

REALISTIC SPEECH ANIMATION OF  
SYNTHETIC FACES

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER  
ENGINEERING AND INFORMATION SCIENCE  
AND THE INSTITUTE OF ENGINEERING AND SCIENCE  
OF BILKENT UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE

By

Barış Uz

June, 1998

TR  
897.7  
.493  
1998

# REALISTIC SPEECH ANIMATION OF SYNTHETIC FACES

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER  
ENGINEERING AND INFORMATION SCIENCE  
AND THE INSTITUTE OF ENGINEERING AND SCIENCE  
OF BILKENT UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE

By

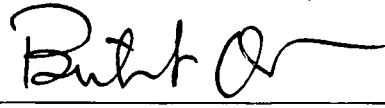
Bariş Uz

June, 1998

TR  
897.7  
.U93  
1998

B042642

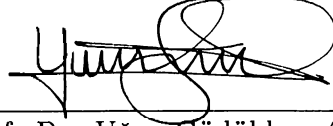
I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



---

Prof. Dr. Bülent Özgüç (Supervisor)

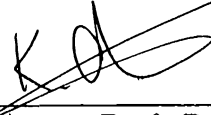
I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



---

Asst. Prof. Dr. Uğur Güdükbay (Co-supervisor)

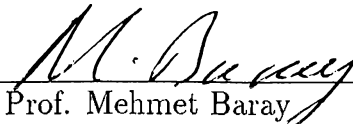
I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



---

Assoc. Prof. Dr. Kemal Oflazer

Approved for the Institute of Engineering and Science:



---

Prof. Mehmet Baray  
Director of Institute of Engineering and Science

# ABSTRACT

## REALISTIC SPEECH ANIMATION OF SYNTHETIC FACES

Barış Uz

M.S. in Computer Engineering and Information Science

Supervisors: Prof. Dr. Bülent Özgüç

and Asst. Prof. Dr. Uğur Güdükbay

June, 1998

In this study, physically-based modeling and parameterization are combined to generate realistic speech animation on synthetic faces. Physically-based modeling is used for muscles which are modeled as forces deforming the mesh of polygons. Parameterization technique is used for generating mouth shapes for speech animation. Each meaningful part of a text, a letter in our case, corresponds to a specific mouth shape, and this shape is generated by setting a set of parameters used for representing the muscles and jaw rotation. A mechanism has also been developed to generate and synchronize facial expressions while speaking. Tags specifying facial expressions are inserted into the input text together with the degree of the expression. In this way, the facial expression with the specified degree is generated and synchronized with speech animation.

"

*Key words:* facial animation, speech animation, muscle-based, physically-based, facial expression.

# ÖZET

## SENTETİK İNSAN YÜZÜ İÇİN GERÇEĞE UYGUN KONUŞMA CANLANDIRMASI

Barış Uz

Bilgisayar ve Enformatik Mühendisliği, Yüksek Lisans

Tez Yöneticileri: Prof. Dr. Bülent Özgüç

ve Yrd. Doç. Dr. Uğur Güdükbay

Haziran, 1998

Bu çalışmada, sentetik insan yüzlerinde gerçekçi konuşma animasyonu için fiziğe dayalı modelleme ve parametrik yaklaşım birleştirilmiştir. Yüzdeki kaslar için fiziksel modelleme kullanılmıştır. Kaslar, poligonlardan oluşan bir ağ yapısını deforme eden kuvvetler olarak modellenmiştir. Metnin her “anlamli” parçası (bu çalışmada bir harf), belli bir ağız şekline karşı gelmektedir ve ağız şekilleri kasları ve çene hareketini etkileyen bir dizi parametrenin değiştirilmesi ile oluşturulmuştur. Ayrıca, yüz ifadelerini konuşma ile eşzamanlı kılacak bir yöntem geliştirilmiştir. Metinde bazı yerlere yüz ifadelerini ve derecelerini gösteren özel etiketler yerleştirilebilir ve böylece, yüz ifadesi oluşturulup konuşma animasyonu ile eşzamanlı yapılabilir.

*Anahtar kelimeler:* Yüz animasyonu, konuşma animasyonu, kas tabanlı, fiziğe dayalı, yüz ifadesi.

To my family who brought me to today,  
and to her who will take me to tomorrow...

## ACKNOWLEDGMENTS

I am very grateful to my supervisors Prof. Bülent Özgüç and Asst. Prof. Uğur Güdükbay not only for their invaluable guidance and motivating support in all steps of my study but also their instructions, advises and their excellent supervision and encouragement during and after the study.

I would like to thank Assoc. Prof. Kemal Oflazer especially for his contributions to natural language part of this study and his comments and remarks on the thesis.

I would like to thank Keith Waters who gave permission for the usage of his basic facial software.

I would like to thank Ümit V. Çatalyürek for his patience during my system problems and Tolga Ekmekçi for his patience and assistance during the production and post-production of the sample video.

I am also grateful to all of my friends who provide moral support during my long study. I would like to thank to all who has a contribution to my thesis in one way or another through formal and informal discussions.

Very special thanks to Berna for her moral support, motivation and comments. She always supports me against all kinds of problems that I have faced. I am grateful to her for being with me, today and tomorrow.

I have a deep gratitude to my family for everything they did for my being where I am today and for their invaluable support all throughout my life.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related Work</b>	<b>5</b>
2.1	Facial Animation . . . . .	5
2.2	Speech Animation . . . . .	7
<b>3</b>	<b>Structure of the Face</b>	<b>10</b>
3.1	Skin . . . . .	10
3.2	Muscles	12
3.2.1	Muscles of the Face . . . . .	13
3.3	Mouth and Jaw . . . . .	15
3.4	Teeth . . . . .	16
3.5	Tongue . . . . .	16
3.6	Eyes . . . . .	17
3.7	Other Important Details of the Face . . . . .	17

<i>CONTENTS</i>	viii
<b>4 Modeling in the System</b>	<b>19</b>
4.1 Overview of the Original Face Model . . . . .	20
4.2 Mouth (Lips) and Jaw . . . . .	21
4.3 Facial Muscles	22
4.3.1 Modeling of Facial Muscles . . . . .	22
4.3.2 Skin Deformations due to Muscle Actions . . . . .	24
4.4 Eyes and Eyebrows . . . . .	28
4.5 Teeth . . . . .	29
4.6 Tongue . . . . .	30
4.6.1 Assembling the Tongue	32
4.7 Summary	35
<b>5 Facial Animation</b>	<b>36</b>
5.1 Overview . . . . .	36
5.2 Expressions and Overlays	37
5.2.1 Expression Overlays	38
5.2.2 Eye Actions . . . . .	39
<b>6 Linguistic Issues</b>	<b>41</b>
6.1 Properties of Turkish Included in the System . . . . .	41
6.2 Coarticulation . . . . .	43

<i>CONTENTS</i>	ix
<b>7 Speech Animation System</b>	<b>45</b>
7.1 Synchronizing Speech with Expressions	45
7.1.1 Guessing Expressions from the Text . . . . .	46
7.1.2 Using Tags for Synchronization . . . . .	47
7.2 Input Text . . . . .	50
7.3 Database	51
7.4 Input Text Parser . . . . .	52
7.5 Facial Animation Display System . . . . .	54
7.6 Animation . . . . .	55
7.6.1 Keyframing . . . . .	55
7.6.2 Interpolation Techniques for Keyframed Animation . . .	55
7.7 Implementation . . . . .	56
7.7.1 Performance Issues . . . . .	57
<b>8 Results</b>	<b>59</b>
<b>9 Conclusions</b>	<b>65</b>
<b>A Implementation</b>	<b>67</b>
A.1 Main Data Structures . . . . .	68
A.2 Operation Flow . . . . .	71
A.3 Explanation of Modules	74
A.4 Database and Structure of Files . . . . .	75

# List of Figures

3.1	Visco elastic behavior of the skin. . . . .	12
4.1	Regions of the face. . . . .	20
4.2	Facial muscles in the model. . . . .	23
4.3	Location of facial muscles on the face model. . . . .	23
4.4	Parameters of a muscle. . . . .	26
4.5	Abstraction of <i>Orbicularis Oris</i> . . . . .	27
4.6	The eye model. . . . .	29
4.7	A tooth model. . . . .	29
4.8	Teeth model. . . . .	30
4.9	Tongue model from different views. . . . .	31
4.10	The parameters of tongue. . . . .	31
4.11	A section of tongue model. . . . .	32
7.1	The facial animation system . . . . .	46
7.2	The algorithm for speech animation. . . . .	52
8.1	Still frames from the animation sequence of the example. . . . .	60

8.2	Still frames from the animation sequence of the example (continued.)	61
8.3	Still frames from the animation sequence of the example (continued.)	62
8.4	Expressions and overlays. . . . .	63
8.5	Expressions and overlays (continued.) . . . . .	64
A.1	Components of the facial animation system. . . . .	68
A.2	Flow of the program. . . . .	71
A.3	Assembling the face data structure. . . . .	72
A.4	Example letter definition.	76
A.5	Example expression definition. . . . .	77

# List of Tables

4.1	Relationships between muscles (motions) and vertices. . . . .	25
4.2	Specifying vertices for tongue. . . . .	33
4.3	Assembling the tongue polygons. Each cell contains the vertices of a polygon. . . . .	34
6.1	Classification of vowels in Turkish. . . . .	42
6.2	Classification of letters based on similar mouth postures. . . . .	43
7.1	Available expressions and overlays with the definitions of their parameters. . . . .	53
7.2	Initialization times for the facial animation system. Times are given in <i>seconds</i> . . . . .	57
7.3	Performance issues. Rendering times are measured to render a frame of an animation. Times are given in <i>seconds</i> . . . . .	57

# Chapter 1

## Introduction

Facial animation is a very active research area that has attracted many researchers in the last decade. Due to its complex structure, animating the face with proper mouth postures quickly and convincingly is one of the most challenging research areas. There is a need for models that are capable of performing realistic facial movements. This is not only necessary for computer graphics applications, but also necessary for plastic surgery and criminal investigations [20].

The face is made up of skin, muscles and bone which heavily affect the appearance of the face. The skin has layers and it can be thought as an elastic material. Muscles apply forces which deform the skin. The shape of the face is mainly determined by the facial bone. It is not very easy to develop a model which is capable of both simulating the complex nature of different parts of the face and visually realistic. However, such a model can be developed by approximating some of the features. Thus, a realistic facial animation system should have a model which is very similar to real human face and should generate realistic mouth and lip postures to support realistic speech animation.

To implement such a realistic model, we model the face composed of the parts mentioned above. However, as some other parts of the face, the bone is not modeled as a real bone. It exists as a logical structure. The movable

part of the facial bone, called the *jaw*, is not modeled as a real jaw bone, either. Instead, a logical jaw bone is implemented by giving some tags to the polygons that reside in the jaw region. The skin is thought as a mass-spring network whose nodes are thought as vertices and edges are thought as springs. This makes deformation calculations easier. The muscles are thought as forces that affect and deform the skin. Thus, deformation rules and other physical properties are easily applicable to face and produce very realistic results. The animation system produces animation by generating necessary keyframes and inbetweens of an animation. To achieve realistic behavior of skin, cosine interpolation scheme is used to calculate the timing of inbetweens.

The system uses two approaches to implement realistic speech animation. The first approach is *physically-based modeling*. The muscles in the system are defined as forces and all of their physical properties exist in the system. A muscle has the following parameters: *contraction value*, *influence zone*, *fall start* and *fall finish* radii (which define the mostly affected area of the skin.) These parameters are used to define a muscle. Each muscle in the system is defined as a linear vector. To generate more realistic results and to achieve realistic mouth postures, we used the *pseudomuscle* based approach [10] which further simplifies calculations. In this way, sphincter muscles, like *Orbicularis Oris*, are approximated using linear muscles to simulate their behaviors.

The other technique used is *parameterization*. Each meaningful part of the text dictates a set of parameters such as muscle contraction values, rotation angle of jaw, etc. Based on these parameters, the shape of the face is then updated.

Facial animation does not only involve animating the lips according to sounds, but also involves animating other facial features. During the speech, the shapes of other facial agents, such as eyebrows, nose, cheeks, etc. also change. The main factors that alter the facial posture are the feelings of the speaker. These changes are called as expressions or emotional overlays and they are incorporated into the face model by the system. Using appropriate tags, the face is set to the desired expression or emotional overlay. These tags also help synchronization of expressions and overlays with speech.



The facial animation system is made up of a database, an input text and a parser. Together with these parts, the facial animation display system generates a properly shaped and shaded face model. The system can also generate keyframes and inbetweens of an animation sequence using cosine interpolation method which fits best to facial animation.

There are several studies in facial animation for English language. Due to the differences between English and Turkish, this study becomes important as it handles linguistic issues differently. English is made up of some little phonetic structures, called *phonemes* and they are minimum pronounceable items of language. It is not very easy to deduct pronunciation information from pure text in English or Turkish since phonemes should be generated first. The approach differs since the written form almost always dictates the spoken form in Turkish. Thus, there is no need to develop an intermediate mechanism to convert letters or syllables into phonemes. Phonemes in Turkish are also important. For example, pronunciation of the two ‘e’s in word ‘erkek’ are different in sound. But these ‘e’s are not visually distinct. We did not consider such phonetic issues in our system, since the main purpose is to generate a speaking synthetic face which is visually realistic.

The rest of the thesis is organized as follows. Chapter ?? explains the previous research on facial animation and speech animation. Different approaches for facial animation like keyframing, parameterization, structure-based and physically based are explained in chronological order together with their capabilities and limitations. Speech animation techniques and approaches for generation of facial expressions while speaking are also explained.

In Chapter 3, the structure of the face is explained. The reader is informed about the facial anatomy and important details of the face, like skin, muscles, mouth, jaw, teeth and tongue.

Chapter 4 discusses the facial modeling used in the system. This chapter discusses the limitations of the earlier model developed by Waters [15] and presents the improvements on the model. This chapter also gives necessary formulation to implement realistic muscle and skin behavior and information about the modeling to achieve realistic results. Information about the modeling

of muscles, mouth, jaw, eyes, teeth and tongue is given in this chapter.

The main ideas behind the facial animation are presented in Chapter 5. This chapter gives the necessary information about facial animation and discusses the techniques of facial animation. It also gives information about the main agents that are effective in facial animation, such as expressions and expression overlays. By comparing different approaches, the most useful and appropriate technique is proposed in this chapter.

This study differs from other facial animation studies that are implemented for English. Linguistic issues are explained in Chapter 6.

Components of the facial animation system, functionality of each component, interpolation schemes that are available for keyframed animation are discussed in Chapter 7. Some performance metrics are also given in this chapter.

Chapter 8 presents sample keyframes from an animation sequence and Chapter 9 concludes the study.

Implementation details which can be helpful in the future developments of the system are explained in Appendix A. Major data structures and some algorithms are explained in this chapter. Information about the modes of operation is also provided. The files associated with the system and their structures are explained.

# Chapter 2

## Related Work

We can group previous studies on facial animation into two categories. First group includes the models and animation systems whereas the second group includes the previous work done on realistic speech animation.

### 2.1 Facial Animation

There are several studies on facial animation. All of these studies aim to manipulate faces over time so that, the faces will have the desired postures in each frame of an animation sequence. One should process the face model data to set vertices to desired positions. First studies in facial animation were done by Parke in 1972 [12]. In his earlier studies, he developed a human face model and a facial animation system. This study used keyframing technique to animate the face. In this approach, each frame is generated by a computer program and then these frames are put together to form a film. Although very convincing results are achieved, this technique is very expensive in terms of the time spend on drawing each frame. Keyframing technique is very suitable for two dimensional animation but it cannot be easily adapted to three dimensional facial animation since each keyframe of the animation must be completely specified which is a tedious process for the user.

A solution for this problem is again proposed by Parke in 1982 [14]. The

direct parameterized model uses a set of parameters to define facial configurations. These models use local region interpolations, geometric transformations and mapping techniques to manipulate the features of the face. A fully parameterized model allows creation of any facial image by specifying the appropriate set of parameter values. There are defined set of parameters for specific parts of the face, such as eyes, mouth, etc. These are mainly about facial expressions. Width of the mouth, rotation angle of the jaw are examples of these parameters. There are also conformation parameters required for each person individually and these parameters represent the aspects varying from one person to the other. These parameters can be length and width of nose, shape of the jaw, etc. Each parameter in a parametric system affects a set of vertices so that any facial state can be defined easily by altering the parameters which move the vertices to desired new positions.

Parametric approach may seem as an exact solution for the bottleneck in keyframing approach. However, it introduces another problem. Since each parameter in the system affects a disjoint set of vertices, it is impossible to blend facial expressions. There are some vertices which must be affected by more than one expressions during speech. Platt [19] proposed a solution for the problem in keyframing approach. He used structure based facial model. Such models are based on anatomic properties of the face.

All of the discussed solutions are fairly satisfactory but they are not aware of the fact that the face is a complex biomechanical system. They always represent the face as a geometric model. Terzopoulos and Waters [20] developed a face model which is very similar to the real human face in terms of physical properties. This solution is called physically-based modeling. The face is composed of three layers and these layers are simulated using a mass-spring network. Their study produced very realistic results presenting the behavior of the skin. As they simulate the skin in a layered fashion, skin deformations like wrinkles are generated automatically.

Their studies are based on physically-based muscle modeling which is first introduced by Waters [23]. The main idea behind his study is that muscles are thought as forces and the skin of the face is thought as a mass-spring network. The face is deformed by applying physical rules to mesh structure. Satisfactory

results have been achieved.

## 2.2 Speech Animation

Researchers working on facial animation were not merely focused on implementing or simulating the realistic behavior of the skin. They also studied on realistic speech animation.

The motivation behind speech animation studies was to generate a convincing speaking face model by modeling a variety of mouth and lip postures. These postures should be interpolated in a realistic way. Initial studies about speech animation were again started by Parke [13]. Pearce et al. [16] used a parametric approach to animate speech. They also consider facial expressions and developed a mechanism to synchronize expressions with speech. Besides these model-based approaches, there are also image-based approaches for speech animation. Watson et al. [26] developed a morphing algorithm to interpolate phoneme images to simulate speech. They specified tiepoints on a still image of a real human and then by capturing 56 phonemes, they tried to simulate realistic speech by applying their morphing algorithm.

Waters and Frisbie [24] developed a coordinated muscle model and they produced a natural-looking speech animation on a facial image. Their study is based on the fact that muscles around the mouth are interacting and they tried to find out these interactions for natural-looking speech on a facial image. Their study was based on finding which muscles are active for each phoneme. However, their muscle structure is two-dimensional.

Basu [1] developed a three-dimensional model of human lips and a framework to train it from real data. His work is mainly the reconstruction of lip shapes from real data, it can also be used for lip shape synthesis for speech animation.

Attempts in speech animation are not limited to generating speaking face models. Another important problem is to synchronize speech with facial animation and a natural-looking speech should include a variety of properties.

For example, expressions and emotional overlays should be included in the system to make animation believable. An animated face should be synchronized with a given audio together with the emotional change on the face. We should keep track of timing information of audio and keyframes to achieve realistic lip motions synchronized with speech.

Parke [13] used parametric approach to achieve this synchronization. Pearce et al. [16] developed a rule-based speech synthesizer to synchronize speech with a three dimensional face model. In this study, they used two channels. They recorded the generated speech to the audio channel and frames of an animation of speaking face model to the video channel. Then the sequence is played to animate the face. This approach is not flexible because it is necessary to repeat whole process if the audio is changed.

“DECFace” [25] overcomes this limitation by proposing a new approach to synchronize audio and video automatically. DECFace has the ability to generate speech and graphics at real-time rates, where audio and graphics are tightly coupled to generate expressive facial characters. To do this, a mouth shape is computed for each phoneme and mouth shapes are interpolated using the cosine interpolation scheme. Audio server is queried to find out which phoneme is to be pronounced so that the appropriate mouth shape is generated synchronously for each frame.

Expressions and emotional changes affect the appearance of the face. This very important point is considered by Kalra et al. [9]. They decompose the problem into five layers. The higher layers are more abstract and specify *what to do* and the lower layers describe *how to do it*. The highest layer allows abstract manipulation of the animated entities. During this process, speech is synchronized with the eye motion and emotions by using a general and extensible synchronization mechanism. The lowest level includes applying abstract muscle action procedures which use *pseudomuscle based* techniques to control basic facial muscle actions. In pseudo-muscle based animation, muscles are simulated using geometric deformation operators [10].

Although speech is mainly related to the face, the other parts of the body usually play as big role as the face in speech. For example, hand gestures, body

movements are also important during speech. These features are considered by Cassell et al. [3]. They developed a system which automatically produces an animated conversation between multiple human-like agents with appropriate speech, intonation, facial expressions and hand gestures that are synchronized. Gestures and expressions are derived from the spoken input automatically in their system.

# Chapter 3

## Structure of the Face

The human face model used is composed of six main agents that affect the appearance of the face. As a summary these parts are:

1. *skin*, which forms the highly visible and elastic part of the face,
2. *muscles*, which have the most important role in deforming the skin and creating facial expressions,
3. *mouth* and *jaw*, which are used when creating lip shapes for speech,
4. *teeth*, which complete the model by increasing the realism,
5. *tongue*, which is visible in some phonemes during speech, and
6. *eyes*, which complete the model for expressions to increase the realism.

### 3.1 Skin

The face is covered by several layers of soft tissue. The mechanical behavior of the face skin is one of the most important details that affect the appearance of the facial expressions. The following properties of the face skin are considered in our implementation [15]:



- *The Poisson effect*: This describes the tendency of a material to preserve its volume when the length is changed. The facial skin is a soft tissue so that it is nearly not compressible. Therefore, when we contract a muscle, the skin in the influence zone of that muscle will tend to preserve its volume. However, its length is changed. Thus, there will be wrinkles appearing in the soft tissue.
- *Elasticity*: When we pull or push a muscle, the nearby surface of the skin will be affected which implies the elasticity of the facial skin tissue. The amount of displacement of a point is determined by the distance of the point from the muscle head, the elasticity of the nearby tissue and the zone of influence of the muscle. The displacement direction and amount is determined by the properties of the muscle fibres. That is, if the muscle fibres are collected and form a linear vector, the displacement will be towards the head of the muscle. For instance, we use *Zygomatic Major* muscle while smiling and this muscle is a *linear* muscle. If it is contracted, the skin will be deformed as if it were pulled from a single static point. In contrast, we use *Occipito Frontalis* to close eyes. This muscle is a *sheet* muscle and skin will be deformed as if it were being pulled from multiple points.

The uppermost layer of the skin is *epidermis*. It is made up of dead cells and has a thickness of  $\frac{1}{10}$  of dermal layer which is protected by *epidermis*. This makes the skin non-homogeneous and non-isotropic. Under low stress, dermal tissue offers low resistance to stretch but for higher stress, fully uncoiled collagen fibres become resistant to stretch. This can be explained by a biphasic stress-strain curve as in Figure 3.1. This type of motion is generally known as *viscoelastic behavior* of the skin [15].

The viscoelastic behavior of the skin is not implemented in the system since the face model has only one thin layer which corresponds to the face skin. In a real face, the skin has different layers which cause the viscoelastic behavior. Realistic speech animation is mainly considered, thus, some skin effects such as wrinkles are beyond the scope of this study. The implementation of expressions and realistic mouth postures is the main idea of this study. This model is not implemented in our system since the system has only one layer. We heavily

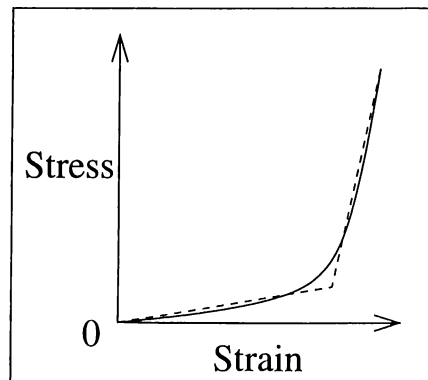


Figure 3.1: Visco elastic behavior of the skin.

worked on realistic speech animation, thus, some skin effects such as wrinkles are not very important for us. We focused on the implementation of expressions and mouth postures.

## 3.2 Muscles

The muscles of the face are generally known as muscles of facial expressions. However, some of the muscles have other important functions such as moving the cheeks and lips during mastication and speech, or constriction (closing) and dilation (opening) of the eyelids.

Muscles are bundles of fibres working together. The length of a fibre alter the power and range of the muscle. Shorter fibres are more powerful but have smaller movement ranges.

There are three main types of muscles in the face:

1. *Linear*: A linear muscle is a bundle of fibres that have a common originating point in the bone. For example, *Zygomatic Major* is a linear muscle used used for smiling which pulls the corner of the mouth.
2. *Sheet*: A sheet muscle has a broad and flat sheet of muscle fibres without a specific emergence point. *Occipito Frontalis* is a sheet muscle used for closing eyes.

3. *Sphincter*: A sphincter muscle consists of muscle fibres that loop around a virtual center. *Orbicularis Oris* is a sphincter muscle used for puckering lips and circling the mouth.

### 3.2.1 Muscles of the Face

Facial muscles generally attach to a layer of skin at their insertion. Some of the muscles attach to skin at both the origin and the insertion, such as *Orbicularis Oris*.

Main Muscles in the face, their places and their functions are given below [15].

---

<i>Orbicularis Oculi</i>	
Place	around the eye
Type	sphincter
Functions	protection of eye, control the eyelids

---

<i>Corrugator Supercilii</i>	
Place	medial end of each eyebrow
Type	linear
Functions	draws eyebrows, produces wrinkles with <i>Orbicularis Oculi</i>

---

<i>Levator Labii Superioris Alaeque Nasi</i>	
Place	from the top side of the nose to lips
Type	linear
Functions	raises and inverts upper lip, produces wrinkles on and around the nose, dilates the nostrils

---

<i>Orbicularis Oris</i>	
Place	around the mouth
Type	sphincter
Functions	puckering lips, circling mouth

---

*Buccinator*

Place from the corner of the mouth to cheeks

Functions : compress cheeks to prevent accumulation of food in the cheek

---

*Levator Labii Superioris*

Place attached to the bone of zygomatic, orbit and maxilla embedded at the other end into the upper lip between *Levator Labii Superioris Alaeque Nasi*

Type linear

Function : raises the upper lip

---

*Zygomatic Major*

Place from the malar surface of the zygomatic bone to the corner of the mouth

Type linear

Function : pulls the corner of the mouth

---

*Zygomatic Minor*

Place inserts in the skin of the upper lip

Type linear

Function : elevates the upper lip

---

*Depressor Anguli Oris and Depressor Labii Inferioris*

Place both arise from the mandible and converges to the corners of the mouth

Type linear

Functions : Depress the corner of the mouth downward and laterally

---

*Risorius*

Place located at the corner of the mouth

Type linear

Function : smiling muscle

---

*Mentalis*

Place originates from the mental tuberosity inserts into the skin

Type linear

Function : elevate the skin of the chin, protrusion/eversion to help drinking

---

*Levator Anguli Oris*

Place	:	mixes with the muscles at the corner of the mouth
Type		linear
Functions		the only deep muscle that open lips raises the modiolus, displays teeth

---

*Depressor Anguli Oris*

Place		arises from the near of platysma and inserts into the angle of mouth
Type		linear
Functions		depresses the modiolus and buccal angle laterally to open mouth and in the expression sadness

---

*Depressor Labii Inferioris*

Place		Originates near the origin of triangular muscle
Type		linear
Function	:	pulls the lower lip down and laterally during mastication

---

### 3.3 Mouth and Jaw

Mouth is the most flexible facial agent. Mouth and the surrounding agents, i.e., the lips, have the most important task in speech animation since the words are recognized according to the shapes of lips. Formation of the mouth shapes, such as wide, narrow and puckered lips gives a clue about the letter (or phoneme) that is pronounced. Furthermore, mouth postures help determining the emotion of a person to some extent. An angry person will speak with his/her lips and teeth tightened. Lips become thin.

Lips are not the only agents that affect the mouth posture. Mouth is meaningful as long as it can be opened. We open our mouth by moving the lower jaw. Therefore, the jaw becomes another important agent of the face that plays great role in facial animation and that makes the mouth posture be more realistic. The jaw is the only facial bone which is movable. The motion of the jaw is essentially a rotation around an axis connecting the two ends of the jaw bones [7].

Mouth and jaw are meaningful when they are cooperated. In other words, we use jaw rotation to open the mouth, together with the muscles around the lips. Jaw rotation is necessary for the mouth to achieve desired speech and expression postures. Rotation of the jaw affects lips. However, the amount of rotation is not fully applied to upper lip and lower lip. The corner of the lips are affected by approximately one third of jaw rotation angle whereas the middle section of the lower lip is fully affected by the rotation of jaw [15]. Another important characteristic of the upper lip is that it can be raised and lowered. This effect can be achieved by using muscles around the lips. Lower lip shape is important for pronunciation of some letters such as ‘f’ and ‘v’. To get realistic mouth shapes for these letters, another muscle is added which is used to pull or push the lower lip to back and front.

### 3.4 Teeth

Teeth are helpful during the generation of some sounds and they define the structure of the mouth as the other bones. However, they differ from other bones in that they are visible.

### 3.5 Tongue

The tongue is a very powerful muscular organ with a really interesting ability to change its shape, orientation and position. It is used and visible for some letters, such as ‘d,’ ‘l,’ ‘n’ and ‘t.’ The tongue is covered by a mucous membrane. The tongue is composed of muscles, nerves and blood vessels only, except its cover. The muscles of the tongue are divided into two groups: *intrinsic muscles* which are placed inside the tongue body, and *extrinsic muscles* which are attached to tongue and originating outside the tongue [15].

## 3.6 Eyes

The eyes are important for vision. They are the end organs of the sense of vision [15]. The eyes are controlled by muscles and these muscles provide the accurate positioning of eyes. An eyeball is composed of four main parts, namely sclera, cornea, iris and retina.

## 3.7 Other Important Details of the Face

The details explained above are the most important ones which affect the realism in speech animation. We could say the following features of the face are also important because of their role in facial appearance.

- *Nose*: Nose is very important especially for the expression *disgust*. The nose also moves during deep respiration and inspiration. It also has an identification feature since the size and shape vary among people.
- *Ears*: Ears increase the realism of the face. They complete the face model.
- *Cheeks*: Especially in emotional states, cheek movements are visible and these movements include the movements of lower teeth and jaw. Actions like puffing, sucking will alter the shape of the cheeks.
- *Neck*: Neck is important for the movements of entire head such as nodding, turning and rolling.
- *Hair*: Hair is necessary to complete a realistic model. Hair style is an indicator of gender, race and individuality. However, modeling and animation of hair is a very active research area [17].
- *Accessories*: The accessories worn on the face and head are related to individuals and they may serve as identification marks used by some people. In this context, glasses, hats, makeup and jewelry become important. However, they have no importance in speech animation.

Although the details mentioned above are very important, all of them are not implemented in our system. Since the system is mainly designed for realistic speech animation, ears, cheeks, hair, neck or accessories are not in the scope of this research.



# Chapter 4

## Modeling in the System

The face is modeled as a mesh of triangles in the system. There are 1700 polygons in the model. The face model is divided into three regions, which is given in Figure 4.1. Upper region contains 610 polygons, Lower region contains 240 polygons. There are 38 polygons in the intermediate region which is named as BOTH. Vertices of the polygons in this region provide the continuity of the face. Teeth have approximately 800 polygons and tongue has about 80 polygons. Eyes are not modeled as polygons because they will not be deformed. We did not model eyes as meshes of polygons since they never deform. Instead, they are drawn using sphere drawing procedures of OpenGL<sup>1</sup> [11].

The face model has five main parts:

1. mouth (lips) and jaw,
2. muscles,
3. eyes and eyebrows,
4. teeth, and
5. tongue.

The face is a composite structure and its components have very complex

---

<sup>1</sup>OpenGL is a registered trademark of Silicon Graphics, Inc.

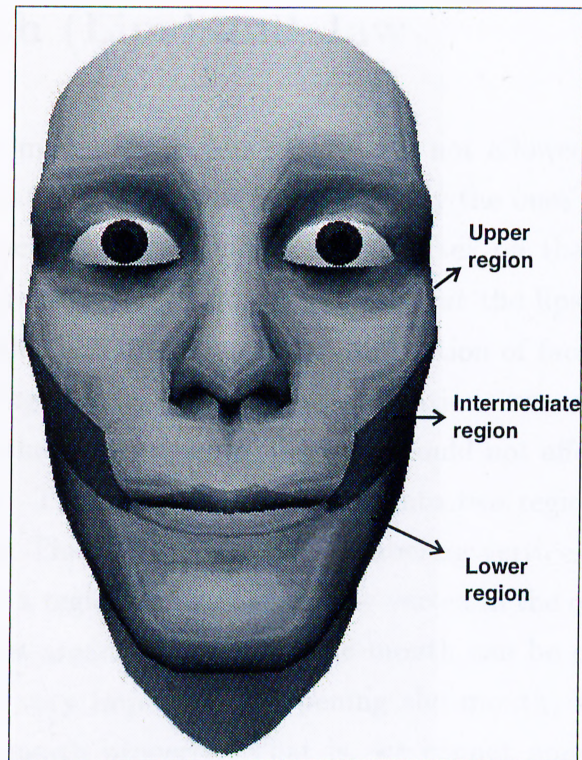


Figure 4.1: Regions of the face.

substructures. In our implementation, however, we reduced the complexity of these components for simplifying the implementation.

## 4.1 Overview of the Original Face Model

We used a face model developed by Waters [15]. The model is composed of polygons and some muscle vectors defining muscles. However, these are not enough for a realistic speech animation since main muscles that control the shape of the mouth do not exist in this model. It is not very suitable for speech animation since the mouth is not designed to be opened. It has also no eyes, no teeth and no tongue. The regions of the face (see the following chapters for details) are not defined and thus it does not allow speech animation to be performed. Therefore, this base model is modified for the needs of a speech animation system. These modifications are explained in the sequel.

## 4.2 Mouth (Lips) and Jaw

In the mentioned model above, the mouth was not allowed to open. This is due to the fact that the vertices of lips, especially the ones between the lower and upper lips, were shared. Thus, when a vertex on that line is changed, both of the lips are affected. It is necessary to *cut* the lips into two parts to get an open mouth. The first step is the duplication of face vertices to make a separation among them. After such a separation, two regions are needed since a muscle in the lower region of the face should not affect a vertex in the upper face region. Thus, the face is divided into two regions, namely upper and lower regions. This is implemented by labeling vertices with appropriate tags. An action in a region will not affect any vertex in the other region. With the help of muscles around the mouth, the mouth can be opened. Although these muscles are very important in opening the mouth, we need to rotate the jaw to open mouth properly. That is, we cannot open the mouth in a realistic manner without jaw rotation. As explained in Section 3.3, the jaw is assumed to be rotating about an artificial axis passing through the back of the face. However, this yields another problem which is the discontinuity in the face and unexpected behavior of the skin: when we rotate the jaw, which polygons should be rotated? If we tag polygons rather than vertices, this is impossible since the polygons near the mouth and in cheeks are shared and provide the continuity of the face. That is, if we give **UPPER** or **LOWER** tags to these polygons, jaw rotation will affect them partially. It is feasible to give tags to vertices rather than giving tags to whole polygons. Thus, vertices of polygons in cheeks and near the mouth are labeled with **UPPER** or **LOWER** tag and some of them are labeled with **BOTH** tag. When the jaw is rotated, polygons with tags **LOWER** and **BOTH** should be affected. So far, three tags seem enough for managing all of the actions. However, when the jaw is rotated, lower teeth should be rotated as well. If we give **LOWER** tag to lower teeth, they will be affected by the actions of the lower face muscles. Thus, we need a new type of action, which purely rotates the lower teeth as a rigid body, namely jaw rotation. So we give **JAWROT** tag to lower teeth to rotate or move them with the jaw bone.

As a summary, the face is divided into three regions to handle specific

muscle actions and jaw rotations correctly. Upper and lower face regions are reserved for vertices which are affected by only upper and lower face muscles, respectively. The tag **BOTH** is used for the vertices which are to be affected by both types of muscle actions. **JAWROT** tag is reserved for lower teeth which is only affected by jaw rotation and nothing else.

### 4.3 Facial Muscles

Earlier model had about 18 muscles to deform the skin of the face and major muscles around the mouth were absent. The first step is to add new major muscles that have great role in changing the shape of the mouth. In addition, the muscles were defined as vectors. Also, sheet or sphincter muscles need extra data structures. In our model, we used muscle vectors to approximate sheet or sphincter muscles. For example, *Orbicularis Oris* is defined as five muscle vectors in five different directions. Four of these muscle vectors are used to deform the mouth in two dimensions, namely  $x$  and  $y$ . The fifth muscle is used to achieve protrusion effects.

Muscles in the model are shown in Figure 4.2. Location of these muscles on the face are given in Figure 4.3.

#### 4.3.1 Modeling of Facial Muscles

Muscles are defined as vectors and they are very important to generate desired realistic facial postures. Structure of the muscles and details of implementation are explained in the sequel.

#### Muscle Parameters in the Model

As a summary, some extra muscles are added to model and *Orbicularis Oris*, is defined using the existing linear muscle structure. Thus, we have 35 muscles and five of them emulate the *Orbicularis Oris*. Each muscle has two symmetric

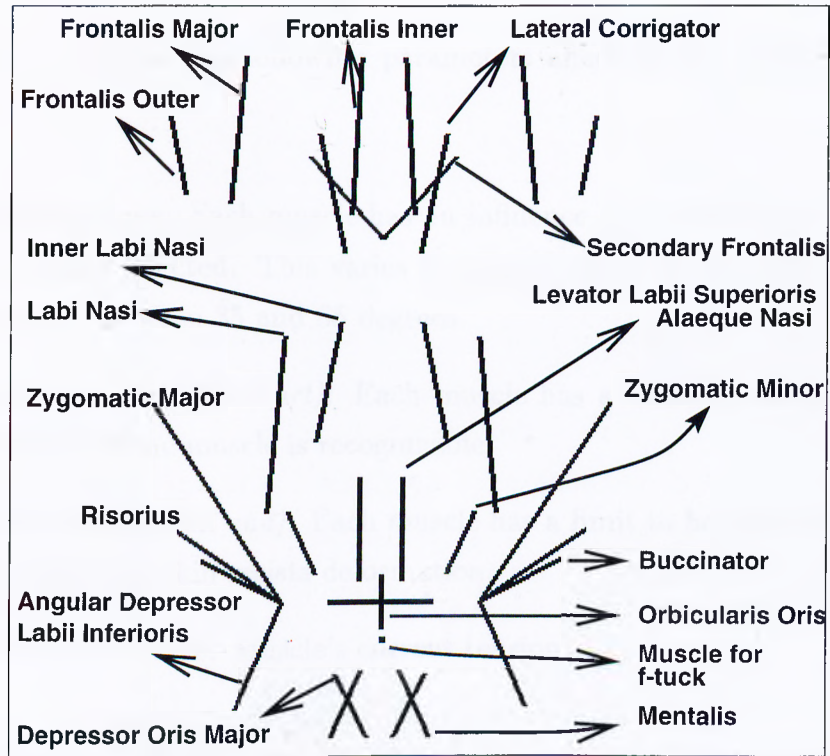


Figure 4.2: Facial muscles in the model.

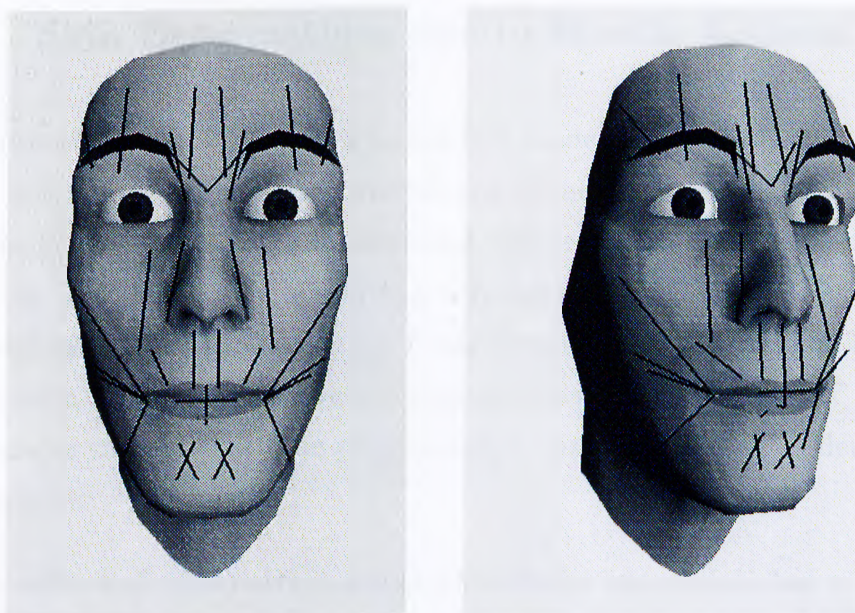


Figure 4.3: Location of facial muscles on the face model.

parts; one on the left and the other on the right of the sagittal (median) plane.

Each muscle has the following parameters affecting the behavior of the muscle:

- *Influence zone*: Each muscle has an influence zone in which the vertices are mostly affected. This varies from one muscle to the other and it is typically between 35 and 65 degrees.
- *Influence start (fall start)*: Each muscle has a tension after which the influence of the muscle is recognizable.
- *Influence end (fall end)*: Each muscle has a limit to be tensioned. After this limit, the skin resists deformation.
- *Contraction value*: Muscle's current tension.

The shape of the face can be altered by updating these parameters for muscles. For example, if we want to set smiling expression on the face, we should increase the *contraction value* of *Zygomatic Major* muscle.

### 4.3.2 Skin Deformations due to Muscle Actions

As mentioned before, each vertex in the face model has a unique tag which is used in deformation. The vertex will be repositioned if the action of the muscle affects it. If a vertex is to be repositioned, this is determined by checking the vertex tag and the muscle action tag and the new position of the vertex is calculated using the formulation for linear muscles given in [23]. The relationship between action tags and vertex tags is shown in Table 4.1. “+” denotes the action or muscle affect the corresponding vertex, whereas “-” denotes no changes occur.

The vertices of each polygon are given with the appropriate tag so that an action of a muscle, jaw or an eyeblink will not affect an irrelevant vertex and so a polygon.

Motion or Muscle Tag	Vertex Tag					
	UPPER	LOWER	BOTH	NONE	JAW_ROT	EYEBLINK
UPPER	+	-	+	-	-	-
LOWER	-	+	+	-	-	-
BOTH	+	+	+	-	-	-
JAW_ROT	-	+	+	-	+	-
EYEBLINK	-	-	-	-	-	+

Table 4.1: Relationships between muscles (motions) and vertices.

The influence zone of the muscle can be viewed as a circular shape and the fall-off is along the radius of this circle. The direction is towards the point of attachment to the bone. At the point of attachment to the bone, we can assume zero displacement whereas the maximum displacement is at the point of attachment to the skin.

All facial muscles in the model are thought as linear muscles. A linear muscle is designed as a force, so that its direction is also important and defined by the direction of the muscle vector. The starting point of that vector is never repositioned and it is the originating point of the muscle. This is similar to real muscle structure since some of the muscles have emergence point at the bone and insertion point into the skin. A muscle pulls or pushes vertices along this vector. A figure representing the parameters of a muscle is given in Figure 4.4.

In Figure 4.4,

- $P$  is a point in the mesh,
- $P'$  is its new position after the muscle is pulled along the  $V_1 V_2$ ,
- $R_s$  and  $R_f$  represents muscle fall start and fall finish radii, respectively,
- $\theta$  represents the maximum zone of influence, typically between 35 and 65 degrees,
- $D$  is the distance of  $P$  from muscle head and
- $\alpha$  is the angular displacement.

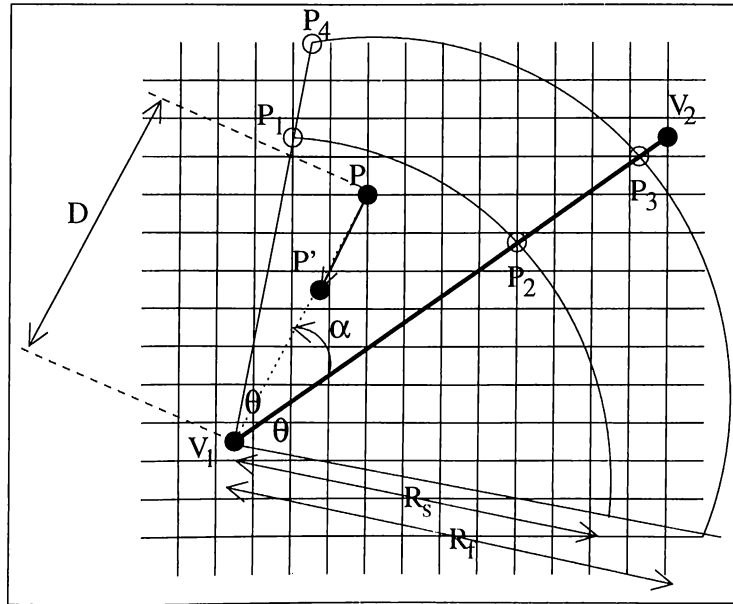


Figure 4.4: Parameters of a muscle.

Please note that Figure 4.4 gives the muscle behavior in two-dimension but the formulation can be extended to the third dimension by applying the same rules to the third dimension.

If  $P(x, y)$  is a mesh node and  $P'(x', y')$  is its new position, and  $P(x, y)$  is in the region of  $V_1P_3P_4$  and moved along  $PV_1$ , the calculation of  $P'$  is as follows:

$$P' = P + k \cdot a \cdot r \frac{PV_1}{\|PV_1\|}$$

where

$k$  is the muscle spring constant,

$a = \cos(\alpha)$  and

$$r = \begin{cases} \cos\left(\frac{1-D}{R_s} \frac{\pi}{2}\right) & \text{if } P \text{ in } (V_1P_1P_2) \\ \cos\left(\frac{D-R_s}{R_f R_s} \frac{\pi}{2}\right) & \text{if } P \text{ in } (P_1P_2P_3P_4) \end{cases}$$

The formulation given above is enough for calculating the new position of a node which is affected by a linear muscle. However, the formulation differs for other types of muscles which have elliptical influence zone. Such muscles have no angular displacement factor, since nodes around a center are squeezed as if



they were drawn together like a spring bag. Thus, the function for repositioning the node becomes:

$$x' \propto f(k, r, x)$$

$$y' \propto f(k, r, y)$$

Although this formulation is very suitable for muscles that have elliptical effect, this scheme is not used in the system. We used an approximation for such sphincter muscles which includes using four linear muscles for elliptical affect. This yields a very good approximation in two dimensions. Since our system has a third dimension, another linear muscle is used to represent muscle behavior along  $z$ -axis to achieve protrusion effect. This approximation is used for *Orbicularis Oris*. We can change the shape of the mouth by updating the muscle parameters for each linear muscle used for defining *Orbicularis Oris*.

We have five linear muscles, including two in horizontal and vertical directions and one for the third dimension. Vertical muscles have an influence zone of 140 degrees and horizontal ones have an influence zone of 40 degrees. The last muscle, which is along the  $z$ -axis has an influence zone of 140 degrees as the vertical ones.

Figure 4.5 represents the abstraction used for *Orbicularis Oris*.

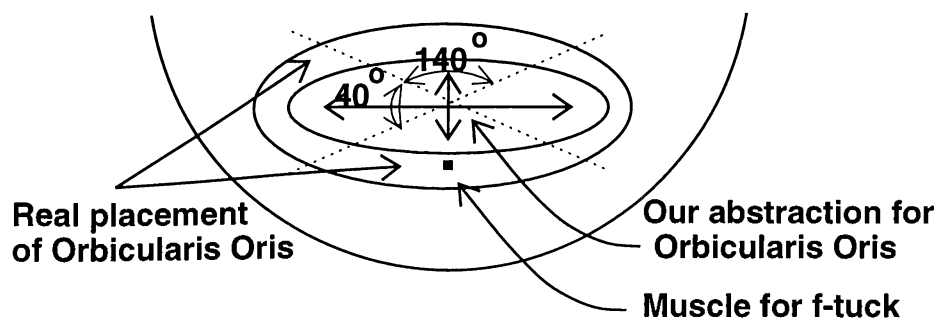


Figure 4.5: Abstraction of *Orbicularis Oris*.

## 4.4 Eyes and Eyebrows

During speech, we do not only change the shape of our mouth. The size of the eye, direction of eyegaze, shape of eyebrows are also changed. So, the facial animation system is not only altering the shape of the mouth according to pronounced letter, but also changing the face shape in other parts. That is, when we animate the speech, we should include important characteristics which may completely change the meaning of the word. For instance, emotional changes in the face posture may result in a completely different understanding of the word being said. Emotions cause changes in eye and eyebrow shapes. Therefore, it is necessary to include eyes and eyebrows in the facial model and simulate their behavior to increase the realism of the animation. For example, a raised eyebrow implies that the speaker is *surprised* whereas frowning means *anger*.

In the face model, the eyes are defined as three concentric spheres which yields a good approximation of the real eye. This approximation is a result of eyeball lenses. Eye spheres are flexible in terms of their radii and center coordinates. The structure of the eye model allows changing parameters for each eye. Eyes have the following features that should be considered in realistic speech animation.

1. *Eye Gaze*: when we look at a target (an object), our left and right eyes move to look at that object. Eyes can rotate about  $x$  - axis,  $y$  - axis or *both* - axes. But it is impossible for an eye to rotate about  $z$  - axis. When the eye is looking at an object, centers of the concentric spheres are rotated according to the position of target. The eye model is given in Figure 4.6.
2. *Eye Blinks*: for eye blinks, we need to close the eyelids. This yields a new region in the face, tagged as EYEBLINK. To close eyes, vertices along the upper part of the eyelid edge will be replaced and their coordinates will be set to lower part neighbors.
3. *Eyebrows*: eyebrows are especially important in implementing expressions. Expressions need different shapes of eyebrows. For example, *anger*

and *happiness* expressions have completely different eyebrow shapes. Eyebrows do not introduce a new region since they can be directly controlled by existing muscles in the model. Eyebrows are generated by painting specific polygons to black.

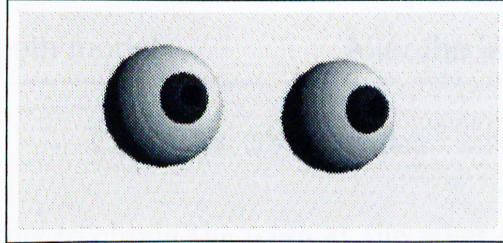


Figure 4.6: The eye model.

## 4.5 Teeth

Another important agent in the face for the facial animation are teeth which makes the animation convincing. Some parts of the teeth are visible during speech. Since the aim is to generate naturally-looking speech animation, we should include all visible agents of the face in the model.

Teeth are composed of two parts, upper and lower teeth. Upper teeth never move, but the lower teeth move with the lower jaw. As the lower jaw rotates, lower teeth move along with the jaw. Teeth are modeled as 32 five-sided polygons. A tooth is shown in Figure 4.7. The complete teeth model is shown in Figure 4.8.

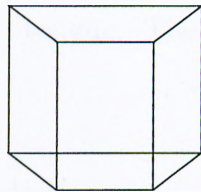


Figure 4.7: A tooth model.

First, half of the upper teeth are defined. They are reflected about  $x$ -axis to get the half of the lower teeth and then these two halves are reflected about

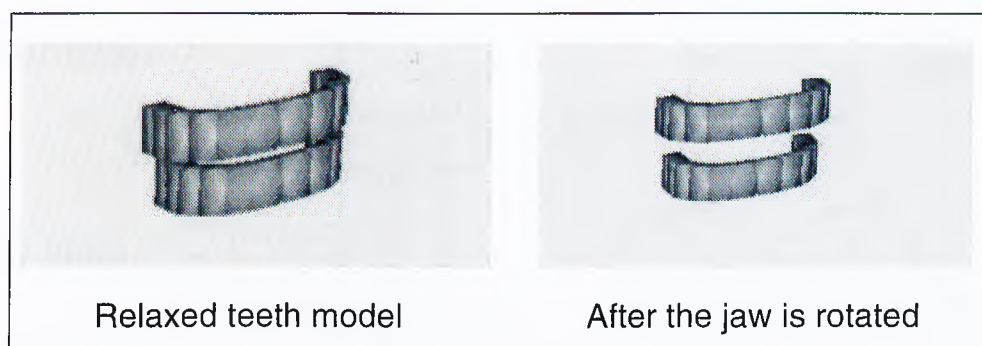


Figure 4.8: Teeth model.

median (sagittal) plane to obtain the teeth at the other side. Lower teeth are scaled down and translated backwards a bit to make them realistic.

## 4.6 Tongue

The tongue is another necessary agent which should be included in a realistic speech animation, since it is visible in some letters, such as ‘*d*’, ‘*l*’, ‘*n*’ and ‘*t*’. The structure of the human tongue is very interesting. Its very complicated, soft and flexible structure is controlled by a variety of muscles which determine the shape of the tongue. Tongue may also change its length and orientation; e.g., the tip of the tongue may be towards upper teeth or to the left of the mouth. The tongue is defined as four sections and each section is controlled by five parameters. The tongue in our face model from different point of views are shown in Figure 4.9.

Each section of tongue model is connected to previous (rearer) section. This provides the connectivity of the tongue. The parameters used to control the tongue are as follows:

1. *width* is the width of a section,
2. *thickness* is the thickness of a section,
3. *height* is the height of the front edge of that section from the lower palate,

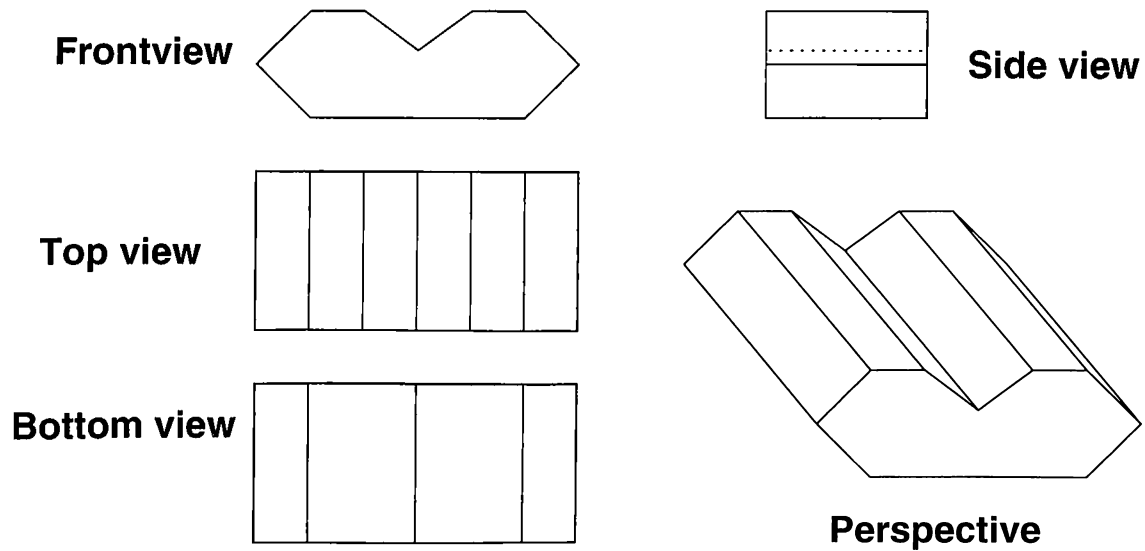


Figure 4.9: Tongue model from different views.

4. *midline* is the height of the middle of the tongue from the tongue base and
5. *length* is the length of the section.

These parameters and their denotations are shown in Figure 4.10.

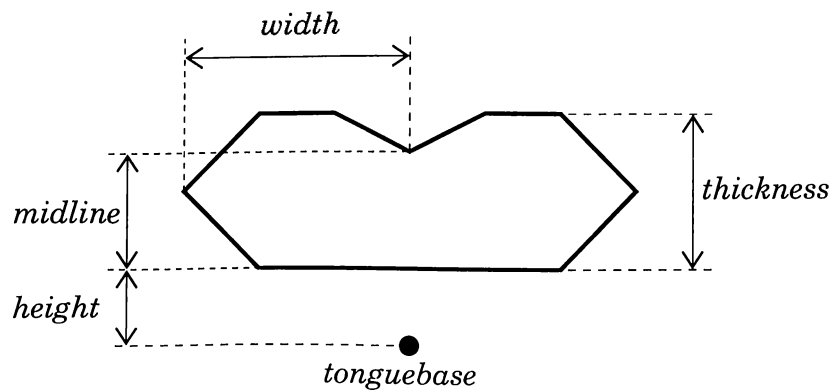


Figure 4.10: The parameters of tongue.

The tongue has four sections of 20 polygons and a section of 12 polygons to close the tip. The implemented tongue model gives a realistic approximation of the human tongue for speech animation. The system allows changing parameters of each section gradually. Tongue has a base which is placed at the back of the tongue.

### 4.6.1 Assembling the Tongue

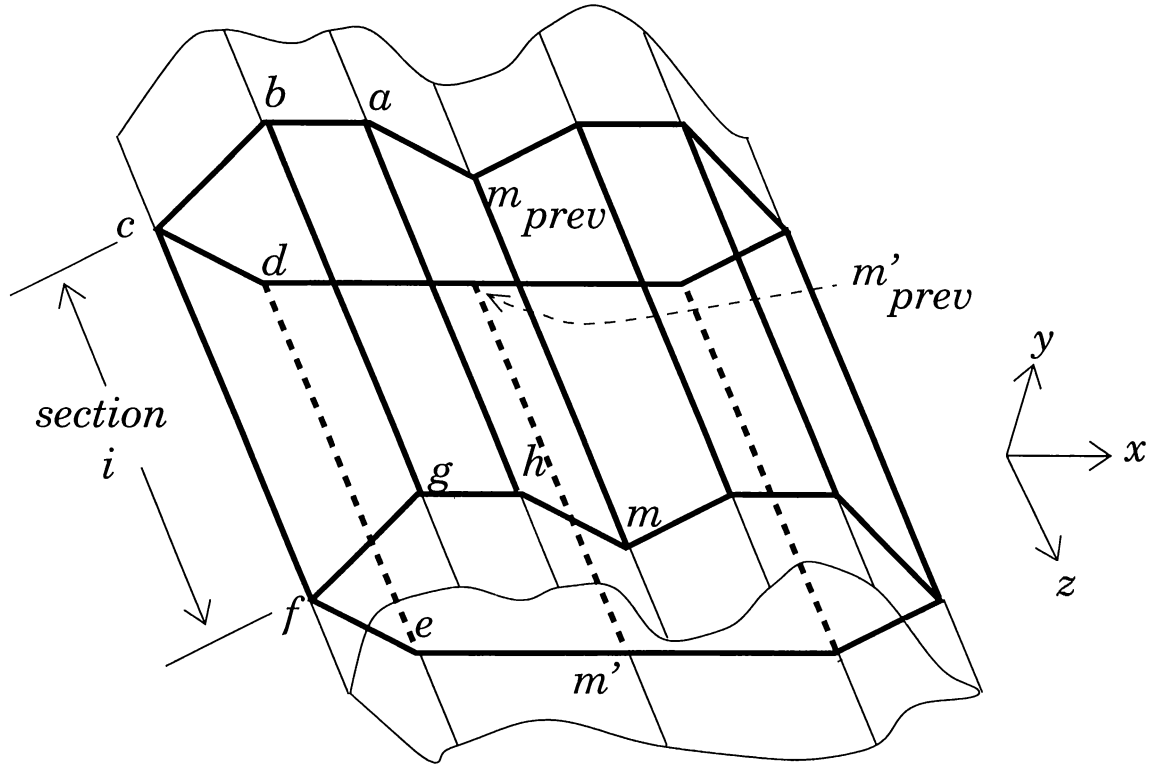


Figure 4.11: A section of tongue model.

To create a section of tongue, we should specify the coordinates of vertices. A section of the tongue is shown in Figure 4.11. The tongue has four such sections and a closed tip. Vertex coordinates are calculated using the parameters of the tongue.

Vertices placed at the back of the tongue section are created using the parameters of the previous section. Sections are indexed starting from the farthest section (the section that has the lowest  $z$  coordinates.)

As shown in Figure 4.11, we need 12 vertices for each section which creates 20 polygons. These vertices are named as  $a$ ,  $b$ ,  $c$ ,  $d$ ,  $e$ ,  $f$ ,  $g$ ,  $h$ ,  $m$ ,  $m_{prev}$ ,  $m'$  and  $m'_{prev}$ . We have a *tonguebase* which is placed in the middle of the first section (the farthest section.) Vertices are created using the scheme in Tables 4.2 and 4.3.

In Table 4.2,

$tb = \text{tonguebase}$

Vertex	Coordinates of the Vertex			Condition
	$x$	$y$	$z$	
$m$	$tb$	$tb + height_i$ $midline_i$	$tb + \sum_{j=0}^i length_j$	
$m'$	$m$	$m - midline_i$	$m$	
$m_{prev}$	$m$	$tb + height_i +$ $midline_i$ $tb + height_{i-1} +$ $midline_{i-1}$	$m - length_i$ $m - length_i$	if $i = 0$ if $i > 0$
$m'_{prev}$	$m_{prev}$	$tb + height_i$ $tb + height_{i-1}$	$m_{prev}$	if $i = 0$ if $i > 0$
$a$	$m_{prev} - width_i/3$ $m_{prev} - width_{i-1}/3$	$tb + height_i +$ $thickness_i$ $tb + height_{i-1} +$ $thickness_{i-1}$	$m_{prev}$	if $i = 0$ if $i > 0$
$b$	$a - width_i/3$ $a - width_{i-1}/3$	$a$	$m_{prev}$	if $i = 0$ if $i > 0$
$c$	$b - width_i/3$ $b - width_{i-1}/3$	$tb + height_i +$ $thickness_i/2$ $tb + height_{i-1} +$ $thickness_{i-1}/2$	$b$	if $i = 0$ if $i > 0$
$d$	$b$	$tb + height_i$ $tb + height_{i-1}$	$b$	if $i = 0$ if $i > 0$
$e$	$m - 2width_i/3$	$tb + height_i$	$m$	
$f$	$e - width_i/3$	$tb + height_i +$ $thickness_i/2$	$m$	
$g$	$e$	$tb + height_i +$ $thickness_i$	$m$	
$h$	$m - width_i/3$	$tb + height_i +$ $thickness_i$	$m$	

Table 4.2: Specifying vertices for tongue.

- $i$  represents the section number,
- $tb$  represents the corresponding coordinate ( $x, y$  or  $z$ ) of *tonguebase*,
- $height_i, midline_i, thickness_i, width_i$  and  $length_i$  are the parameters of  $i^{th}$  section and
- $a, b, c, \dots, m, m_{prev}, m', m'_{prev}$  are the vertices as explained above.

Let us give an example to explain how it works. If we consider the coordinates of vertex  $m'$ ;

$$m'_x = m_x = tb_x$$

$$m'_y = tb_y + height \text{ parameter of section } i$$

$$m'_z = m_z$$

After specifying the vertices, the following polygons will be generated. These vertices are generated for each change in the parameters of tongue and tongue polygons are updated accordingly. Polygons in the tongue model are generated using the rule given in Table 4.3.

$m, m_{prev}, a$	$a, h, m$	$a, b, h$	$b, g, h$	$b, c, g$
$c, f, g$	$f, d, c$	$f, e, d$	$m', d, m'_{prev}$	$m', e, d$

Table 4.3: Assembling the tongue polygons. Each cell contains the vertices of a polygon.

To close the tip of the tongue, 6 new polygons are generated. These polygons form a hexagon to close the tip of the tongue. The hexagon in one side, then, will be reflected to the other side thus, the tip of the tongue is closed after rendering.



## 4.7 Summary

As a summary, we have three main regions on the face, namely **UPPER**, **LOWER** and **BOTH** (intermediate) regions. Rotation of the jaw introduces a new region, called **JAWROT**. Eye blinks need another region, which is **EYEBLINK** region. This classification is also used in classification of actions (muscle actions or motions like jaw rotation) which affect the face. Actions are classified so that they affect only specific regions of the face.

As mentioned before, the face is deformed using some actions and repositioning the corresponding vertices. The question, now, is “how can we choose these vertices?” Actually, the answer is straightforward. Vertices are labeled with proper tags implying the type of muscle action or motion that affects the vertex. For instance, if a vertex has the tag **UPPER**, then it can be affected only by muscle actions or motions which have **UPPER** tag.

# Chapter 5

## Facial Animation

### 5.1 Overview

There are several ways for animating a face model. The most important techniques are known as interpolation, performance-driven, direct parameterization, pseudo-muscle based, muscle based and discussed in Chapter ??.

We used a hybrid scheme which combines interpolation, pseudo-muscle based and parameterization. We defined muscles as linear vectors and we need parameters to control facial expressions and mouth postures for letters. Animation frames are generated using interpolation mechanism and each keyframe is written to the disk by the system.

As discussed in Chapter 4, muscles are modeled as linear vectors. The starting point of a muscle is never repositioned.

Although muscles are defined as vectors and the structure of the model is similar to mass and spring systems, our system needs parameterization to some extent. Emotional changes are driven by parameterization. For example, each expression has an intensity level and according to that “level” parameter, the face model is updated and desired expression is set on the face.

## 5.2 Expressions and Overlays

In facial animation, letters (or phonemes) are not satisfactory unless supported by some additional agents which increase the realism of the animation.

If we want to make a realistic speech animation, we should consider lots of features that affect the realism of the model being animated. Animating the lips only may not give the desired result. Emotional changes on the face would be more explanatory. Sometimes people prefer changing their facial postures than speaking. In addition, people make specific postures during the speech. These postures are generated by the speaker to make sure that the listener follows him/her. Furthermore, some physical needs of the face should be taken into account so that the synthetic speaker would be interpreted as if he were really speaking. For instance, eye blinking is an important emotional overlay. Such little but very important details make animation believable. Thus, we should consider such emotional changes during the implementation of speech animation.

A facial expression is a representation of an emotional state. During the talk, people generally do not stand still. According to the content of the talk and feelings of speaker, s/he may make additional movements. These movements are mainly based on facial emotions, and generated facial postures are generally called as facial expressions. For instance, when a person is angry, his/her talk will usually be fast. Muscles around the eyebrows are contracted and teeth are tightened. This is an example of a person who has “anger” as an expression. The feeling that this person has is the “emotion.” There are six universal facial expressions: *anger*, *disgust*, *fear*, *happiness*, *sadness* and *surprise*. These expressions are linked to the emotions and attitudes. Each emotion is characterized by specific facial changes. For instance, “*fear*” is associated with tense stretched lips with raised eyebrows [15].

### 5.2.1 Expression Overlays

Since emotional changes heavily affect the realism of facial animation, it is necessary to add expressions to animation. Although implementation of facial expressions is very important and is a must for an animation, it is not enough for a realistic speech animation.

During a conversation, speaker generally makes specific postures and is rarely still. According to the content of the text, speaker's eyes may move, eyebrows may raise, eyelids may blink etc. Thus, a realistic animation of speech should include such characteristic facial movements such as expressions, eye movements that occur during speech. Necessary expressions can be emotionally based or non-emotional.

- *Emotional overlays*: Emotional overlays are generally communicated via audible words. Emotional qualities of the voice and visible facial expressions are the two mostly used communication channels to convey emotional information.
- *Emotions*: Emotion combines visceral and muscular physiological responses, autonomic nervous system and brain responses, verbal responses and facial expressions [6].
- *Non-emotional overlays*: During the speech the expression of the face usually depends on the emotion. There are, however, cases in which the resulting expression is due to other causes. Some facial movements are used as communication punctuation marks, similar to those in written text. Non-emotional overlays can be classified into the following groups as stated in [15]:
  - *Emblems* depend on the culture. These are movements and their meanings are well known. Used instead of verbal communication; for example, raising eyebrows means “No” among Turks.
  - *Manipulators* result from the physical needs of face. For instance, blinking the eyes makes the eye wet.

- *Conversational signals* or *illustrators* are generally accompanied with accented words. *Illustrators* are affected by emotion. For example; an angry person has more and quick motions than a sad person. Also the intensity level of emotion affects the movements and frequency of movements.
- *Punctuators* occur at pauses. These are similar to the ones that we use in the written text. Obviously, the rate of occurrence and the type of punctuator depend on the emotion. Generally, eyeblinks and head movements occur at pauses.
- *Regulators* are movements that control the flow of the conversation. That is, *regulators* are used to determine who is to speak. As we can deduct from a conversation, the following are the *regulators* [5]:
  1. The speaker wants to give up his/her turn. S/He turns his/her head to the listener and takes a more relaxed position.
  2. The speaker starts speaking. S/He turns his/her head away from listener and begins some gestures with hands and arms.
  3. The speaker wants to continue speaking. S/He wants to make sure that the listener is following. S/He turns his/her head to the listener.
  4. The speaker continues speaking. S/He turns his/her head and eyes away from the listener. Generally follows the previous type of *Regulators*.

Regulators are also related with the whole human body, so during the implementation of a facial animation, there is no need to implement such signals.

### 5.2.2 Eye Actions

Eye actions include eye blinking, changes in the direction of eye gaze and changes in pupil size [15].

#### Eye Blinks

Blinking is an important characteristic which should be included in a

realistic animation of speaking faces. Eye blinks are very important part of speech as they serve as a part of response systems. Eye blinks may be intentional or unintentional. We blink our eyes to keep them wet (unintentional), or we blink our eyes to confirm a sentence (intentional). The timing of an eye blink should be synchronized with the speech. The eye might close over one syllable and start opening again over another word or syllable. Blinks can also occur on stressed vowels.

Occurrence of blinks also depends on emotions. During fear, tension, anger, surprise, lying, the amount of blinking increases, whereas it decreases during a concentrated thought.

Observations showed that, eye blinks generally occur at pauses. Listener's eye blinks and head nodding are also synchronized with the speaker's voice. Listener eye blinks frequently take place while nodding.

### **Eye Gaze**

During speech, eyes always move. When we are looking at an object or a person, we scan it from the most important details to the least important ones. Duration and the frequency of scanning depend on social and cultural context of the person looking.

Eye gaze is very important in communication. "Eye contact" is a way of communication which is often used instead of verbal communication. Eye contact degree changes depending on the friendship and trust. If a person is lying, he generally avoids "eye contacting." When a person is thinking, or trying to remember something, eyes generally look upwards.

### **Pupil Size**

The size of the pupil, although not very recognizable, changes in size during the speech depending on the amount of light in the environment. It also changes in size depending on the emotion of the person. If a person is happy, the pupil become larger, however, if s/he is angry, the pupil will be constricted.

# Chapter 6

## Linguistic Issues

### 6.1 Properties of Turkish Included in the System

This study is based on Turkish. We can consider Turkish as a syllable based language. This classification does not depend on the linguistic issues however it reflects the idea behind the implementation of our approach. As in all other natural languages, Turkish also has phonemes which are used to form words. However, we can approximate mouth postures for phonemes by considering letters due to the special property of Turkish. For example, phonemes in the following words are different and hence this results in the different meanings. ‘hala’ which means ‘aunt’ and ‘hala’ which means ‘still.’ But the shapes of the mouth are not visually distinct for these two words. Therefore, phonetic structure of Turkish is not considered since it does not yield visual differences in almost all of the words. We can say that, we can generate phonemes in Turkish by using letter definitions and we can use *letters* as the basic structures.

The shape of the mouth changes according to the syllable being pronounced. In a realistic modeling of speech in Turkish, we need to define these syllables and mouth shapes corresponding to each syllable. However, the number of syllables in a language is quite high making such modeling impossible. On the

other hand, syllables are made up of letters from a small inventory. Furthermore, in Turkish we speak what is written. That is, written form dictates the pronunciation and there is no exception *in theory*. In addition, each word is made up of syllables and each syllable is made up of letters in the Turkish Alphabet. Furthermore, each letter corresponds to a specific mouth posture in the ideal case. This is not the case in English. We cannot achieve a realistic and adequate animation system by defining mouth shapes for each letter in English Alphabet because English speakers use another minimal structure to form words, which is called *phonemes* (English also has syllables, but while pronouncing words, phonemes are generated and they make the visual difference.) Therefore, we should define mouth shape for each phoneme to generate the realistic animation.

Because of these reasons, the system depends on the definitions of mouth postures for each letter. Using these definitions, we can generate all of the words in Turkish. Each letter is associated with a mouth posture and a sound. These mouth postures are formed by altering the parameters of muscles, tongue and rotation angle of jaw. A lip shape is mostly affected by the vowel in a syllable. Table 6.1 shows the classification of vowels in Turkish.

	Low	High
Non-round	a e	ı i
Round	o ö	u ü

Table 6.1: Classification of vowels in Turkish.

To justify our approach we can give the following example: ‘de’ and ‘do’ syllables have different mouth postures due to the characteristics of vowels following ‘d.’ For ‘do,’ mouth is round but for ‘de,’ mouth is flat. So, it is meaningful to define ‘d,’ ‘e’ and ‘o’ mouth shapes first. Then, the model can say both ‘de’ and ‘do’ by interpolating from the posture for ‘d’ to posture for ‘o’ and by interpolating from the posture for ‘d’ to posture for ‘e.’ The characteristics of Turkish allows grouping letters based on similar mouth postures for speech animation rather than defining different mouth postures for each letter. The grouping of letters based on similar mouth postures is given in Table 6.2. These groups may not be reflecting the linguistic facts, however, we focused on the lip shapes generated for each letter, thus, this classification yields generating



the visually distinct mouth shapes. According to this classification, we can generate mouth postures for the following two-letter syllables, ‘de,’ ‘di,’ ‘di,’ ‘do,’ ‘dö,’ ‘te,’ ‘ti,’ ‘ti,’ ‘to,’ ‘tö,’ ‘ne,’ ‘ni,’ ‘ni,’ ‘no,’ ‘nö,’ ‘ey,’ ‘iy,’ ‘iy,’ ‘oy,’ ‘öy’ and other syllables composed of more than two letters. Thus, by using the classification in Table 6.2, we can generate approximately 198 syllables in the form of  $\backslash VC\backslash$ ,  $\backslash CV\backslash$ ,  $\backslash VCV\backslash$  and  $\backslash CVC\backslash$  ( $V$  denotes a vowel and  $C$  denotes a consonant) using only three letter definitions. This classification is similar to the classification for consonants given in [2]. First column is the classification used in the system and the second column is the classification of these sounds according to their phonemic structures as described in [2]. The classification used in the system mainly depends on the lip shapes during the pronunciation of these sounds. In Table 6.2, letters in the same row have similar or the same mouth postures.

row	Classification	
	used in the system	according to linguistic issues
1	a	
2	b, m, p	b, m, p
3	c, ç, j	c, ç, j, ʃ
4	d, t, n	d, l, n, r, s, t, z
5	e, i, i, y	y
6	f, v	f, v
7	h, k, g	h
8	l	see fourth row
9	o ö	
10	r	see fourth row
11	s, ş, z	see third and fourth rows
12	u, ü	
13	see seventh row	k, g

Table 6.2: Classification of letters based on similar mouth postures.

## 6.2 Coarticulation

As in all other languages, a syllable may affect the previous or next syllable or syllables. This is another important problem and is known as *coarticulation*. Coarticulation refers to changes in the articulation of a speech segment depending on the preceding (backward coarticulation) and upcoming (forward

coarticulation) [4]. An example of backward coarticulation is a difference in articulation of a final consonant in a word, depending on the preceding vowel, e.g., boot vs. beet (or 'bot' and 'bit' in Turkish.) 't' in the first word will be round and 't' in the second word will be flat. An example of the forward coarticulation is the lip rounding at the beginning of word 'stew' (or 'kol' in Turkish.)

Studies have been made in order to find to what degree coarticulation is important. If a consonant is between two vowels, coarticulation has a very high effect [4].

# Chapter 7

## Speech Animation System

The facial animation system developed in this study is given in Figure 7.1. The system has five main components:

1. input text,
2. database,
3. input text parser,
4. emotional signal, expression or letter; generated by parser, and
5. facial animation display system.

### 7.1 Synchronizing Speech with Expressions

Speech animation is not very impressive without additional features like expressions. Since we are interested in realistic speech animation, we should be able to generate necessary expressions and emotional overlays which may occur at any time in the speech. In addition, these important characteristics should be synchronized with the speech to achieve satisfactory results. Synchronization process can be performed in two ways [22]: first guessing the desired expressions and emotional overlays from the content of the text to be spoken,

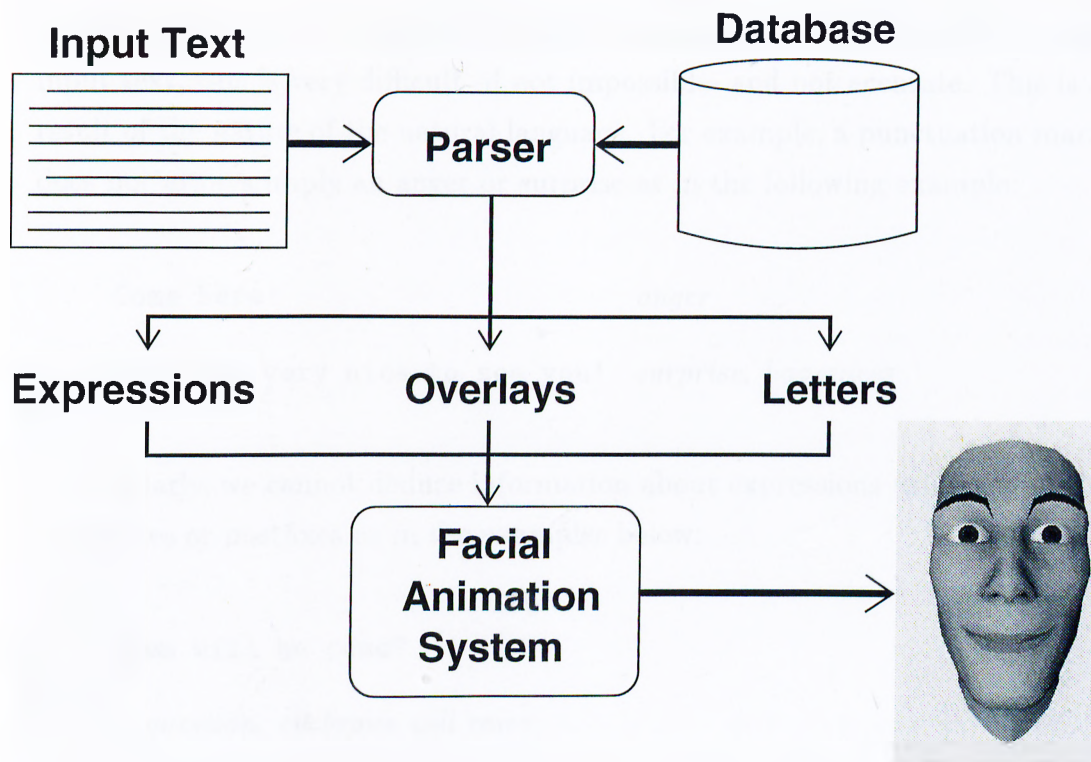


Figure 7.1: The facial animation system

and second, inserting special marks into the text which imply the timing of the expression or emotional overlay to be generated. These techniques will be discussed in the following sections.

### 7.1.1 Guessing Expressions from the Text

During speech, we generate expressions as a result of our feelings. Expressions may change and help interpretation of words. If we want to dictate a conversation, we usually use punctuation marks to show the feelings of the speaker. For example, we put an *exclamation mark* at the end of a sentence to imply that the speaker is *shouting*. We also use some keywords to show feelings. For example, we say *good* to present our positive ideas. Sometimes, postfixes (or prefixes) may help present different feelings. For example, we use *interesting* to show our surprise and we use *uninteresting* to show a negative idea. These examples give an idea about deducing expression information from the sentence using prefixes.

Although there are ways to generate expressions using some *keys* in the input text, this is very difficult -if not impossible- and not accurate. This is a result of the nature of the natural language. For example, a punctuation mark does not always imply an anger or surprise as in the following example:

Come here!

*anger*

Oh, it's very nice to see you!

*surprise, happiness*

Similarly, we cannot deduce information about expressions using keywords or prefixes or postfixes as in the examples below:

When will he come?

*A question, eyebrows will raise*

I don't know when he will come.

*A negative sentence with a question word as a key*

Thus, we cannot say that, when *when* appears, raise eyebrows; or nod head. The content of the sentence and the feelings may completely differ.

Using the ideas above, we can say that it is really very difficult to generate information about the expressions and feelings of a person by considering only punctuation marks, keywords, etc. Natural language allows changing the meaning of a word with expressions and different intonations. Therefore, it is feasible to generate these expressions and synchronize them with the speech by inserting necessary tags to the proper positions in the text, so that we can achieve realistic synchronization of speech with expressions.

### 7.1.2 Using Tags for Synchronization

As clearly stated above, using punctuation marks or other features of a natural language cannot give a unique result to obtain necessary expressions. Thus, it is

useful and easy to determine the timing of an expression in speech as input. For this purpose, tags are inserted in necessary places in the input text to generate an expression or to remove an expression from the face. Each expression is represented by a tag and intensity level. An expression is allowed to be in one of ten intensity level which shows the strength of the expression. For example, if the intensity level for *happiness* expression is 1, this is no more than a satisfaction; however, if the level is 10, this means that the person is very happy, may be laughing loudly. The highest level and the lowest level definitions of an expression are stored in database and the intermediate levels are created by linearly interpolating between the highest and the lowest intensity levels. Thus, if a muscle is contracted to 0.3 units in lowest level and to 1.3 units in the highest level, the intensity level of 4 is generated by contracting this muscle to 0.7 units.

As well as expressions, expression overlays are also synchronized with speech by using tags. For each overlay, a tag is reserved. There is a difference between expression tags and overlay tags. Expression tags have only one parameter and overlay tags may have one or two parameters. Furthermore, expression tags have intensity levels but overlay tags have no intensity levels. Instead, overlay tags directly control the agent that they are responsible for. For example, when EYEGAZE tag appears, both of the eyes will be looking at a direction which is specified by the parameters of this tag.

Facial expressions are specified using the following format:

`\b{expr level}` starts an expression `expr` of degree `level`. If this expression is set before, `level` is used to increase the degree of the expression, instead of setting expression to degree of `level`.

`\e{expr level}` ends or decreases the degree of an expression by `level`. If `level` is -1, expression is completely removed from the face.

An overlay tag is specified using the following format:

`\b{overlay parameter1 parameter2}` sets the specified expression overlay `overlay` to face according to the specified parameters.

There is no need for a `\e{...}` tag for an overlay since it can be reset by updating the parameters properly. However, an expression needs such a tag which is the real situation in our life. For example, when we speak, we may be very excited at the beginning of the expression. But according to the content and the flow of the conversation, we can get more or less excited. The `\e{...}` tag helps simulating such behavior.

Using this scheme, we can also blend facial expressions with overlays. These tags need not to be nested. Let us say, a text contains two tags for happiness and raising the eyebrow. We can set happiness first and then we can raise eyebrows while the degree of happiness is reduced, but not removed. Such a blending operation is given in the following example:

```
\b{HAPPINESS 3} merhaba nasılsın? \e{HAPPINESS 1}
\b{EYEBROW 8} yeni araban nasıl? \e{HAPPINESS -1}
```

During the first sentence, `HAPPINESS` expression is set on the face and the degree of the expression is 3. This expression pulls the corners of the mouth to give a *smiling* effect. After the first sentence is completed, the degree of expression is reduced by 1 by using `\e{HAPPINESS 1}` tag. At the same time, another expression overlay is also set on the face which is `EYEBROW`. The degree of `EYEBROW` is 8. This overlay raises eyebrows to give a *surprised* or *excited* effect. Thus, at that point, we have an expression and an overlay which are blended. The tag at the end of the second sentence, removes `HAPPINESS` expression from the face. The face still has raised eyebrows. As explained before, eyeblinks occur at pauses. To make eyes blink, we should add following tag sequences to the proper places:

```
\b{BLINK <anynumber> 3}\b{BLINK <anynumber> 2}
```

The first parameter is not used if the second parameter affects both eyes.

By manually inserting tags where necessary, we can manage the synchronization of facial expressions with speech. Furthermore, we can achieve some

other behaviors related with speech. For example, we can say a word, or a part of the word with our eyes closed. We can assign special meanings to necessary parameters of the tag to get such an effect. The tag `BLINK p1 p2` allows such an effect. The following setting will say `merhaba` with left eye closed.

```
\b{HAPPINESS 3}
\b{BLINK 1 1}merhaba\b{BLINK 1 0} nasIlIsIn?
\e{HAPPINESS 1}
\b{EYEBROW 8} yeni arabana nasIl?
\e{HAPPINESS -1}
```

## 7.2 Input Text

The input text is a plain ASCII text composed of meaningful characters, such as letters, parser-specific characters and punctuation marks.

- **Letters:** We only allow lowercase in the input file since certain uppercase letters have special meanings. For special Turkish letters, like ‘ş’, corresponding uppercase ASCII character is used, that is ‘S’. The other substitutions used for special Turkish letters in the input file are as follows: ‘C’ for ‘ç,’ ‘G’ for ‘ğ,’ ‘I’ for ‘ı,’ ‘O’ for ‘ö,’ and ‘U’ for ‘ü.’ Valid letters in the input file are as follows:

```
a b c C d e f g G h I i j k l m n o O p r s S t u U v y z
```

- **Parser-specific characters:** These characters have certain uses and are not meaningful to the reader. These characters are interpreted by the text parser and used to synchronize expressions with the speech. Parser-specific characters are:

- ‘\’, which means that a tag will begin or end depending on the character following it, which can be either ‘b’ or ‘e.’



- ‘{,’ ‘}’ which contain expression names, overlay names and their parameters.
- **Punctuation marks and whitespace:** Whitespace is used to distinguish words and valid punctuation marks are ‘!’, ‘,’, ‘.’ and ‘?’.

## 7.3 Database

The database of the system is divided into three parts:

1. **Letters file:** This file stores letter definitions. Each letter definition needs the following parameters:
  - the *name* of the letter,
  - muscle contraction values,
  - jaw rotation angle and
  - tongue parameters for each section.
2. **Expressions file:** This file stores six basic universal expression definitions together with some other basic expression units. Fields of that file include:
  - a *key* representing the expression,
  - muscle contraction values for the lowest intensity level,
  - muscle contraction values for the highest intensity level,
  - jaw rotation angle for the lowest intensity level and
  - jaw rotation angle for the highest intensity level.
3. **Emotional overlays:** This file stores emotional overlay names. Only the names are stored since parameters and overlays are controlled within the program by specific procedures. Each overlay is also generated according to the parameters written in the input file. These parameters are explained in Table 7.1.

- ‘{,’ ‘}’ which contain expression names, overlay names and their parameters.
- **Punctuation marks and whitespace:** Whitespace is used to distinguish words and valid punctuation marks are ‘!’, ‘,’, ‘.’ and ‘?’.

## 7.3 Database

The database of the system is divided into three parts:

1. **Letters file:** This file stores letter definitions. Each letter definition needs the following parameters:
  - the *name* of the letter,
  - muscle contraction values,
  - jaw rotation angle and
  - tongue parameters for each section.
2. **Expressions file:** This file stores six basic universal expression definitions together with some other basic expression units. Fields of that file include:
  - a *key* representing the expression,
  - muscle contraction values for the lowest intensity level,
  - muscle contraction values for the highest intensity level,
  - jaw rotation angle for the lowest intensity level and
  - jaw rotation angle for the highest intensity level.
3. **Emotional overlays:** This file stores emotional overlay names. Only the names are stored since parameters and overlays are controlled within the program by specific procedures. Each overlay is also generated according to the parameters written in the input file. These parameters are explained in Table 7.1.

## 7.4 Input Text Parser

The parser of the system reads the input file and distinguishes expressions and letters from the input file. Then, it updates necessary parts of the face using the parameters in the input file. That is, it generates necessary parameters for the system to get desired face posture.

Syntax of the input file is very straightforward. As mentioned before, input file will contain only letters, space character, '\', '{' and '}'.' '\' denotes the beginning of a tag, and the tag and its parameters are placed in braces. Complete listing of the tags used in the system and their parameter definitions are given in Table 7.1.

The parsing algorithm is given in Figure 7.2 [22].

```

While not all of the text is processed
  Read a character
  If a tag is beginning // "\" is read
    Read tag           // name and degree of expression
    If degree is -1,
      Remove expression from the face
    else
      Set face according to expression with specified degree
  If a valid character // a letter or a punctuation mark
    If this is the first character to say
      Set face using current expression and letter settings
      Display face
    else
      for each in-between
        Calculate vertex coordinates using cosine interpolation
        Display face
  Store vertex coordinates for future reference

```

Figure 7.2: The algorithm for speech animation.

TAG	First Parameter	Second Parameter	Effect
SURPRISE	$p1$	no effect	Sets the face to the expression <i>surprise</i> to the intensity level given by $p1$
DISGUST	$p1$	no effect	Sets the face to the expression <i>disgust</i> to the intensity level given by $p1$
FEAR	$p1$	no effect	Sets the face to the expression <i>fear</i> to the intensity level given by $p1$
ANGER	$p1$	no effect	Sets the face to the expression <i>anger</i> to the intensity level given by $p1$
HAPPINESS	$p1$	no effect	Sets the face to the expression <i>happiness</i> to the intensity level given by $p1$
SADNESS	$p1$	no effect	Sets the face to the expression <i>sadness</i> to the intensity level given by $p1$
SMILE	$p1$	no effect	Sets the face to the overlay <i>smile</i> to the intensity level given by $p1$
EYEBROW	$p1$	no effect	Sets the face to the overlay <i>eyebrow</i> to the intensity level given by $p1$
EYEGAZE	$p1$	$p2$	Eyes will look at $p1$ degrees left (right) and $p2$ degrees up (down) depending on the sign of the parameter.
PUPILSIZE	$p1$	no effect	Size of the pupil will be $p1$ .
IRISSIZE	$p1$	no effect	Size of the iris will be $p1$ .
BLINK	$p1$	$p2$	$p1$ is used to distinguish left and right eyes. If $p1 = 0$ , affect the right eye If $p1 = 1$ , affect the left eye If $p2 = 0$ , open corresponding eye If $p2 = 1$ , close corresponding eye If $p2 = 2$ or $p2 = 3$ , $p1$ becomes nonsense since these cases affect both eyes If $p2 = 2$ , open both eyes If $p2 = 3$ , close both eyes

Table 7.1: Available expressions and overlays with the definitions of their parameters.

## 7.5 Facial Animation Display System

Expression tags, overlay tags, letters and punctuation marks may appear in any order in the input file which yields the realistic synchronization of speech with expressions.

When the parser recognizes a tag for expression or emotion, it sets the activation parameter of that expression or emotion in the system. However, displaying an emotion or expression is delayed until a letter is recognized. A copy of the face with current expression or emotion parameters is generated and this face is displayed by interpolating from the existing face.

The system uses double buffering mechanism of OpenGL. Once an in-between is created and the next one is known, first frame is displayed and the next frame is written into a second buffer. When we swap these display buffers, we get a flickering-free animation. After swapping two buffers, the background one is filled with the next in-between.

The system is capable of displaying differently rendered face shapes. There are three main versions of the face model:

- *Wire-framed*: The lines of polygons in the model are displayed.
- *Flat shading*: Pixels inside a polygon have the same color without any variation. This is a very low-cost rendering method but not acceptable in terms of realistic speech animation.
- *Phong shading*: This method is a shade interpolating method. Each normal is calculated at each vertex of a polygon and for the other points in the polygon, these normals are interpolated. Thus, a normal is created for each point and a new shade is calculated for each point. This method eliminates some bad visual effects, such as Mach Bands which are encountered when Gouraud shading is used (In Gouraud shading, color intensities are linearly interpolated instead of normals) [8].

## 7.6 Animation

### 7.6.1 Keyframing

To generate a realistic animation of a speaking face model, keyframing based on parameters of the muscles around the mouth and jaw rotation parameters are used. Each keyframe of the animation sequence includes a properly positioned mouth and face shape generated according to the current settings of the expressions and letter to be spoken. In Turkish, letters are pronounced by strict rules. Hence, the database for mouth shapes can be based on letters. The inbetweens are generated by using the cosine interpolation technique which is explained below.

### 7.6.2 Interpolation Techniques for Keyframed Animation

There are three popular interpolation schemes for animation:

- *Linear interpolation*: Uses a simple formula to calculate the time of displaying an in-between. Intermediate frames occur after the same  $\Delta t$  time. This approach is not useful for facial animation.

$$tB_j = t_1 + \Delta t j$$

where  $j = 1, 2, 3, \dots, n$  for  $n$  in-betweens.

- *Sine interpolation*: Uses a sine interpolation function to calculate the time of displaying an in-between. This scheme is also known as *deceleration*.

$$tB_j = t_1 + \Delta t \cdot \sin \frac{j\pi}{2(n+1)}$$

where

$$\theta = \frac{j\pi}{2(n+1)}, \quad 0 < \theta < \frac{\pi}{2}$$

and  $j = 1, 2, 3, \dots, n$  for  $n$  in-betweens

- *Cosine interpolation*: Uses a cosine interpolation function to calculate the time of displaying an in-between. This scheme is also known as *acceleration* and fits well to the facial animation. We used cosine interpolation scheme in our system to model the visco elastic behavior of the skin.

$$tB_j = t_1 + \Delta t(1 - \cos \frac{j\pi}{2(n+1)})$$

where

$$\theta = \frac{j\pi}{2(n+1)}, \quad 0 < \theta < \frac{\pi}{2}$$

and  $j = 1, 2, 3, \dots, n$  for  $n$  in-betweens

We applied cosine interpolation technique to the whole face by the algorithm in Figure 7.2 to achieve realistic skin behavior.

## 7.7 Implementation

The facial animation system is implemented on Iris Indigo Silicon Graphics<sup>1</sup> running under the UNIX<sup>2</sup> operating system. The workstation has a single 100 MHz IP20 CPU and 32 Mbytes main memory. Facilities of OpenGL were used to render and display the face.

The usage of the system depends on key-bindings. Each feature of the system is bound to keys. The following are the main functions of the system:

- Updating muscle contraction values,
- changing rendering style (wire-frame, flat shading, Phong shading),
- creating, saving and loading a specific letter,
- creating, saving and loading a specific expression,
- creating, saving and loading a specific expression overlay as an expression unit,

---

<sup>1</sup>Silicon Graphics is a registered trademark of Silicon Graphics Inc.

<sup>2</sup>UNIX is a registered trademark of AT&T Bell Laboratories

- changing tongue parameters,
- changing the size and positions of pupil, iris, eyeball, and changing the direction of eyegaze,
- jaw rotation,
- observing the effect of changes in parameters during the update, and
- changing the environmental settings, such as the position of the light, direction of the face, etc.

### 7.7.1 Performance Issues

Table 7.2 presents initialization times for the system. Rendering times for wire-frame, and Phong shading are given in Table 7.3. All images are 600x400 pixels.

Components included in the system	Number of polygons	Creating data structure	Loading and composing face structure
Only face polygons	888	1.39	12.25
Teeth added	1620	2.33	40.81
Tongue added	1704	2.60	48.19

Table 7.2: Initialization times for the facial animation system. Times are given in *seconds*.

Components included in the system	Number of polygons	Wire Frame	Phong Shading
Only face polygons	888	0.90	1.04
Teeth added	1620	0.90	1.07
Tongue added	1704	0.90	1.41

Table 7.3: Performance issues. Rendering times are measured to render a frame of an animation. Times are given in *seconds*.

As given in Table 7.3, rendering algorithm works equally efficient for different number of polygons. There are no big gaps between display times of



the model. However, creating the data structure and loading parameters and forming composite face structure take long time if the number of polygons is increasing. This is because of the algorithm used for composing the face structure.

The structure of the face is composed of polygons whose vertices can be shared. The algorithm initially finds all neighbors of a vertex, thus, such long composition times occur. For each vertex, all vertices are checked if their coordinates match. If so, these vertices are recorded as neighbors. Since this operation is repeated for all vertices, the bigger number of polygons results in the longer facial structure creation times.

Expressions and overlays increase the displaying times of an animation frame. Because in each frame, each expression and/or expression overlay which is active will be set on the face by applying the formulation given in Section 4.3. Each expression and/or overlay should be activated on each frame since they can be mixed in the system. As explained before, each expression dictates specific contraction values for each muscle. In implementation, if the contraction value of a muscle is not zero, i.e. corresponding muscle takes place on that expression, it will take a time to activate that muscle. Thus, the performance of the facial animation system is directly affected by the number of muscles activated in a keyframe. The processing time for creating a keyframe varies from 5 seconds to 14 seconds, the average is 9.5 seconds. This is really a very long time to display all frames of an animation sequence. As a result, each frame in the animation sequence are stored in the disk. Writing into a disk file may take an average of 3 minutes for  $600 \times 400 = 240000$  pixels in TGA<sup>3</sup> format [21].

---

<sup>3</sup>TGA is a trademark of Truevision, Inc.

# Chapter 8

## Results

In this chapter, some example frames from the example animation sequence describing the salient features of the animation system are given.

```
\b{BLINK 0 3}\b{BLINK 0 2}
```

```
\b{HAPPINESS 3}merhaba nasIlIn ?\e{HAPPINESS 1}
```

```
\b{BLINK 0 3}\b{BLINK 0 2}
```

```
\b{EYEBROW 8}yeni araban nasIl\e{EYEBROW -1}?
```

```
\b{BLINK 0 3}\b{BLINK 0 2}
```

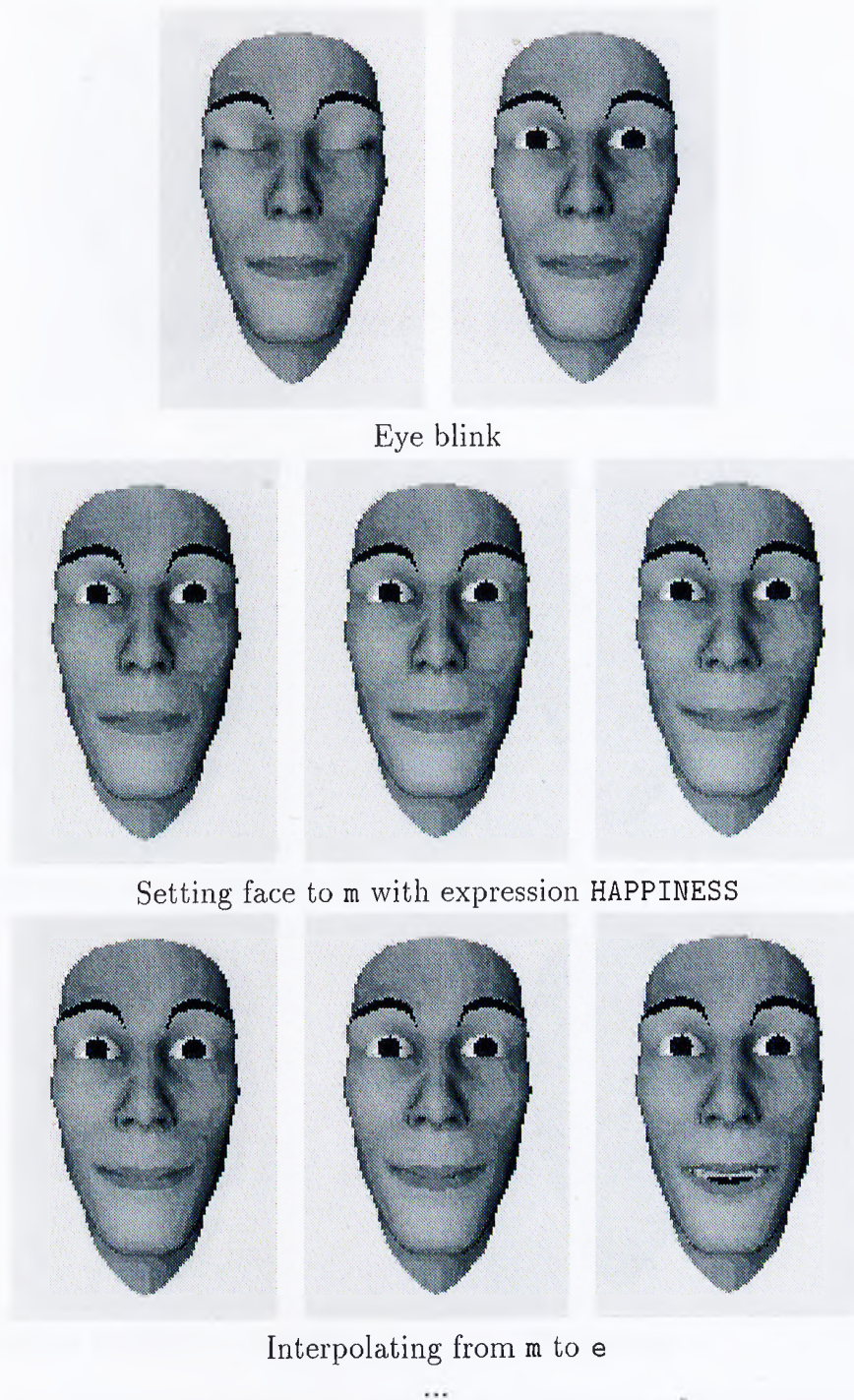


Figure 8.1: Still frames from the animation sequence of the example.

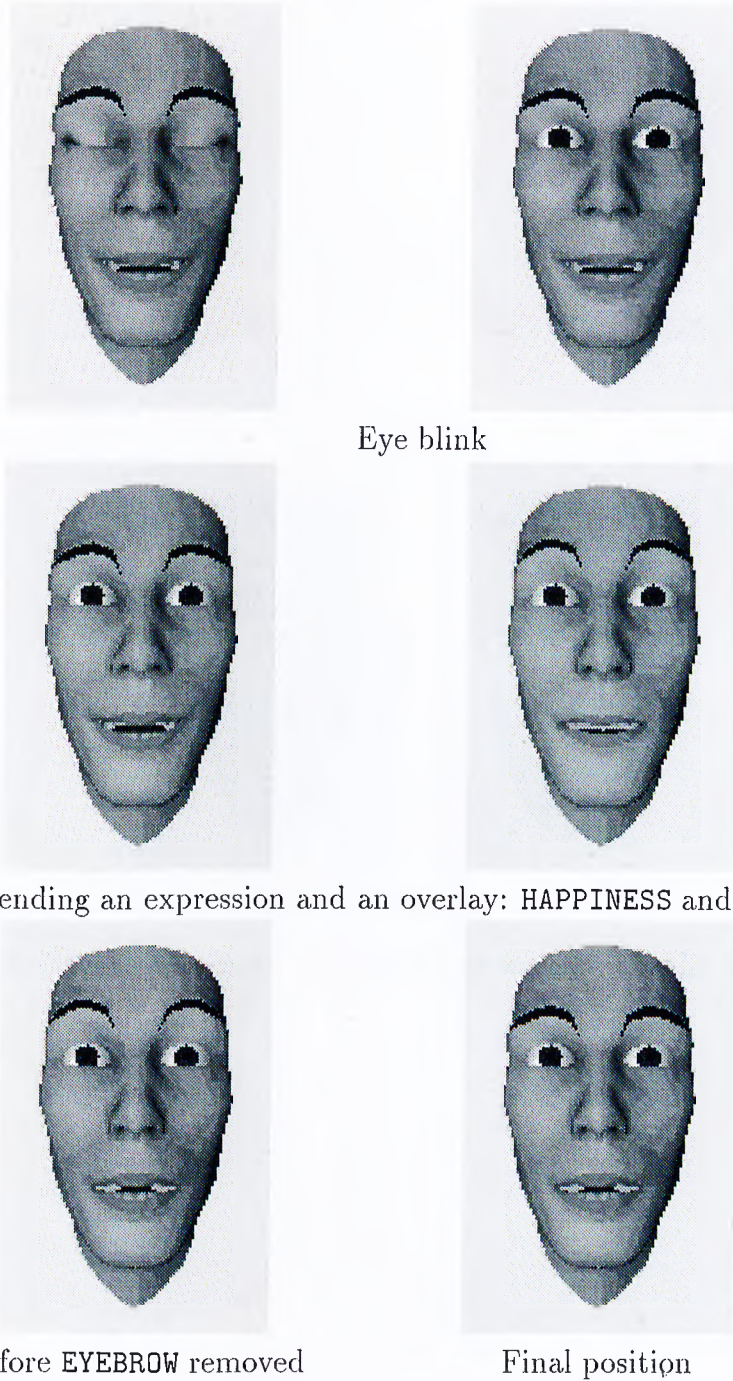
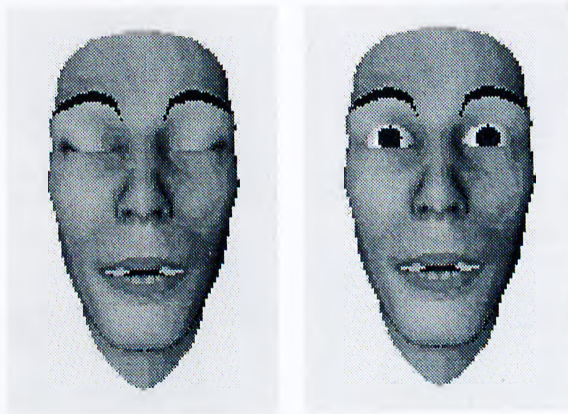


Figure 8.2: Still frames from the animation sequence of the example (continued.)



Final eye blink

Figure 8.3: Still frames from the animation sequence of the example (continued.)

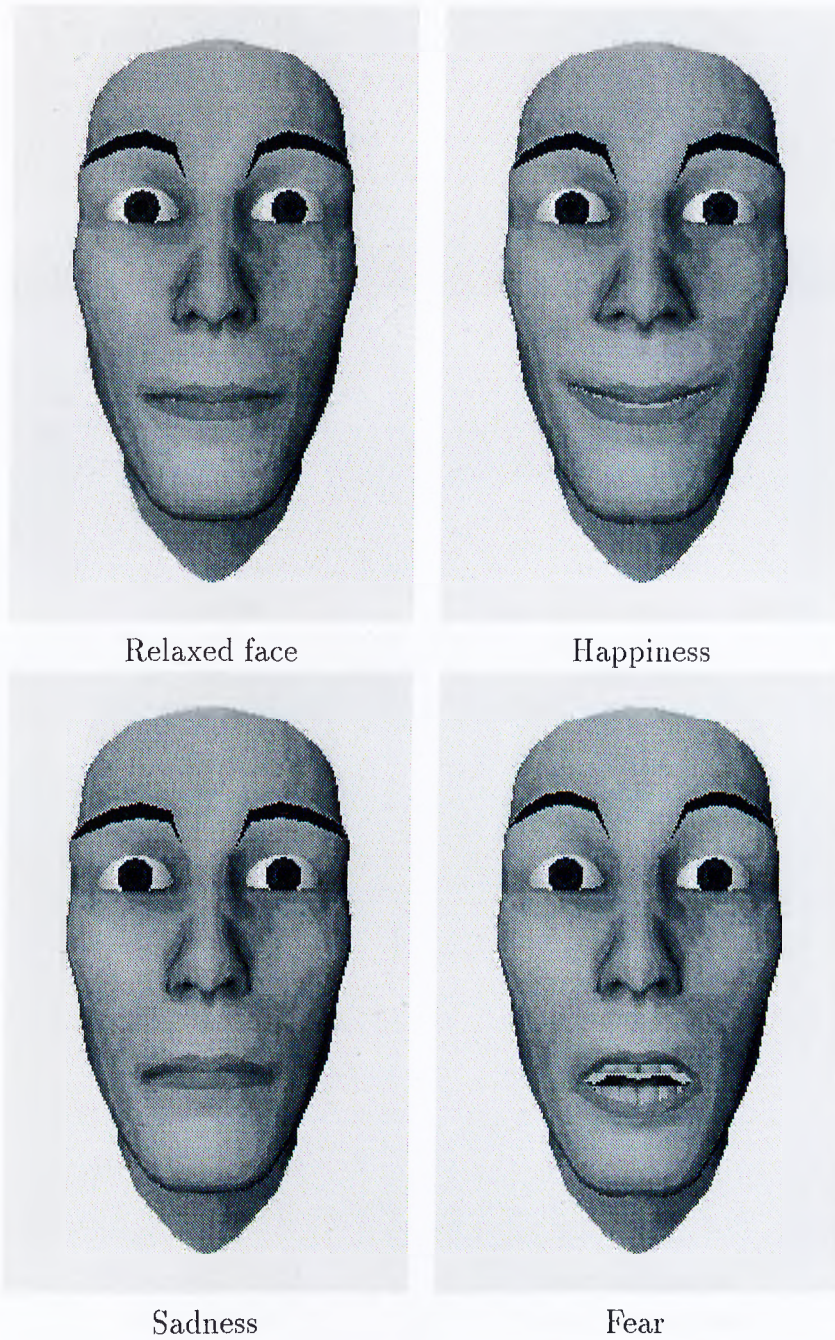


Figure 8.4: Expressions and overlays.

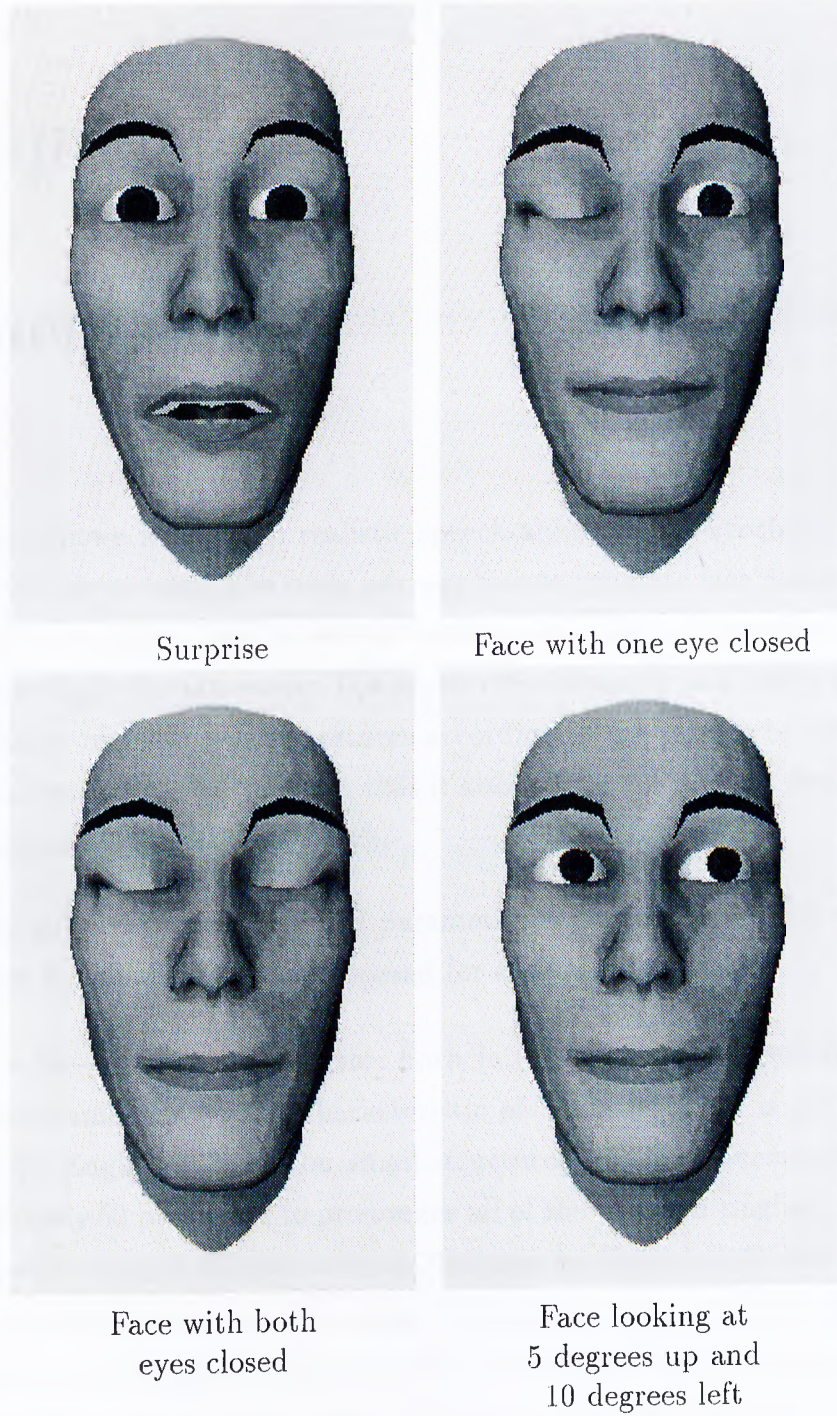


Figure 8.5: Expressions and overlays (continued.)

# Chapter 9

## Conclusions

In this study, we focused on realistic speech animation of synthetic faces according to a given text. The main purpose was to generate face postures which are visually realistic. That is, we did not focus on realistic behavior of human skin nor realistic human scene. The system developed in this study is capable of generating realistic mouth postures according to the text to be spoken. An additional feature of the system is that it also synchronizes other visual effects, such as expressions with the speech.

Physically-based modeling and parameterization techniques are combined and a new hybrid approach is proposed for speech animation.

The work is based on Turkish. Each letter in a word corresponds to an animation frame due to the characteristic of Turkish. This is not the case in English. English is based on small structures, called *phonemes*. We need approximately 50 phonemes to pronounce all of the words in English. However, there are 18 visually distinct mouth postures in English [15], we need less number of mouth postures in Turkish. We need at most 29 letter definitions to generate convincing face postures. We also classified these letters according to their shape; thus, we need only 13 groups of letters to generate all of the words. This is a very significant reduction.



Since facial animation is not only animating lips with proper mouth postures nor animating only emotional changes, a mechanism is developed to synchronize such realistic behaviors with speech. We proposed a mechanism to synchronize expressions and emotional overlays with speech. Since it is very difficult, if not impossible, to deduct feeling information from the text, some tags identifying the feelings and other emotional behaviors of the speaker are inserted into the proper positions in the text. Since the intensity of feelings are variable, an additional parameter representing the level of feeling is added.

Other facial agents such as teeth, tongue, etc. exist in the system. Each agent is modeled separately. A complex model for tongue is developed. According to our approach, the tongue is defined as 4 sections. The shape of the tongue is defined by altering the parameters of each section.

Although this work gives very realistic results, the following can be implemented to make the system generate more realistic results:

- i. The shaded face model is a synthetic face model. Texture mapping could be implemented to make the animations convincing.
- ii. Ears could be implemented to increase realism.
- iii. Hair modeling which is a quite active research area could be added. Realistic behavior of the hair could be implemented so that, when nodding or turning the head, the hair can move as if it were a real human hair.
- iv. Coarticulation which is another active research area could be implemented. As explained before, coarticulation is very important especially in determining mouth postures according to letters. Because, each letter may affect previous or following letters' mouth postures which is the case in real life.

# Appendix A

## Implementation

Components of the speech animation system are given in Figure A.1. The system is built on top of the basic facial animation software developed by [15].

1. Major modules are represented by a rectangle.
2. Header file is represented as rectangle with rounded corners.
3. Disk files are represented by common disk figure.
4. Arrows denote the flow of data or direction of function calls.

Before explaining the components, let us describe the meanings of arrows.

- $M1 \rightarrow M2$  denotes that M1 uses functions of M2. If the arrow is bidirectional ( $\leftrightarrow$ ), M1 and M2 are calling functions from each other.
- $M1 \leftarrow H1$  denotes that M1 uses data structures from the header file H1.
- $M1 \leftarrow D1$  denotes that M1 reads disk file D1.
- $M1 \leftrightarrow D1$  denotes that M1 both reads from and writes into disk file D1.

Let us briefly explain each component and its functionality.

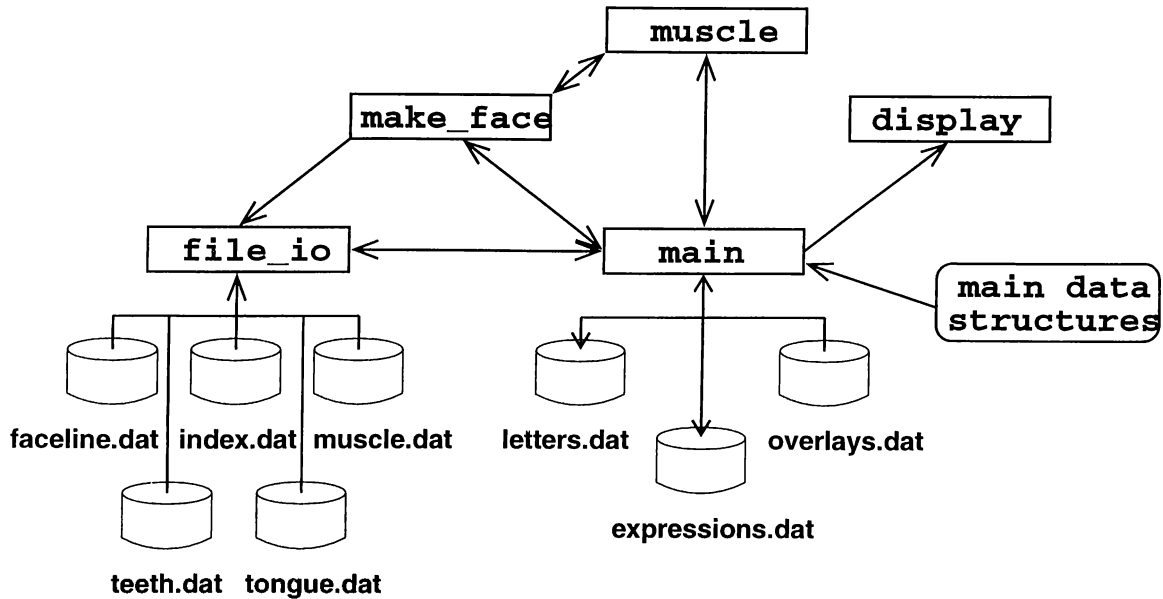


Figure A.1: Components of the facial animation system.

## A.1 Main Data Structures

As mentioned before, the face model is composed of polygons, each polygon is a triangle and hence a polygon has three vertices. Polygonal structure of the face is stored in the following data structure:

```

struct VERTEX {
    float xyz[3];      // current coordinates of the vertex,
                     //   modified during execution
    float nxyz[3];    // original coordinates of the vertex,
                     //   never modified
    float sxyz[3];    // previous coordinates of the vertex;
                     //   used in interpolation
    int    np;        // number of polygons associated with
                     //   this vertex
    int    plist[200]; // list of indices of polygons associated
                     //   with this vertex
    float norm[3];    // three dimensional normal of a vertex
    int    vtag;      // tag of the vertex; can be one of
                     //   UPPER, LOWER, BOTH, NONE,

```

```

//      JAW_ROT, EYEBLINK
}

```

A polygon is composed of three vertices as in the following structure:

```

struct POLYGON {
    VERTEX *vertex[3]; // vertices of polygon
}

```

Expressions and expression overlays should be stored in separate data structures:

```

struct EXPRESSION {
    char  name[20]    ; // name of the expression
    int   degree     ; // degree of expression
    float m1[40]     ; // an expression vector, contains muscle
                        // contraction values for the lowest
                        // degree of expression
    float mh[40]     ; // an expression vector, contains muscle
                        // contraction values for the highest
                        // degree of expression
    float j1, jh     ; // jaw rotation angles for low and high
                        // degrees of expression
}

struct EXPR_OVERLAY {
    char  name[20]    ; // name of the expression overlay
    float parameter[2] ; // parameters of overlay
    int   degree     ; // intensity level of overlay
}

```

Structure of the tongue is stored in a data structure containing all parameters as explained in Section 4.6.1.

Muscles in the system are stored using the muscle parameters explained in Section 4.3. There are three additional parameters: The name and tag of the muscle and a flag which denotes whether the muscle is active or not.

```
char name[40]    ;    // name of the muscle
int  mtag       ;    // tag of the muscle
int  active     ;    // flag for the activity of muscle
```

The main data structure for the whole face is given below. Since the face is updated according to parameters, we need to keep all of the data related to the face for each frame. The major fields in the face data structure are defined as follows:

```
struct HEAD {

    int      npolygons      ; // total number of polygons
    POLYGON **polygon      ; // pointer to the polygon list

    float    eyeballang[2] ; // rotation angle of eyeballs about
                          //      x & y axes

    int      nteethpolygons ; // number of polygons in teeth

    float    jawang        ; // rotation of the jaw
    float    rotvec[3]     ; // normalized jaw rotation vector

    int      nmuscles      ; // number of muscles in the face
    MUSCLE  **muscle       ; // pointer to the muscle list

    int      nexpressions  ; // number of expressions
    EXPRESSION **expression ; // pointer to an expression vector

    int      nexpr_overlays ; // number of expression overlays
    EXPR_OVERLAY **expr_overlay ; // pointer to an overlay vector
}
```

Necessary fields of face data structure are updated by the caller function. Thus, the current face will contain the most recent parameters for the frame.

## A.2 Operation Flow

MAIN module is responsible for all of the operations. It manages the flow of the execution. The flow of the program is given in Figure A.2.

1. Perform initializations
  - 1.1. Initialize windows
  - 1.2. Initialize graphics parameters for OpenGL(tm)
  - 1.3. Initialize face structure
2. According to the key pressed, switch to corresponding mode

Figure A.2: Flow of the program.

Steps 1.1 and 1.2 are trivial and they do not need to be explained. Majority of the work is performed by steps 1.3 and 2 and these steps will be explained below.

### 1.3. Initialize face structure

Initialization of the face is performed by `MAKE_FACE` module. Creation of the face data structure dynamically and assembling the face model are done by `MAKE_FACE` module. Before assembling the face model, `MAKE_FACE` module calls loader modules of `FILE_IO` module to load all parameters. Loader modules read the following input files and update corresponding fields in the face structure:

- `index.dat`
- `faceline.dat`
- `teeth.dat`
- `tongue.dat`

After reading these files, the facial structure is created, but not assembled yet. After initializations completed, the structure will be assembled using the explanations below. There are two data files left. One of them is `muscle.dat` to get muscle information and `rotaxis.dat` to get jaw rotation axis coordinates.

Parameters and information for the speech animation are ready now. After reading all of the parameters, `MAIN` module calls the face assembling routine from `MAKE_FACE` module to compose the face.

Since all of the data about assembling the face is ready, it is necessary to find out which polygons are associated and which tag their vertices will get.

The algorithm in Figure A.3 is used to find all neighbors of a vertex.

```

for each polygon (i) in the system
  for each vertex of this polygon      // for three vertices
    for each polygon (j) in the system
      if (coordinates of polygon i matches
          the coordinates of polygon j)
        add polygon j to neighborhood list of polygon i
        increase the number of neighbors by 1

```

Figure A.3: Assembling the face data structure.

## 2. Switch to corresponding mode

There are seven operation modes of the program:

- *Expression Mode*: In expression mode, the user can save the current face configuration as an expression. S/He can also load an existing expression and may update it.
- *Eye Parameters Mode*: In this mode, the user can change the size and place of eyes. S/He can save these as new expressions.
- *Eye Gaze Mode*: In this mode, the user can change the direction of eye gaze. S/He can save the direction as a new expression.

- *Expression Overlay Mode*: This mode allows user to see the effects of overlays on the face. User is not allowed to update or change the parameters of an overlay permanently; since overlays are stored by names only. Overlay parameters are directly given as parameters in animation text and hence there is no use of storing, for example, a closed eye, an eye looking at a specified direction or an eye with big pupil and iris.
- *Tongue Parameters Mode*: The user is allowed to change the parameters of the tongue. This mode is an auxiliary mode for the user as *Expression Overlay Mode*. These two modes are just to present effects of changes in the parameters of the tongue, etc. After observing these effects, user will decide the parameters and s/he will set these parameters when defining a letter or an expression.
- *Letter Definition Mode*: After user set parameters, s/he can save the current configuration of the face as a letter definition. S/He can also load existing letter definitions and update any of them.
- *Animation Mode*: User has full control over animation. S/He can use controls below and can change the flow of the displayed animation frames. Allowed controls are as follows:
  1. See next inbetween: Displays next interpolation step of the face.
  2. See previous inbetween: Displays previous interpolation step of the face.
  3. Pass to next letter: Displays the next letter. When user wants to skip some parts, s/he will use this control.
  4. Manually saving the current face configuration: Saves the current face figure into disk file named: <filename>.tga. <filename> represents the frame number of the current figure and calculated by the system.
  5. Display all of the animation step by step: System continues displaying frames. User will not do anything but observe the changes. If desired, the system can write each inbetween into a disk file as named above.



Tracing over the animation step by step allows controlling and debugging the animation sequence. User can also capture inbetweens and keyframes with the help of traceability.

## A.3 Explanation of Modules

### MAIN Module

This module is responsible for controlling the flow of the program and this is the major module as the name implies.

### MAKE\_FACE Module

This module is responsible for assembling the structure of the face. Initialization of the facial structure and its components is performed by this module.

### MUSCLE Module

This module is responsible for handling muscle actions. Deformation of the mesh structure of the face by activating a muscle is handled by this module.

### DISPLAY Module

As the name implies, all of the display action is taken here. This module is responsible for generating a properly shaped and shaded face. Thus, calculations for normals are performed here. Specific graphics procedures of OpenGL<sup>tm</sup> are called by this module.

## FILE\_IO Module

Loader modules of the face data and all file input/output operations are handled by this module. This module mainly takes place in the initialization phase.

### A.4 Database and Structure of Files

There are 8 main files associated with the system:

- *faceline.dat*: This file contains three dimensional coordinates of vertices in the system.
- *index.dat*: Contains the *rule* of assembling polygons. This file is used as a rule file to determine which vertices make which polygon.
- *muscle.dat*: Contains parameters of each muscle. Each muscle is represented by a pair and it has the following arguments:
  1. muscle name,
  2. tag of the muscle (LOWER, UPPER, BOTH, etc.),
  3. coordinates of the muscle head and muscle tail,
  4. fall start and fall finish radii,
  5. zone of influence as angle in degrees and
  6. muscle clamping value
- *teeth.dat*: Teeth polygons are written in this file.
- *tongue.dat*: Tongue base and parameters for each section of the tongue are written in this file.
- *letters.dat*: Definitions for each letter are written in that file. An example entry in this file is given in Figure A.4. Parameters are as follows:
  1. *name* of the letter,

2. muscle contraction values,
  3. rotation angle of jaw and
  4. tongue base and other parameters of the tongue sections
- *expressions.dat*: Expression definitions are stored in this file. An example entry in this file is given in Figure A.5. Parameters in this file are as follows:
    1. *name* of the expression,
    2. degree of the expression,
    3. muscle contraction values for the lowest degree,
    4. muscle contraction values for the highest degree and
    5. rotation angles of jaw for the lowest and the highest degrees.
  - *overlays.dat*: Names of overlays are written in that file.

Appearance in the file:

```

a
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.5 0.2 0.1 0.0 -8.0
0.0 -3.1 6.1
1.30 0.0 0.5 0.85 0.4
1.25 0.0 0.5 0.6 0.4
1.18 0.0 0.5 0.45 0.4
1.08 0.0 0.0 0.0 0.4

```

Figure A.4: Example letter definition.

Let us explain the meanings of numbers in Figure A.4.

*name* of the letter: a

Muscle contraction values;

For inactive muscles:  $\overbrace{0.0 \dots 0.0}^{30}$

For active muscles: Horizontal abstractions of *Orbicularis Oris* are contracted equally by 0.5 and 0.5 units, vertical abstraction first (upwards) is contracted by 0.2 units and the other vertical abstraction (downwards) is contracted by 0.1 units.

Jaw is rotated 8.0 degrees downwards.

The rest of the parameters are about tongue.

Tongue base: 0.000    -3.100    6.100

Remaining numbers are width, height, thickness, length and midline parameters for 4 sections.

Appearance in the file:

```
HAPPINESS
0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 -2.000
```

Figure A.5: Example expression definition.

Let us explain the meanings of numbers in Figure A.5.

*name* of the expression: HAPPINESS

Muscle contraction values for the lowest degree: Left and right *Zygomatic Major* muscles are contracted equally by 0.5 units for the lowest degree of expression.

Muscle contraction values for the highest degree: Same muscles are contracted to the values of 1.0 and 1.0, respectively.

There is no change in other muscles for any level of expression.

Jaw is not rotated for the lowest degree but it is rotated 2.0 degrees downwards for the highest degree.

# Bibliography

- [1] Basu, S, "A Three Dimensional Model of Human Lip Motion," M.Sc. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1997.
- [2] Boeschoten H. and Verhoeven L., eds., **Turkish Linguistics Today**, pp. 12-13, 1991.
- [3] Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S. and Stone, M., "Animated Conversation: Rule-Based Generation of Facial Expression Gesture and Spoken Intonation for Multiple Conversational Agents," *ACM Computer Graphics (In Proceedings of SIGGRAPH'94,)* pp. 413-420, July 1994.
- [4] Cohen M. M. and Massaro D. W., "Modeling Coarticulation in Synthetic Visual Speech," *Models and Techniques in Computer Animation*, pp. 139-156, 1994.
- [5] Duncan S., "On the Structure of Speaker-Auditor Interaction During Speaking Turns," **Language in Society**, Vol. 3, pp. 161-180, 1974.
- [6] Ekman P., "The Argument and Evidence About Universals in Facial Expressions of Emotion," **Handbook of Social Psychophysiology**, pp. 143-1674, Chichester, Wiley, 1989.
- [7] Gdkbay, U., "A Movable Jaw Model for the Human Face," *Computers & Graphics*, Vol. 21, No. 5, pp. 549-554, 1997.
- [8] Hearn, D. and Baker M. P., **Computer Graphics**, Englewood Cliffs, N.J., Prentice-Hall, 1986

- [9] Kalra, P., Mangili, A., Magnenat-Thalmann, N. and Thalmann, D., "SMILE: A Multilayered Facial Animation System," *In Proceedings of IFIP WG 5.10*, pp. 189-198, Tokyo, 1991.
- [10] Magnenat-Thalmann, N., Primeau, N. E. and Thalmann, D., "Abstract Muscle Action Procedures for Human Face Animation," *Visual Computer*, Vol. 3, No. 5, 290-297, 1988.
- [11] Neider J., Davis T. and Woo M., **OpenGL Programming Guide**, Addison-Wesley, 1994.
- [12] Parke F. I., "Computer Generated Animation of Faces," *In Proceedings of ACM National Conference*, Vol. 1, pp. 451-457, August 1972.
- [13] Parke F. I., "A Model for Human Faces that Allows Speech Synchronized Animation," *Computers & Graphics*, Vol. 1, No. 1, pp. 1-4, 1975.
- [14] Parke F. I., "Parameterized Models for Facial Animation," *IEEE Computer Graphics and Applications*, Vol. 2, No. 9, pp. 61-70, November 1982.
- [15] Parke F. I. and Waters, K., **Computer Facial Animation**, A. K. Peters, Wellesley, MA, 1997.
- [16] Pearce A., Wyvill B., Wyvill G. and Hill D., "Speech and Expression: A Computer Solution to Face Animation," *In Proceedings of Graphics Interface'86*, pp. 136-140, 1986.
- [17] Pelachaud C., Badler N.I. and Viaud M., "Final Report to NSF of the Standards for Facial Animation Workshop," University of Pennsylvania, Department of Computer and Information Science, October 1994.
- [18] Pelachaud C., van Overveld C. W. A. M. and Seah C., "Modeling and Animating the Human Tongue during Speech Production," *In Proceedings of Computer Animation'94*, Geneva, Switzerland, May 1994.
- [19] Platt S. M., "A Structural Model of the Human Face," Ph.D. Thesis, University of Pennsylvania, Department of Computer and Information Science, 1985.

- [20] Terzopoulos D. and Waters K., "Physically-based Facial Modeling, Analysis and Animation," *The Journal of Visualization and Computer Animation*, Vol. 1, pp. 73-80, 1990.
- [21] Truevision TGA, **File Format Specification, v2.0**, *Technical Manual v2.2*, Truevision, Inc, January 1991.
- [22] Uz B., Gdkbay U. and zgc B., "Realistic Speech Animation of Synthetic Faces," *In Proceedings of Computer Animation'98*, pp. 111-118, June 1998.
- [23] Waters K., "A Muscle Model for Animating Three-Dimensional Facial Expression," *ACM Computer Graphics (In Proceedings of SIGGRAPH'87)* Vol. 21, no. 4, pp. 17-24, July 1987.
- [24] Waters K. and Frisbie J., "A Coordinated Muscle Model for Speech Animation," *In Proceedings of Graphics Interface'95*, pp. 163-170, May 1995.
- [25] Waters K. and Levergood T. M., "DECFace: An Automatic Lip-Synchronization Algorithm for Synthetic Faces," Technical Report, CRL 93/4, *DEC Cambridge Research Laboratory*, Cambridge, MA, September 1993.
- [26] Watson S. H., Wright J. R., Scott K. C., Kagels D. S., Freda D. and Hussey K. J., "An Advanced Morphing Algorithm for Interpolating Phoneme Images to Simulate Speech," Technical Report, Jct Propulsion Laboratory, California Institute of Technology, 1996.