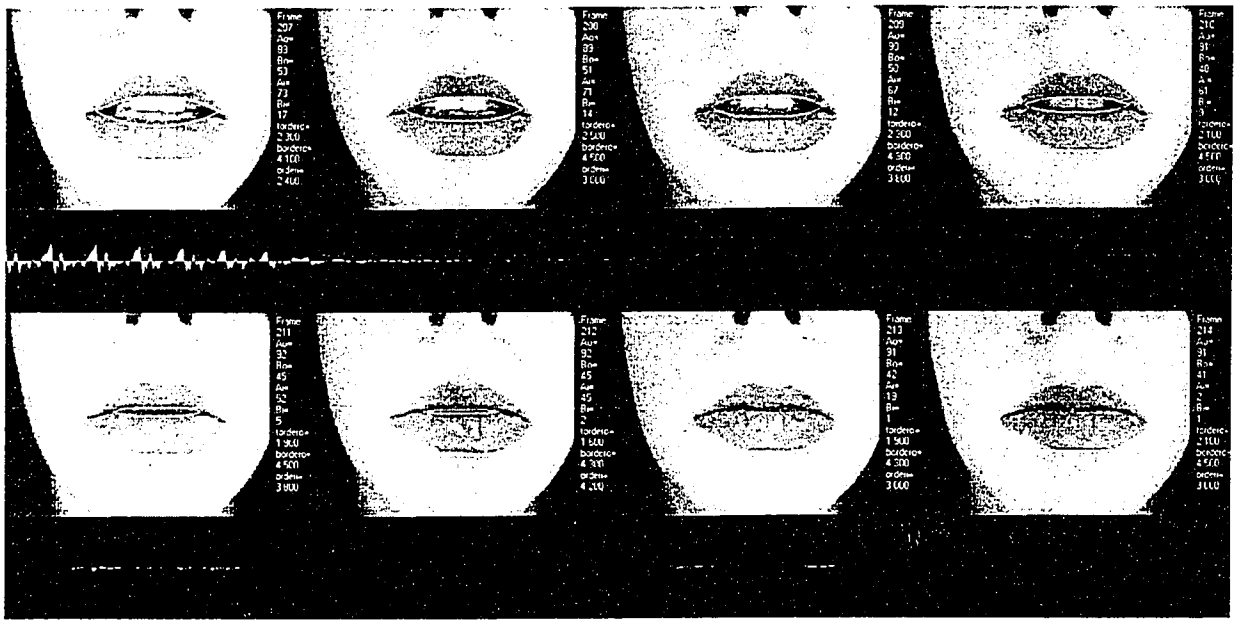

Figure 73 Example showing the articulatory retention phenomenon.

Using a frame by frame analysis method the shape of the speaker's lip shape remains undefined when there is no sound being produced. Obviously there is need for more research to solve this problem.

6.3 References

[Abry94], [Cathiard92], [Escudier90], [Jourlin96], [Jourlin98].

CHAPTER 7 – Conclusions and Further Development

7.1 Conclusion

This thesis describes an experimental system that uses the voice track to determine the shape of a speaker's lips. A parametric model was introduced and was used to measure and display lip shape. The cepstral coefficients derived from a linear predictive coding analysis of speech were used as audio recognition features. The system was divided into two stages the training stage and the animation stage.

In the training stage the speaker dependent cepstral coefficient to lip shape parameter mapping was created. A vector quantization algorithm was used to group similar cepstral coefficients together and the lip shapes associated with those cepstral coefficients were averaged to create one average lip shape for the entire cepstral coefficient group. Averaging the lip shapes from similar cepstral coefficients averaged out the variation that naturally occurs when a person speaks and also created a compact mapping.

In the animation stage the previously created cepstral coefficient to lip shape parameter mapping was used along with the audio analysis to determine the shape of a speaker's lips from the voice stream.

The system was tested and the results show that this method works to a satisfactory degree of accuracy. This accuracy depends largely on the quality of the information present in the mapping. These results warrant, in our opinion, further research and development.

7.2 Further Development

Methods for improving the accuracy or expanding the capabilities of this lip shape from the voice stream system are of interest for further development.

7.2.1 Improving the mapping

One way to improve the accuracy of this system is to improve the quality of the audio parameter to lip shape mapping. Excluding contradictory articulatory and audio information from a large sample of articulatory and audio information using some kind of processing method could be introduced to weed out inconsistent information before it is used in building the reference mapping.

7.2.2 Unlimited vocabulary

Perhaps the most critical element to improve is to create a speaker dependent audio to video mapping that can be used to determine the lip shape of an *unlimited vocabulary*. Some kind of phonetically balanced reference file could be recorded to create a mapping that would contain enough information to allow the lip shape to be driven from any speech segment a given speaker utters.

7.2.3 Speaker independence

Another direction for the expansion of this system is to create a system to animate a lip model from the voice tract that is speaker independent. However, there is so much variation in the lip shape of one individual, trying to design a system that takes into account variations that occur between individuals might prove to be a formidable task. The best solution might be continue developing better speaker dependent systems by perfecting the training (mapping building) algorithms.

7.2.4 Contextual effects

A method of recognizing not only what is contained within a given frame but also what is going on over a group of frames is required to effectively model the contextual effects of human speech. The shape a person's lips take does not only depend on the sound spoken at that moment but also what sound was spoken just before and which sound is coming after. A moving window that considers a group of audio frames at one time is a possible answer to this problem.

7.2.5 Completing the global model based communication system

To complete the entire model based audio/video communication system outlined in the introduction, section 1.1.2, further development in the following areas is required:

7.2.5.1 Adapted low bit rate voice coding/decoding system

A custom built low bit rate voice coding/decoding system that uses a similar audio analysis as described in section 2.1 is required so that the cepstral parameters of the audio frames can be calculated directly from the coded audio parameters. The audio can be decoded and the lip shape can be determined in parallel on the receiver side.

7.2.5.2 Model-based head and face video coding system

A complete real-time model based head and face animation system (outlined in section 1.1.2.3) is currently being developed at SMRLab in the S.I.T.E. (School of Information Technologies and Engineering) at the University Of Ottawa. This system is required to complete the model-based audio/video coding system as outlined in section 1.1.2.

7.2.5.3 Integration of animated lip model with head model

The next step after the model based head and face model are up and running is to design a method of integrating the animated lip model and with the rest of the face completing the entire talking head avatar.

Bibliography

Abbs73 — Abbs H., Gilbert B.N., a strain guage transduction system for lip and jaw motion in two dimensions: Design criteria and calibration data. Journal of Speech and Hearing Research, 1973, 16, pp 248-256.

Abry94 — C. Abry, M.T. Lalouache (1994) Pour un modele d'anticipation dependant du locuteur- données sur l'arrondissement en francais. Bulletin de la communication parlée - ICP Grenoble.

Adjoudani97 — Adjoudani, A. (1997). Reconnaissance automatique de la parole audiovisuelle. PhD thesis, INPG, Grenoble.

Agelfors98 — Agelfors, Eva. Beskow, Jonas. Dahlquist, Martin. Granstrom, Bjorn. Lundeberg, Magnus. Spens, Karl-Erik. Ohman, Tobias. Teleface - the use of a synthetic face for the hard of hearing IEEE Workshop on Interactive Voice Technology for Telecommunications Applications, IVTTA 1998. IEEE, Piscataway, NJ, USA, 98TH8376.. p 130-134

Assaf97 — M.H. Assaf, M. Cordea, **M. Bondy**, C.M. Nafornita, E. Petriu, H.J.W. Spoelder, Image/Voice modeling and Synchronization for Model-Based Video-Telephony, Proc. ET & VS-IM/97 IEEE Workshop on Emergent Technol. and Virtual Systems for Instrum. Meas., pages 165-171, Niagara Falls, Ont., 1997.

Azencott97 — Azencott R, Younes L. An energy minimization method for matching and comparing structured object representations. Energy Minimization Methods in Computer Vision and Pattern Recognition. International Workshop EMMCVPR '97. Proceedings. Springer-Verlag. 1997, pp.441-56. Berlin, Germany.

Badin94 — Badin P, Motoki K, Miki N, Ritterhaus D, Lallouache M-T. Some geometric and acoustic properties of the lip horn. Journal of the Acoustical Society of Japan (E), vol.15, no.4, July 1994, pp.243-53. Japan.

Badin96 — Badin, Pierre. Abry, Christian. Articulatory synthesis from X-rays and inversion for an adaptive speech robot International Conference on Spoken Language Processing, ICSLP, Proceedings. v 2 1996. IEEE, Piscataway, NJ, USA, 96TH8208. p 1125-1128

Basu98 — Basu, Sumit. Oliver, Nuria. Pentland, Alex. 3D lip shapes from video: A combined physical-statistical model Speech Communication. v 26 n 1-2 Oct 1998. p 131-148

Benoit96 Benoit C. Synthesis and automatic recognition of audio-visual speech. IEE Colloquium on Integrated Audio-Visual Processing for Recognition, Synthesis and Communication (Ref. No.1996/213). IEE. 1996, pp.1/1-6. London, UK.

Benoit98 Benoit, Christian. Le Goff, Bertrand. Audio-visual speech synthesis from French text: Eight years of models, designs and evaluation at the ICP Speech Communication. v 26 n 1-2 Oct 1998. p 117-129

Berns96 Berns, Roy S. Methods for characterizing CRT displays Displays-Technology & Applications. v 16 n 4 May 1996. p 173-182

Blokland98 Blokland, Art. Anderson, Anne H. Effect of low frame-rate video on intelligibility of speech Speech Communication. v 26 n 1-2 Oct 1998. p 97-103

Bothe93 Bothe, Hans H. Rieger, Frauke. Visual speech and coarticulation effects Image and Multidimensional Signal Processing Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. v 5 1993. p V-634-V-637

Bothe94 Bothe HH, Wieden EA. Artificial visual speech synchronized with a speech synthesis system. Computers for Handicapped Persons. 4th International Conference, ICCHP '94 Proceedings. Springer-Verlag. 1994, pp.32-7. Berlin, Germany.

Bothe96 Bothe HH, Wieden EA. Phoneme to facial-movements transformation by classification of selected human lip shapes. IIA'96/SOCO'96. International ICSC Symposia on Intelligent Industrial Automation and Soft Computing. Int. Comput. Sci. Conventions. 1996, pp.C16-20. Millet, Alta., Canada.

Bouabana98 Bouabana, Soumya. Maeda, Shinji. Multi-pulse LPC modeling of articulatory movements Speech Communication. v 24 n 3 June 1998. p 227-248

Breen96 Breen, A P. Bowers, E. Welsh, W. Investigation into the generation of mouth shapes for a talking head International Conference on Spoken Language Processing, ICSLP, Proceedings. v 4 1996. IEEE, Piscataway, NJ, USA,96TH8206. p 2159-2162

Bregler97 Bregler C, Covell M, Slaney M. Video Rewrite: driving visual speech with audio. Computer Graphics Proceedings, SIGGRAPH 97. ACM. 1997, pp.353-60. New York, NY, USA.

Brezinski99 Brezinski, C. Multiparameter descent methods. Linear Algebra and its Applications, Volume: 296, Issue: 1-3, July 15, 1999, pp. 113-141.

Brooke94 Brooke NM, Scott SD. Computer graphics animations of talking faces based on stochastic models. ISSIPNN '94. 1994 International Symposium on Speech, Image Processing and Neural Networks Proceedings (Cat. No.94TH0638-7). IEEE. Part vol.1, 1994, pp.73-6 vol.1. New York, NY, USA.

Brooke96a Brooke, N M. Scott, S D. Tomlinson, M J. Making talking heads and speechreading with computers IEE Colloquium (Digest). n 213 Nov 28 1996. IEE, Stevenage, Engl. p 2/1-2/6

Brooke96b Brooke, N Michael. Using the visual component in automatic speech recognition International Conference on Spoken Language Processing, ICSLP, Proceedings. v 3 1996. IEEE, Piscataway, NJ, USA,96TH8206. p 1656-1659

Brooke96c Brooke NM, Petajan ED. Seeing speech: investigations into the synthesis and recognition of visible speech movements using automatic image processing and computer graphics. International Conference on Speech Input/Output; Techniques and Applications (Conf. Publ. No. 258). IEE. 1996, pp.104-9. London, UK.

Cathiard92 Cathiard MA, Cirot-Tseva A, Lallouache MT. Visual identification of protrusion and retraction gestures of the lips during acoustic pauses: performance of French and Greek subjects. Journal de Physique IV, vol.2, no.C1, pt.1, April 1992, pp.319-22. France.

Chan99 Chan, Syin. Ngo, Chong Wah. Lai, Kok F. Motion tracking of human mouth by generalized deformable models Pattern Recognition Letters. v 20 n 9 1999. p 879-887

Chen96 Chen, Tsuhan. Graf, Hans Peter. Haskell, Barry. Petajan, Eric. Wang, Yao. Chen, Homer. Chou, Wu. Speech-assisted lip synchronization in audio-visual communications IEEE International Conference on Image Processing. v 2 1996. IEEE, Los Alamitos, CA, USA,95CB35819. p 579-582

Childers95 Childers DG. Glottal source modeling for voice conversion. Speech Communication, vol.16, no.2, Feb. 1995, pp.127-38. Netherlands.

Choi00 Choi, Seung Ho. Kim, Hong Kook. Lee, Hwang Soo. Speech recognition using quantized LSP parameters and their transformations in digital communication Speech Communication. v 30 n 4 2000. p 223-233

Cohen93 Cohen MM, Massaro DW. Modeling coarticulation in synthetic visual speech. Models and Techniques in Computer Animation. Springer-Verlag. 1993, pp.139-56. Tokyo, Japan.

Cordea99 Cordea, M. Real Time 3D Head Pose Recovery for Model Based Video Coding, Master thesis, SITE, University Of Ottawa, 1999.

Cosi96 Cosi, P. Dugatto, M. Ferrero, F. Caldognetto, E Magno. Vagges, K. Phonetic recognition by recurrent neural networks working on audio and visual information Speech Communication. v 19 n 3 Sep 1996. p 245-252

Coughlan00 Coughlan, James. Yuille, Alan. English, Camper. Snow, Dan. Efficient deformable template detection and localization without user initialization Computer Vision & Image Understanding. v 78 n 3 2000. p 303-319

Cranen96 Cranen, Bert. Schroeter, Juergen. Physiologically motivated modelling of the voice source in articulatory analysis/synthesis Speech Communication. v 19 n 1 Jul 1996. p 1-19

Dai96 Dai, Ying. Nakano, Yasuaki. Face-texture model based on SGLD and its application in face detection in a color scene Pattern Recognition. v 29 n 6 Jun 1996. p 1007-1017

Deng00 Deng, Jing; Newton, Nina M; Hall-Craggs, Margaret A; Shirley, Rebecca A; Linney, Alfred D; Lees, William R; Rodeck, Charles H; McGrouther, Duncan. A Novel technique for three-dimensional visualisation and quantification of deformable, moving soft-tissue body parts. The Lancet, Vol: 356, Issue: 9224, pp. 127-131, July 8, 2000

Escolano97 Escolano F, Cazorla M, Gallardo D, Rizo R. Deformable templates for tracking and analysis of intravascular ultrasound sequences. Energy Minimization Methods in Computer Vision and Pattern Recognition. International Workshop EMMCVPR '97. Proceedings. Springer-Verlag. 1997, pp.521-34. Berlin, Germany.

Esme96 Esme B, Sankur B, Anarim E. Facial feature extraction using genetic algorithms. Signal Processing VIII, Theories and Applications. Proceedings of EUSIPCO-96, Eighth European Signal Processing Conference. Edizioni LINT Trieste. Part vol.3, 1996, pp.1511-14 vol.3. Trieste, Italy.

Ezzat00 Ezzat T, Poggio T. Visual speech synthesis by morphing visemes. International Journal of Computer Vision, vol.38, no.1, 2000, pp.45-57. Publisher: Kluwer Academic Publishers, Netherlands.

Fabian97 Fabian, V. Simulated annealing simulated Computers & Mathematics with Applications. v 33 n 1-2 Jan 1997. p 81-94

Faruquie00 Faruquie TA, Neti C, Rajput N, Subramaniam LV, Verma A. Translingual visual speech synthesis. 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532). IEEE. Part vol.2, 2000, pp.1089-92 vol.2. Piscataway, NJ, USA.

Figueiredo97 Figueiredo MAT, Leitao JMN, Jain AK. Adaptive parametrically deformable contours. Energy Minimization Methods in Computer Vision and Pattern Recognition. International Workshop EMMCVPR '97. Proceedings. Springer-Verlag. 1997, pp.35-50. Berlin, Germany.

Finn88 Finn KE, Montgomery AA. Automatic optically-based recognition of speech. Pattern Recognition Letters, vol.8, no.3, Oct. 1988, pp.159-64. Netherlands.

Flanagan70 J.L. Flanagan ,C.H. Coker, L.R. Rabiner, R.W. Schafer, and N. Umeda, Synthetic Voices for Conmputers, IEEE Spectrum, 7 (10): 22-45, October 1970.

Flanagan72 J.L. Flanagan, Speech analysis, synthesis, and perception, 2nd ed, Springer-Verlag, New York, 1972.

Fort96 Fort, A. Ismaelli, A. Manfredi, C. Bruscaglioni, P. Parametric and non-parametric estimation of speech formants: application to infant cry Medical Engineering & Physics. v 18 n 8 Dec 1996. p 677-691

Fort98 Fort, A. Manfredi, C. Acoustic analysis of newborn infant cry signals Medical Engineering & Physics. v 20 n 6 Sep 1998. p 432-442

Franke00 Franke, U. Joos, A. Real-time stereo vision for urban traffic scene understanding IEEE Intelligent Vehicles Symposium, Proceedings 2000. IEEE, Piscataway, NJ, USA,00TH8511.. p 273-278

Gray00 Gray, Robert M. FUNDAMENTALS OF VECTOR QUANTIZATION. Available from IEEE Service Cent (Cat. Publ by IEEE, New York, NY, USA. n 87CH2423-2), Piscataway, NJ, USA, Oct 2000 p 1262-1271

Guiard-Marigny96 Guiard-Marigny T, Tsingos N, Adjoudani A, Benoit C, Gascuel M-P. 3D models of the lips for realistic speech animation. Proceedings. Computer Animation '96. IEEE Comput. Soc. Press. 1996, pp.80-9. Los Alamitos, CA, USA.

Hani98 Hani Yehia, Rubin P, Vatikiotis-Bateson E. Quantitative association of vocal-tract and facial behavior. Speech Communication, vol.26, no.1-2, Oct. 1998, pp.23-43. Publisher: Elsevier, Netherlands.

Hara00 Hara F, Endo K. Dynamic control of lip-configuration of a mouth robot for Japanese vowels. Robotics & Autonomous Systems, vol.31, no.3, 31 May 2000, pp.161-9. Publisher: Elsevier, Netherlands.

Hennecke94 Hennecke ME, Prasad KV, Stork DG. Using deformable templates to infer visual speech dynamics. Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers (Cat. No.94CH34546). IEEE Comput. Soc. Press. Part vol.1, 1994, pp.578-82 vol.1. Los Alamitos, CA, USA.

Hennecke96 Hennecke ME, Prasad KV, Stork DG. Automatic speech recognition system using acoustic and visual signals. Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers. IEEE Comput. Soc. Press. Part vol.2, 1996, pp.1214-18 vol.2. Los Alamitos, CA, USA.

Howing99 Howing F, Dooley LS, Wermser D. Tracking of non-rigid articulatory organs in X-ray image sequences. Computerized Medical Imaging & Graphics, vol.23, no.2, March-April 1999, pp.59-67. Publisher: Elsevier, UK.

Huitao00 Huitao Luo, Eleftheriadis A, Kouloheris J. Statistical model-based video segmentation and its application to very low bit-rate video coding. Signal Processing: Image Communication, vol.16, no.3, Oct. 2000, pp.333-52. Publisher: Elsevier, Netherlands.

Ip96 Ip, Horace H S. Chan, C S. Script-based facial gesture and speech animation using a NURBS based face model Computers & Graphics. v 20 n 6 Nov-Dec 1996. p 881-891

Jain96 Jain, Anil K. Zhong, Yu. Lakshmanan, Sridhar. Object matching using deformable templates IEEE Transactions on Pattern Analysis & Machine Intelligence. v 18 n 3 Mar 1996. p 267-278

Jain98 Jain, Anil K. Zhong, Yu. Dubuisson-Jolly, Marie-Pierre. Deformable template models: A review Signal Processing. v 71 n 2 Dec 1998. p 109-129

Jones97 Jones, Jeffery A. Munhall, Kevin G. Effects of separating auditory and visual sources on audiovisual integration of speech Canadian Acoustics - Acoustique Canadienne. v 25 n 4 Dec 1997. p 13-19

Jourlin96 Jourlin P. Handling disynchronization phenomena with HMM in connected speech. Signal Processing VIII, Theories and Applications. Proceedings of EUSIPCO-96, Eighth European Signal Processing Conference. Edizioni LINT Trieste. Part vol.1, 1996, pp.133-6 vol.1. Trieste, Italy.

Jourlin98 Jourlin P. (1998). Approche bimodale du traitement automatique de la parole: application a la reconnaissance du message et du locuteur, PhD thesis, Université d'avignon, aix-marseille.

Jyh-Yuan97 Jyh-Yuan Deng, Feipei Lai. Region-based template deformation and masking for eye-feature extraction and description. Pattern Recognition, vol.30, no.3, March 1997, pp.403-19. Publisher: Elsevier, UK.

Kapfer97 Kapfer, M. Benois-Pineau, J. Detection of human faces in color image sequences with arbitrary motions for very low bit-rate videophone coding Pattern Recognition Letters. v 18 n 14 Dec 1997. p 1503-1518

Karunaratne99 Karunaratne SK, Hong Yan. An emotional viseme compiler for facial animation. ISSPA '99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No.99EX359). Queensland Univ. Technol. Part vol.1, 1999, pp.459-61 vol.1. Brisbane, Qld., Australia.

Kaucic96 Kaucic, Robert. Reynard, David. Blake, Andrew. Real-time lip trackers for use in audio-visual speech recognition IEE Colloquium (Digest). n 213 Nov 28 1996. IEE, Stevenage, Engl. p 3/1-3/6

Kervrann97 Kervrann, C. Davoine, F. Perez, P. Forchheimer, R. Labit, C. Generalized likelihood ratio-based face detection and extraction of mouth features Pattern Recognition Letters. v 18 n 9 Sep 1997. p 899-912

Kim00a Kim, Jong Il. Bovik, Alan C. Evans, Brian L. Generalized predictive binary shape coding using polygon approximation Signal Processing: Image Communication. v 15 n 7 2000. p 643-663

Kim00b Kim, Hong Kook. Choi, Seung Ho. Lee, Hwang Soo. On approximating line spectral frequencies to LPC cepstral coefficients IEEE Transactions on Speech & Audio Processing. v 8 n 2 2000. p 195-199

Kober94 Kober, R. Schiffers, J. Schmidt, K. Model-based versus knowledge-guided representation of non-rigid objects: a case study IEEE International Conference on Image Processing. v 1 1994. IEEE, Los Alamitos, CA, USA,94CH35708. p 973-977

Kovesi99 Kovesi, Balazs. Saoudi, Samir. Boucher, Jean Marc. Horvath, Gabor. Real time vector quantization of LSP parameters Speech Communication. v 29 n 1 Sep 1999. p 39-47

Kreyszig88 Kreyszig, Erwin, Advanced Engineering Mathematics, John Wiley & Sons Inc. 1988.

Kunt88 Kunt, Murat. Hugli, Heinz. OVERVIEW OF DIGITAL TECHNIQUES FOR PROCESSING SPEECH SIGNALS. NATO ASI Series, Series F: Computer and Systems Sciences. v 16. Publ by Springer-Verlag, Berlin, West Ger and New York, NY, USA, 1988, p 1-71

Lagana96 Lagana, A. Lavagetto, F. Storace, A. Visual synthesis of source acoustic speech through Kohonen neural networks International Conference on Spoken Language Processing, ICSLP, Proceedings. v 4 1996. IEEE, Piscataway, NJ, USA,96TH8206. p 2183-2186

Lallouache88 Lallouache M-T, Worley C. Acquisition, editing and processing of video and articulatory images and signals for lip and jaw movements. Journal D'Acoustique, vol.1, no.3, Sept. 1988, pp.215-20. France.

Lallouache91 Lallouache M.T. Un poste visage-parole couleur. Acquisition et traitement automatique des contours des levres. These de doctorat, Systemes Electroniques, Institut National Polytechnique de Grenoble, France. 1991.

Lavagetto96 Lavagetto, F. Pandzic, I S. Kalra, P. Thalmann, N Magnenat. Synthetic and hybrid imaging in the humanoid and vidas projects IEEE International Conference on Image Processing. v 3 1996. IEEE, Los Alamitos, CA, USA,96CH35919. p 663-666

Lee96 Lee, Youngjik. Hwang, Kyu-Woong. Selecting good speech features for recognition Etri Journal. v 18 n 1 Apr 1996. p 29-40

Lepsoy98 Lepsoy, Skjalg. Curinga, Sergio. Conversion of articulatory parameters into active shape model coefficients for lip motion representation and synthesis Signal Processing: Image Communication. v 13 n 3 Sep 1998. p 209-225

Lewis91 Lewis J. Automated lip-sync: background and techniques. Journal of Visualization & Computer Animation, vol.2, no.4, Oct.-Dec. 1991, pp.118-22. UK.

Li95 Li, Xiaobo. Roeder, Nicholas. Face contour extraction from front-view images Pattern Recognition. v 28 n 8 Aug 1995. p 1167-1179

Lindblom71 Lindblom B. Sundberg J. Acoustical consequences of lip, tongue, jaw and larynx movement. Journal of the Acoustical Society of America, 50:1166-1179.

Liu99 Liu Y, Yu L, Yao Q. Automatic extraction of facial features using deformable templates. Picture Coding Symposium '99. Oregon State Univ. 1999, pp.189-92. Corvallis, OR, USA.

Luettin96a Luettin, Juergen. Thacker, Neil A. Beet, Steve W. Speechreading using shape and intensity information International Conference on Spoken Language Processing, ICSLP, Proceedings. v 1 1996. IEEE, Piscataway, NJ, USA,96TH8206. p 58-61

Luettin96b Luettin, Juergen. Thacker, Neil A. Beet, Steve W. Visual speech recognition using active shape models and hidden Markov models ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. v 2 1996. IEEE, Piscataway, NJ, USA,96CB35903. p 817-820

Luo94 Luo, S -H. King, R W. Novel approach for classifying continuous speech into visible mouth-shape related classes Proceedings - ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing. v 1 1994. IEEE, Piscataway, NJ, USA,94CH3387-8. p I-465-468

Markel72 Markel JD. The SIFT algorithm for fundamental frequency estimation. IEEE Transactions on Audio & Electroacoustics, vol.Au-20, no.5, Dec. 1972, pp.367-77. USA.

Masuko98 Masuko, Takashi. Kobayashi, Takao. Tamura, Masatsune. Masubuchi, Jun. Tokuda, Keiichi. Text-to-visual speech synthesis based on parameter generation from HMM ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. v 6 1998. IEEE, Piscataway, NJ, USA,98CH36181. p 3745-3748

McAllister98 McAllister, David F. Rodman, Robert D. Bitzer, Donald L. Freeman, Andrew S. Speaker independence in automated lip-sync for audio-video communication Computer Networks & Isdn Systems. v 30 n 20-21 Nov 12 1998. p 1975-1980

McKerrow91 Phillip John McKerrow, Introduction to Robotics, p. 611, 1991.

Mirhosseini98 Mirhosseini AR, Hong Yan, Kin-Man Lam. Adaptive deformable model for mouth boundary detection. Optical Engineering, vol.37, no.3, March 1998, pp.869-75. Publisher: SPIE, USA.

Montgomery86 Montgomery AA, Finn KE. The use of visible lip information in automatic speech recognition. Signal Processing III: Theories and Applications. Proceedings of EUSIPCO-86: Third European Signal Processing Conference. North-Holland. 1986, pp.577-80 vol.1. Amsterdam, Netherlands.

Moore88 Moore, Roger K. SYSTEMS FOR ISOLATED AND CONNECTED WORD RECOGNITION. NATO ASI Series, Series F: Computer and Systems Sciences. v 16. Publ by Springer-Verlag, Berlin, West Ger and New York, NY, USA 1988, p 73-143

Morishima93 Morishima, Shigeo. Harashima, Hiroshi. Facial expression synthesis based on natural voice for virtual face-to-face communication with machine 1993 IEEE Annual Virtual Reality International Symposium 1993 IEEE Annu Virtual Reality Int Symp. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA,(IEEE cat n 93CH3336-5). 1993. p 486-491

Morishima99 Morishima, Shigeo. Real-time voice driven facial animation system Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. v 6 1999. IEEE, USA. p VI-59 - VI-63

Musmann95 Musmann HG. A layered coding system for very low bit rate video coding. Signal Processing: Image Communication, vol.7, no.4-6, Nov. 1995, pp.267-78. Publisher: Elsevier, Netherlands.

Ng93 Ng Vincent, Freeman George. AGES: Articulator Gesture Emulation System. Hardware Interfaces, 1993. p. 114-117

Ngan96 Ngan, King N. Rudianto, Raymond L. Automatic face location detection and tracking for model-based video coding International Conference on Signal Processing Proceedings, ICSP. v 2 1996. IEEE, Piscataway, NJ, USA,96TH8116. p 1098-1101

Olives99 Olives J-L, Sams M, Kulju J, Seppala O, Karjalainen M, Altosaar T, Lemmetty S, Toyra K, Vainio M. Towards a high quality Finnish talking head. 1999 IEEE Third Workshop on Multimedia Signal Processing (Cat. No.99TH8451). IEEE. 1999, pp.433-7. Piscataway, NJ, USA.

Painter96 Painter, Edward. Spanias, Andreas. MATLAB software tool for the introduction of speech coding fundamentals in a DSP course Technology-Based Re-Engineering Engineering Education Proceedings - Frontiers in Education Conference. v 2 1996. IEEE, Piscataway, NJ, USA,96CB35946. p 603-608

Pelachaud96 Pelachaud C, Badler NI, Steedman M. Generating facial expressions for speech. Cognitive Science, vol.20, no.1, Jan.-March 1996, pp.1-46. Publisher: Ablex Publishing, USA.

Perkell92 Perkell JS, Cohen MH, Svirsky MA, Matthies ML, Garabieta I, Jackson MTT. Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements. Journal of the Acoustical Society of America, vol.92, no.6, Dec. 1992, pp.3078-96. USA.

Petajan00 Petajan E. Approaches to visual speech processing based on the MPEG-4 Face Animation standard. 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532). IEEE. Part vol.1, 2000, pp.575-8 vol.1. Piscataway, NJ, USA.

Petajan96 Petajan E, Graf HP. Robust face feature analysis for automatic speechreading and character animation. Proceedings of the Second International Conference on Automatic Face and Gesture Recognition (Cat. No.96TB100079). IEEE Comput. Soc. Press. 1996, pp.357-62. Los Alamitos, CA, USA.

Petzold99 Petzold, Charles, Programming Windows, Microsoft Press, 1999.

Pham98 Pham TD, Wagner M. A geostatistical model for linear prediction analysis of speech. Pattern Recognition, vol.31, no.12, Dec. 1998, pp.1981-91. Publisher: Elsevier, UK.

Rabi97 Rabi G, Si Wei Lu. Energy minimization for extracting mouth curves in a facial image. Proceedings. Intelligent Information Systems. IIS'97 (Cat. No.97TB100201). IEEE Comput. Soc. 1997, pp.381-5. Los Alamitos, CA, USA.

Rabiner77 Rabiner LR, Atal BS, Sambur MR. LPC prediction error-analysis of its variation with the position of the analysis frame. IEEE Transactions on Acoustics, Speech, & Signal Processing, vol.ASSP-25, no.5, Oct. 1977, pp.434-42. USA.

Rabiner81 Rabiner LR, Levinson SE. Isolated and connected word recognition-theory and selected applications. IEEE Transactions on Communications, vol.COM-29, no.5, May 1981, pp.621-59. USA.

Rabiner93 L. Rabiner, B.H. Juang, Fundamentals of speech recognition. Alan V. Oppenheim series editor, 507 p. 1993.

Rabiner95 Rabiner, L R. Impact of voice processing on modern telecommunications Speech Communication. v 17 n 3-4 Nov 1995. p 217-226

Rao94 Rao, Ram R. Mersereau, Russell M. Lip modeling for visual speech recognition Conference Record of the Asilomar Conference on Signals, Systems & Computers. v 1 1994. IEEE, Los Alamitos, CA, USA,94CH34546. p 587-590

Rao95 Rao RR, Mersereau RM. On merging hidden Markov models with deformable templates. Proceedings. International Conference on Image Processing (Cat. No.95CB35819). IEEE Comput. Soc. Press. Part vol.3, 1995, pp.556-9 vol.3. Los Alamitos, CA, USA.

Rogozan98 Rogozan, Alexandrina. Deleglise, Paul. Adaptive fusion of acoustic and visual sources for automatic speech recognition Speech Communication. v 26 n 1-2 Oct 1998. p 149-161

Rouat97 Rouat J, Liu YC, Morisette D. A pitch determination and voiced/unvoiced decision algorithm for noisy speech. Speech Communication, vol.21, no.3, April 1997, pp.191-207. Publisher: Elsevier, Netherlands.

Sahandi98 Sahandi R, Vine DSG, Longster J. Text-to-visual speech synthesis. Informatica (Ljubljana), vol.22, no.4, Dec. 1998, pp.445-50. Publisher: Slovene Soc. Informatika, Slovenia.

Saji97 Saji H, Nakatani H. Deformable templates for tracking the facial components. Proceedings. 6th IEEE International Workshop on Robot and Human Communication. RO-MAN '97 Sendai (Cat. No.97TH8296). IEEE. 1997, pp.364-7. New York, NY, USA.

Sakalli98 Sakalli M, Yan H. Feature-based compression of human face images. Optical Engineering, vol.37, no.5, May 1998, pp.1520-9. Publisher: SPIE, USA.

Scales85 Scales L. E. Introduction to non-linear optimization. New York, Springer-Verlag, 1985.

Schroeder85 Schroeder MR. Linear predictive coding of speech: review and current directions. IEEE Communications Magazine, vol.23, no.8, Aug. 1985, pp.54-61. USA.

Simon99 Simon, Richard, Windows NT Win32 API Superbible. Waite Group Press, 1999.

Smith96 Smith Ray Alvy, Iyons Ray Eric. HWB-A more intuitive hue-based color

model. Journal of Graphics Tools, 1(1):3-17, 1996.

Sobottka98 Sobottka K, Pitas I. A novel method for automatic face segmentation, facial feature extraction and tracking. Signal Processing: Image Communication, vol.12, no.3, June 1998, pp.263-81. Publisher: Elsevier, Netherlands.

Storey88 Storey D, Roberts M. Reading the speech of digital lips: motives and methods for audio-visual speech synthesis. Visible Language, vol.22, no.1, Winter 1988, pp.112-27. USA.

Sungyun98 Sungyun Wee, Seunghwan Ji, Changyong Yoon, Mignon Park. Face detection using pattern information and deformable template in motion images. Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems. Methodologies for the Conception, Design and Application of Soft Computing. World Scientific. Part vol.1, 1998, pp.213-16 vol.1. Singapore.

Swenson98 Swenson RL, Dimond KR. A universal colour transformation architecture. Pattern Recognition Letters, vol.19, no.9, July 1998, pp.805-13. Publisher: Elsevier, Netherlands.

Tawfik99 Tawfik, Youssef S. Darwish, Ahmed M. Shaheen, Samir I. Energy matching based on deformable templates Proceedings - Icassp, IEEE International Conference on Acoustics, Speech & Signal Processing. v 6 1999. p 3481-3484

Tomlinson96 Tomlinson, M J. Russell, M J. Brooke, N M. Integrating audio and visual information to provide highly robust speech recognition ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings. v 2 1996. IEEE, Piscataway, NJ, USA,96CB35903. p 821-824

Tsallis96 Tsallis, C. Stariolo, D A. Generalized simulated annealing Physica A. v 233 n 1-2 Nov 15 1996. p 395-406

Wang00 Wang Zhan, Huangfu, Wan Jian-Wei. Human face feature extraction using deformable templates. Journal of Computer Aided Design & Computer Graphics, vol.12, no.5, May 2000, pp.333-6. Publisher: Science Press, China.

Williams00 Williams JJ, Katsaggelos AK, Randolph MA. A hidden Markov model based visual speech synthesizer. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100). IEEE. Part vol.4, 2000, pp.2393-6 vol.4. Piscataway, NJ, USA.

Woodard97 Woodard JP, Hanzo L. A range of low and high delay CELP speech codecs between 8 and 4 kbits/s. Digital Signal Processing: a Review Journal, vol.7, no.1, Jan. 1997, pp.37-46. Publisher: Academic Press, USA.

Xie94 Xie X, Sudhakar R, Zhuang H. On improving eye feature extraction using deformable templates. Pattern Recognition, vol.27, no.6, June 1994, pp.791-9. UK.

Xu98 Xu, Chenyang. Prince, Jerry L. Generalized gradient vector flow external forces for active contours Signal Processing. v 71 n 2 Dec 1998. p 131-139

Yamamoto98 Yamamoto, E. Nakamura, S. Shikano, K. Lip movement synthesis from speech based on Hidden Markov Models Speech Communication. v 26 n 1-2 Oct 1998. p 105-115

Yang00 Yang J, Xiao J, Ritter M. Automatic selection of visemes for image-based visual speech synthesis. 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532). IEEE. Part vol.2, 2000, pp.1081-4 vol.2. Piscataway, NJ, USA.

Yu99 Yu, Keren. Jiang, Xiaoyi. Bunke, Horst. Lipreading using signal analysis over time Signal Processing. v 77 n 2 1999. p 195-208

Yuille00 Yuille, A L. Coughlan, James M. A* perspective on deterministic optimization for deformable templates Pattern Recognition. v 33 n 4 2000. p 603-616

Yuille92 Yuille, Alan L. Hallinan, Peter W. Cohen, David S. Feature extraction from faces using deformable templates International Journal of Computer Vision. v 8 n 2 Aug 1992. p 99-111

Zhang97a Zhang, Liang. Estimation of the mouth features using deformable templates IEEE International Conference on Image Processing. v 3 1997. IEEE Comp Soc, Los Alamitos, CA, USA, 97CB36144. p 328-331

Zhang97b Zhang, Liang. Tracking a face for knowledge-based coding of videophone sequences Signal Processing: Image Communication. v 10 n 1-3 Jul 1997. p 93-114

Zhang98 Zhang, Liang. Automatic adaptation of a face model using action units for semantic coding of videophone sequences IEEE Transactions on Circuits & Systems for Video Technology. v 8 n 6 Oct 1998. p 781-795

Zhong98 Zhong, Jialin. Flexible face animation using MPEG-4/SNHC parameter streams IEEE International Conference on Image Processing. v 2 1998. IEEE Comp Soc, Los Alamitos, CA, USA, 98CB36269. p 924-928

Appendix 1

The annex following was taken from the Help file of the MELP (Mixed-Excitation Linear Predictive) speech coder demonstration program. Created by Atlanta Signal Processors.

Street Address: Atlanta Signal Processors, Inc.
1375 Peachtree St. NE, Suite 690
Atlanta, GA 30309-3115
Phone Number: (404) 892-7265
email: info@aspi.com
Web: http://www.aspi.com

MELP General Description

Atlanta Signal Processors' MELP (Mixed-Excitation Linear Predictive) coder is a high-quality, very low bit-rate speech coder algorithm. It offers high speech quality at a data rate of 2,400 bps. In formal listening tests, the MELP coder has been shown to be clearly superior to similar LPC Vocoders with a binary voicing decision and no mixed-excitation. The speech quality of the MELP coder, operating at 2,400 bps, is equivalent to that of the Department of Defense (DoD) 4,800 bps CELP standard.

MELP Algorithm Description

Traditional pitched-excited LPC vocoders use either a periodic pulse train or white noise as the excitation for an all-pole synthesis filter. These vocoders produce intelligible speech at very low bit rates, but they sometimes sound mechanical or buzzy and are prone to annoying thumps and tonal noises. These problems arise from the inability of a simple pulse train to reproduce all kinds of voiced speech. The MELP coder uses a mixed-excitation model that can produce more natural sounding speech because it can represent a richer ensemble of possible speech characteristics. The MELP coder is also robust in difficult background noise environments such as those frequently encountered in commercial and military communication systems.

The MELP coder is based on the traditional LPC parametric model, but also includes four additional features. These are mixed-excitation, aperiodic pulses, pulse dispersion, and adaptive spectral enhancement.

The mixed-excitation is implemented using a multi-band mixing model. This model can simulate frequency dependent voicing strength using a novel adaptive filtering structure based on a fixed filterbank. The primary effect of this multi-band mixed-excitation is to reduce the buzz usually associated with LPC vocoders, especially in broadband acoustic noise.

When the input speech is voiced, the MELP vocoder can synthesize speech using either periodic or aperiodic pulses. Aperiodic pulses are most often used during transition regions between voiced and unvoiced segments of the speech signal. This

feature allows the synthesizer to reproduce erratic glottal pulses without introducing tonal noises.

The pulse dispersion is implemented using fixed pulse dispersion filter based on a spectrally flattened triangle pulse. This filter has the effect of spreading the excitation energy within a pitch period. This, in turn, reduces the harsh quality of the synthetic speech.

The adaptive spectral enhancement filter is based on the poles of the LPC vocal tract filter and is used to enhance the formant structure in the synthetic speech. This filter improves the match between synthetic and natural bandpass waveforms, and introduces a more natural quality to the speech output.

The first ten Fourier magnitudes are obtained by picking peaks in the FFT of the residual signal. The information embodied in these coefficients improves the accuracy of the speech production model at the perceptually important lower frequencies. This increases the quality of the coded speech, particularly for males and in the presence of background noise.

In formal listening tests, the 2,400 bps MELP coder received a Diagnostic Acceptability Score which was ten points higher than the DoD standard (LPC-10e) operating at the same rate, and which was statistically indistinguishable from that of the DoD FS-1016 CELP standard operating at 4,800 bps.

MELP Specifications

Data rate: 2,400 bps
Sampling rate: 8 kHz
Signal input: 16-bit linear
Bit stream format: For each 22.5 mS frame of input speech, the following 54 bits are placed into the bit-stream:

Description	Number of bits
Pitch index	7
Jitter flag	1
Bandpass voicing decision	4x1
Gain for second half of frame	5
Gain for first half of frame	3
LSP frequencies (10 line spectrum pairs)	25
Fourier magnitudes (10 harmonics)	8
Sync bit	1
Total	54

The speech is broken down into frames of 180 samples (44.444 frames per second). The bit rate is given by: 54 * 44.4444 = 2,400 bps.