

**THE FACIAL ACTION CONTROL EDITOR,
FACE:
A PARAMETRIC FACIAL EXPRESSION EDITOR FOR COMPUTER
GENERATED ANIMATION**

BY

CLEA THERESA WAITE

S.B. MASSACHUSETTS INSTITUTE OF TECHNOLOGY
(1984)

*Submitted to the Media Arts and Sciences Section
In Partial Fulfillment of the Requirements
for the Degree of
Master of Science in Visual Studies*

at the

Massachusetts Institute of Technology

February, 1989

© MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 1989 ALL RIGHTS RESERVED

SIGNATURE OF AUTHOR _____
MEDIA ARTS AND SCIENCES SECTION
JANUARY 19, 1989

CERTIFIED BY _____
DAVID ZELTZER
ASSOCIATE PROFESSOR OF COMPUTER GRAPHICS
THESIS SUPERVISOR

ACCEPTED BY _____
STEPHEN A. BENTON
CHAIRMAN DEPARTMENTAL COMMITTEE ON GRADUATE STUDENTS

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

FEB 24 1989

LIBRARIES
Rotch

**THE FACIAL ACTION CONTROL EDITOR,
FACE:
A PARAMETRIC FACIAL EXPRESSION EDITOR FOR
COMPUTER GENERATED ANIMATION**

BY

CLEA THERESA WAITE

*Submitted to the Media Arts and Sciences Section
on January 19, 1989,
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Visual Studies*

ABSTRACT

The *Facial Action Control Editor, face*, is a system for emulating human facial expressions. It contains an editor which operates on a conformal muscular model of the face. This model uniquely combines a bicubic B-spline model of the skin with an underlying model of the facial muscles. The expression editor employs an intuitive user interface to simplify manipulating the muscular parameters using an empirical implementation of the FACS Action Units representation. It also provides facilities for the keyframe-based animation of facial movements.

Thesis Supervisor: Dr. David Zeltzer

Title: Associate Professor of Computer Graphics

To my mother, Angela, and Mario — with love;

*And to the hard core —
who made the place,
the time,
and the hours
endurable,
and even fun—
John, Paul, and Bill.
It was an era.*

CONTENTS

TITLE PAGE	1
ABSTRACT	2
INTRODUCTION	6
CHAPTER 1: OVERVIEW	8
1.1 The History of Parametric Facial Animation	10
1.2 The Model	13
1.3 Issues Addressed	15
1.4 Applications	16
1.5 Support	18
CHAPTER 2: THE ANATOMY OF THE HUMAN FACE	19
2.1 The Anatomical Structure of Muscles	19
2.2 The Muscles of the Human Face	21
2.3 Summary	25
CHAPTER 3: THE ACTION UNITS OF THE FACIAL ACTION CODING SYSTEM	27
3.1 Upper Face Action Units	29
3.2 Lower Face Action Units	31
3.3 Miscellaneous Actions	35
3.4 Implemented Action Units	35
CHAPTER 4: THE SKIN	37
4.1 The Waters Spring Model	38
4.2 Bicubic B-spline Patches	40
4.3 <i>spline-molder</i> : A Tool for Shaping B-Spline Surfaces	46
4.4 Control Point Constraints	48
CHAPTER 5: THE MUSCLES (UNDER THE SKIN)	61
5.1 Quantifying Facial Movement	62
5.1.1 Reversing FACS	62
5.1.2 Methodology	64
5.2 The Structures	68
5.3 Example: Filling the Structures	74
CHAPTER 6: MOTION	84
6.1 The Mass/Spring Model	85
6.2 Response of the Muscle	87

6.3	The Interpolation Curve	88
6.4	Simultaneous Muscle Action	91
CHAPTER 7: CONCLUSION		92
7.1	Intentions	92
7.2	Implementation Summary	93
7.3	Comparative Analysis	95
7.4	Results	74
7.5	Future Work	105
7.6	Shortcomings of the <i>face</i> Model	108
7.7	Concluding Remarks	110
• • •		
APPENDIX A: STRUCTURES AND DATA FILES		112
APPENDIX B: SPACE TRANSFORMATIONS		131
BIBLIOGRAPHY		133
ACKNOWLEDGEMENTS		136
FREEDOM		∞

INTRODUCTION

The face is the most expressive tool available to humankind, probably the most important tool in human communication. The face has a language all its own, more elemental than speech, writing, or gesture; more universal than any other form of communication:

...[perceptually] the face is more accurately judged, produces higher agreement, than do voice or speech...Younger children can be fooled easily by "salient paraphernalia" such as clothing [but] facial expression was not found to be a source of confusion in children of any age.[5]

The vast range and subtlety of expression evoked in the human face, its structure and manifestation, are the domain of this thesis.

• • •

The *Facial Action Control Editor*, *face*, was designed to facilitate the creation of subtle and realistic facial expressions for computer generated figures. *face* is a tool for sculpting a three-dimensional model of the face, for moving this model through the range of human facial expressions, and for animating these expressions to bring the model to life.

face is the first parametric model of the human face to combine an underlying structure of the facial muscles with a B-spline surface model of the skin. This unique combination creates a conformal model of the face, general to the spectrum of human facial types. The use of a muscle model creates realistic expressions which can be matched to human examples. The bicubic B-spline creates a "skin" which is smooth and propagates the changes to the face's surface and structure easily.

face emulates the human facial muscles using the *Facial Action Coding System, FACS* [6], a facial expression notation system, to organize and group the muscles into expression-creating units. Through a set of parameters, these expression units are programmed by the user to create the facial expressions. An intuitive and simple user-interface facilitates this editing of the muscle parameters. Expressions are created interactively and stored into scripts as keyframes for animated sequences.

The goal of this thesis has been to create a tool for making expressive faces using computer graphics; a tool based on a model of the muscles of the face. *face* has set out to create a facial model which compares to FACS in its generality, a control structure which is more intuitive and easy to use than its predecessors, and to demonstrate that the bicubic B-spline provides an equally good, if not better, model of the surface of the face for creating expressions than the polygonal models used up until its creation.

CHAPTER 1:

OVERVIEW

animate: - v.t. 1. To impart life to; make alive. 2. To move to action. 3. To produce the illusion of motion in (a projected film) a series of drawings.

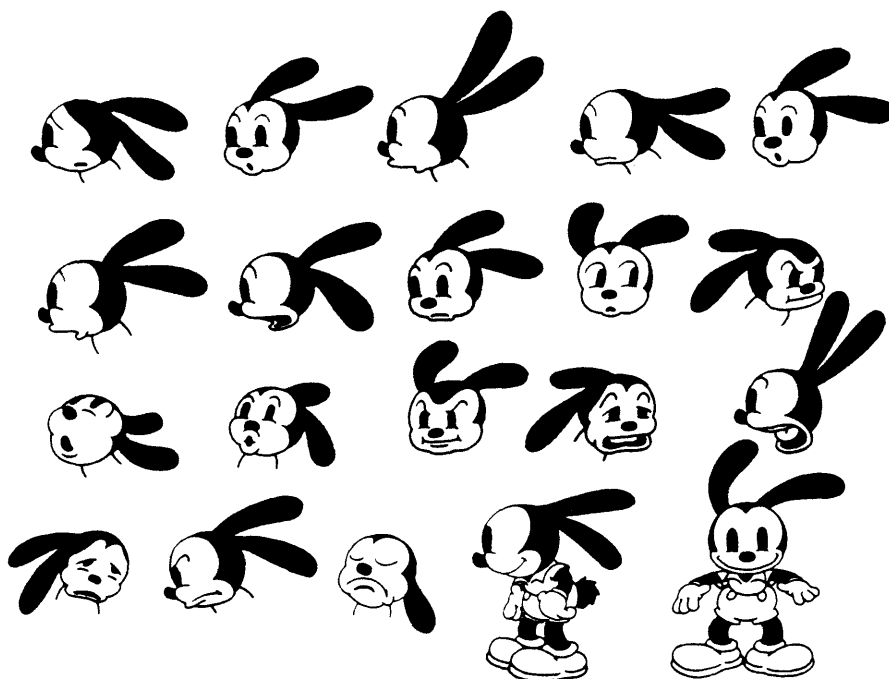
emulate: -v.t. To try to equal or surpass, especially by imitating.

-Funk and Wagnalls Standard Dictionary, 1980

Animation, the ability to analyze movement and draw it, can be traced back to prehistoric times and the cave paintings of hunters trapping bison [33]. Now, thousands of years later, we possess the technology to take this technique a step further, creating “motion pictures” of these drawings. Animators draw hundreds of drawings to create animated films, each drawing being viewed for a fraction of second to create the illusion of movement. In recent years, the computer has been enlisted to aid animators, taking over much of the tedium of the animation process.

One of the goals of computer generated animation is to take care of the “details” of animating. These details can range from drawing the in-between frames of a key-framed sequence to having an “intelligent” character who “knows” how to walk, the animator simply specifying its path and pace.

Because of the time involved in drawing realistic figures, animators who draw each frame by hand use simple figures to create their characters. Though simple, these figures are capable of expressing a great range of human emotions, as illustrated below by *Oswald the Lucky Rabbit*, a Disney precursor to *Mickey Mouse* [7].



(1.1) OSWALD THE LUCKY RABBIT [7]

But these characters, in general, cannot be called “realistic”.

With the aid of computers, realistic-looking scenes and characters can be created more quickly and easily than by hand, but most characters in computer generated animations lack any subtlety of expression and movement. Though often more photographically accurate, computer animated figures do not seem as “alive” as those in hand drawn

animations. The expressiveness of the animated character is crucial to this “life”. The face is key in expressing these human emotions. A realistic and expressive representation of the human face has been a topic of interest for over a decade in the computer graphics community. The *Facial Action Control Editor* presents one such model.

1.1 THE HISTORY OF PARAMETRIC FACIAL ANIMATION

The parametric control of data objects is an elegant application of the computer’s power to simplify complex representations. For example, with as few as two parameters, radius and location, a model of a sphere can be invoked into the graphical world. This same simplicity of representation can be extended beyond basic geometric shapes to more complex figures, such as the human face.

The job of the animator is made easier by the use of a parametric model. Rather than redrawing the entire face for each important change in expression, the animator can specify a few parameters, such as “move the left eyebrow up 50%” or “raise the corners of the mouth all the way, taking 30 frames to complete the motion”. The computer can then take care of the “tweening”, interpolating the parameters of the model to create motion.

• • •

The groundwork of *face* is based on the earlier works of Parke [24, 25], Platt [29, 30], Ekman [6], and Waters [37].

The Facial Action Coding System, FACS [6], was developed by Ekman and Friesen in 1977. FACS describes the set of all possible basic actions performable on a human face. The facial muscles are broken down into “Action Units”, *AU*’s, which describe the minimum performable actions. Each *AU* comprises a small set of closely related muscles, independent of any particular face. FACS is a notation-based environment that determines emotional states from visible facial distortion. Minimum units are specified in terms of what is both anatomically possible and reliably distinguishable. “FACS...

tends to simplify the process of controlling facial expression whilst simultaneously making the results more natural.” [14].

Parke was the first to use a parametric approach to define both face and facial action representation. He classified the face by defining parameters to describe the size attributes of the facial subsections. To perform animations his model interpolated between parameters encoded for different expressions, scaling and translating the subsections. Parke dealt well with using constraints to move his facial model through expressions, but his method probably would not work over a wide range of facial types because of the degree of hand tuning required to create each expression. This model would also require increasing the number of parameters to encode more subtle interactions of the face.

The basic premise of Parke’s model was to simulate the results of actions on a model of the skin. His model is concerned with the end resulting “look” of the face, not with the accuracy of the method by which the expression was achieved. Platt and Badler incorporated the actual motivators of the actions rather than just simulating the results directly. Platt is the first to model muscles pulling on the skin and to use FACS to encode the actions performable on the face. FACS was chosen from a group of possible action based notations because of its ease of computer adaptation, completeness, and extensibility to more general problems [30].

In Platt and Badler’s model, a muscle is a group of points with an arc stretching from each fiber point to a point on the underlying bone and another from the fiber point to one or more skin points. Information such as length and elasticity parameters are stored on the arc. The basic action is the application of a force to a portion of the tension net created by the skin. Forces are propagated outward along the net. Action Units consist of a list of muscles and relative magnitudes. But, because of the necessity to tie the muscles to a bone point, generality is again lost in this model as soon as it is applied to a new facial topology.

The most recent model, developed by Waters, ascribes parameters to muscles or groups of muscles that remain constant between one face and the next in the same way that FACS is universal across all different facial types.

The development of a muscle process that is controllable by a limited number of parameters and is non-specific to facial topology allows a richer vocabulary and a more general approach to the modeling of the primary facial expressions.[37].

Waters' model incorporates two types of muscles, linear/parallel muscles that pull and sphincter muscles that squeeze. Like Platt and Badler, Waters uses a simple mass/spring model of the muscles and skin. The limit of the muscle action is determined by a spring constant, K , representing the maximum displacement of the muscle. Unlike Platt and Badler, he models his muscles as vectors which are independent of the underlying bone structure, making his model general to a variety of different faces. The vector model allows a "free range in polygonal construction" [37]. This vector model is also central to the *face* implementation.

The muscle is modeled as a vector which affects a *zone of influence*, an area of the skin grid. This zone tapers off according to some specified falloff function along a radial path from the vector point, out along the grid. The positioning of the muscles is determined by identifying keynodes on the face and relating them to a 3D computer model of the face for the location of the muscle vector tail and head. Waters also uses a parameter structure based on FACS. He implanted ten muscles into the facial topology, representing those that are required for the AU's of FACS [37]. Waters' *zone of influence* occupies a finite area on the facial structure which can lead to unnaturally abrupt changes in the facial surface when a node is pulled far from its original location, especially if it is pulled outside the zone [27]. A better elasticity model of the skin is needed.

1.2 THE MODEL

The *Facial Action Control Editor* is a parametric face editor, designed for creating expressions on a computer graphics model of the human face. The goal of *face* was to create a tool with three main features:

- a model of the face which accurately portrays human facial expressions using a muscle emulation control structure,
- a conformal model which is general to a variety of facial types, and
- a system with an intuitive control structure interface.

THE MUSCLES

The structure of the *Facial Action Control Editor* is based on a muscular model of the human face. These muscles pull on control points of the face's surface, the skin, deforming it into facial expressions. The muscles are modeled as vectors having a single direction and a mobility factor. These muscle-vectors are then bundled together, like the fibers of an anatomical muscle, into groups which act upon the face in unison, the *Affectors*.

The muscle-vectors are the simplest unit in the control structure hierarchy. Bundles of vectors are grouped into *ActionUnits*. This grouping is based on the Facial Action Coding System. Much of the research done on parameterized models of the face for computer graphics animation [14, 19, 29, 30, 37] has used FACS as a basis for parameterization. *face* is an empirical implementation of FACS. Chapter 5 describes in detail the translation of FACS into a computational model of the facial muscle groups.

THE SKIN

The *face* muscle model acts upon a bicubic B-spline model of the skin. This is the first parametric model of the human face to combine a spline surface representation of the skin with a muscle-based control structure. The user interactively specifies parameter values to control the muscles which then move the spline control points in a constrained manner. The bicubic B-spline forms a good model of the facial skin, emulating the elasticity of the skin and the features well, and propagating changes to the surface smoothly. The spline is also a compact representation of the surface, requiring far less data storage than polygonal models.

The bicubic B-spline surface is molded into a model of the face called a *mask*. The *mask* emulates the key features of the face: the eye, nose, and mouth openings, the structure of the head, and the important surface wrinkles, the *furrows*. The *mask* is formed from a rectilinear grid of control points. The points of the grid are moved in space to mold the spline surface into the shape of a face. Each of the control points is assigned to one or more muscles and must be located in the area of the face to which it has been relegated. Within this constraint, the *mask* can be molded into any anatomically correct shape of the face. This conformability of the facial model extends to the vector muscle representation, making the entire *face* model general.

THE EDITOR INTERFACE

A group of virtual sliders, one for each Action Unit of FACS, is attached to the *mask*, making an interactive graphical interface for specifying the parameter values of the AU's. These sliders are called *zippers*. The muscles are controlled by applying a contraction force, represented as displacement on a zipper. The zipper control interface is very easy to use; knowledge of FACS is not necessary since each slider is located over the area of the face it affects, making its function obvious. Changes to the face are immediately visible, aiding the animator further.

The *mask* is put into an expression using the zippers, then that parametric configuration of the face is stored in a script. A series of such configurations are written consecutively into the script as keyframes. The settings of the different parameters are then interpolated over time between each keyframe and recorded, played back as a smoothly moving face.

MOTION

While in the editor, changes in length made to the muscles are linear. These changes represent end positions as a result of a contraction, not the contraction itself. The contraction of a muscle is a non-linear function, most simply approximated by a cosine curve [24, 37]. The function used in *face* to interpolate the scripted keyframes imposes such a non-linear contraction on the facial muscles as they move the *mask* between expressions.

1.3 ISSUES ADDRESSED

The *Facial Action Control Editor* addresses several of the issues raised in the previous work done by Waters, Platt, and Ekman. *face* borrows from the successes of their models, combining and extending many of their capabilities.

The muscle construct of *face* is based directly on the vector-muscle model described by Waters [37]. Waters' model was chosen because of its ability to conform to a variety of facial topologies. This muscle model is applied in *face* to a bicubic B-spline model of the skin rather than a polygonal representation, replacing the function of Waters' *zone of influence* with the spline basis functions. The B-spline skin model emulates the elastic properties of the skin in a natural fashion, propagating changes smoothly over the entire surface. The bicubic B-spline skin also models the furrows of the face successfully, and

suggests a method for emulating the skin of the face flowing over the underlying bone, issues presented by Waters as topics for future research [37].

Platt was the first to apply the FACS notation to a parametric computer model of the human face. FACS has since become the accepted notational system for specifying facial movements in parametrically controlled facial models [14, 19, 37]. The *Facial Action Control Editor* has also used this notation, building its structural abstractions around the FACS Action Unit representation. *face* translates the qualitative description of the facial movements in FACS into a computational model of the facial expression motivators, a task not described by Platt or Waters.

The FACS notation is a complex system to learn. Its construction is non-intuitive to the notator, requiring a great deal of training before it can be used easily. *face* has applied an intuitive user interface to the FACS notation, using graphic representation and interactive manipulation — the “zippers” described above — to simplify its use.

1.4 APPLICATIONS

An accurate and expressive computer model of the human face has many applications in a variety of areas. Two areas of great current interest are “synthetic actors” [20, 40] and speech synthesis [14]. Other areas include medical applications, law enforcement, telecommunications, and machine vision.

- In medicine, pre-operative tests on a realistic facial model help plan facial or dental surgery. Bone alignment [4] or remaining or regained mobility can be predicted. These systems also help train interns more inexpensively than cadavers.

- Law enforcement agencies can be aided by a 3D face modeler that is easy to use by crime victims, an aid to the identification of criminals.

- A parametrically controlled face model is applicable to communications, especially teleconferencing and visual telephones. These systems have a low image transmission bandwidth. If sufficient processing power is available at both ends of the transmission, a small set of data, *i.e.* just the facial control parameters, can be transmitted from one end to effect changes at the other end. A moving image of the face can be seen by both parties, despite the limitations of the transmission system.

- An accurate and precisely controllable model of the face could aid the study of perception. "Facial expression is relevant in the study of developmental psychology, personal perception, theories of emotion, and the neurophysiology of emotion." [5]. An especially interesting application of this is as an aid for the deaf — a lip reader.

The ability to create realistic facial expressions with a computer can also be translated into the ability of the computer to recognize expressions. In the fields of robotics and vision science, research is being done on computer facial recognition and head-motion detection by machines [21, 26]. The combined abilities of the computer to both recognize and create human facial expressions could affect the forms of human-computer interfaces in the future.

Basic motion understanding experiments have proven successful for various facial images...Computers possessing "eyes" will be able to smoothly interact with a person. Computers having a "face" and "body" will also be a great help in facilitating human-machine interaction in various situations.[21]

1.5 SUPPORT

The *Facial Action Control Editor, face*, was created in conjunction with the ongoing research of the Computer Graphics and Animation Group at the MIT Media Laboratory. The goal of this research is to “explore and implement computer animation systems which draw on new ideas about representing and organizing knowledge for computer animation.”[40] *face* has been designed to facilitate the creation of graphical faces with more subtle and realistic expressions than have been created by computer graphics in the past. *face* was implemented on the Hewlett-Packard Bobcat series computers, taking advantage of the SRX smooth shading hardware and the Starbase graphics library provided with these machines. Partial funding was provided by the National Broadcasting Company of Japan (NHK) and an equipment grant from the Hewlett-Packard Corporation.

CHAPTER 2:

THE ANATOMY OF THE HUMAN FACE

Whereas muscles may be neglected in the production of other human movements [by computers], they must be taken into account for facial animation.[34].

The underlying principle of *The Facial Action Control Editor* is the emulation of the muscles used by the human face in creating “expressions”. Though *face* uses a simplified model of muscles and only a subset of the muscles found in the face, a basic understanding of facial anatomy was necessary to design this simplified model. This chapter is a digression into the human anatomy, an explanation of the general structure of the striated muscles and a more detailed description of the principle expression muscles of the face.

2.1 THE ANATOMICAL STRUCTURE OF MUSCLES

Fundamentally, muscles are bundles of fibers working in unison. The grouping of these fibers is important in determining the relative strength and range of movement of the muscle.

Muscles with fewer, longer fibers have great range of movement and less strength, while short muscles with numerous fibers have great power, but less range of motion.

Muscles have points of origin and points of insertion. The *origin* of a muscle is the more fixed or central place of attachment of the muscle. The *insertion* is the moveable point of the muscle. The force of a muscle is directed towards the insertion, contracting it isotonicly. The origin remains static at its point of attachment.

In addition to length, the orientation of the muscle fibers also determines the functionality of a muscle. Below are the six major types of bundle orientations of the striated muscles:

- *Quadrilateral muscles*- The fibers of a quadrilateral muscle are parallel and run directly from the origin to the insertion.
- *Fusiform muscles*- The fibers of the fusiform muscle are not quite parallel but rather are slightly curved, tapering at either end. The action of this muscle is very similar to that of the quadrilateral muscle.
- *Triangular muscles*- The triangular muscle fibers arise at a broad origin and converge to a narrow or pointed insertion. This fiber arrangement is also found in pyramidal muscles.
- *Rhomboidal or penniform muscles*- The fibers in this case are situated obliquely, coming together on one side of a tendon which runs the entire length of the muscle.
- *Bipenniform muscles*- The bipenniform muscle contains oblique fibers which converge to both sides of a central tendon which runs the length of the muscle.

•*Sphincter muscles*- The fibers of a sphincter muscle are arranged in curved bundles in one or more planes which contract radially towards the center of the muscle.

These six groupings can be further generalized into two basic contraction orientations of the muscular fibers. The direction of pull relative to the point of attachment can be broken into groups based on unidirectional, linear motion, and multidirectional angled or spiral motion. The unidirectional motion group includes the quadrilateral, rhomboidal, bipenniform, and fusiform muscles, and the multidirectional group includes the triangular, and sphincter muscles. These are the two groupings used in *face*.

2.2 THE MUSCLES OF THE HUMAN FACE

The muscles used in facial expression are subcutaneous voluntary muscles, attaching to the bone or fascia of the head and inserting into the soft tissue of the skin. In medical literature, the head is divided into ten regions [10] and the muscles of the cranium and face are divided into groups according to the regions in which they are situated. Not all of these regions and muscles are involved in creating facial expressions, so below are described only those which do. Many of these muscles act alone to create the “Primal Set” [5] of expressions.

CRANIAL REGION

The occipito frontalis is a thin, quadrilateral muscle running from the forehead to the back of the skull. The frontal portion of this muscle raises the eyebrows and the skin over the root of the nose, at the same time drawing the skin of the skull forward, forming transverse wrinkles on the forehead. The posterior portion of the muscle draws the scalp backwards. The medial portion of this muscle is capable of moving independently from the lateral portions. The occipito frontalis is primary in the expression of “surprise” when the eyebrows are elevated.

When exaggerated, raising the eyebrows even higher and forming the transverse wrinkles, the expression of “fright” or “horror” is present on the face.

PALPEBRAL REGION

The orbicularis palpebrarum is a sphincter muscle which surrounds the circumference of the orbit and the eyelids, functioning to close the eyelid. It is attached to the frontal bone, the superior maxillary bone and the occipito frontalis. The *palpebral* portion acts involuntarily, closing the lids gently as in sleep or blinking. The *orbicular* portion is subject to will. When the entire muscle is in action, the skin of the forehead, temple, and cheeks draw inwards, firmly closing the eyes and forming “crow’s feet” wrinkles.

The corrugator supercilii, intimately blended with the orbicularis palpebrarum, is a small, narrow, pyramidal muscle at the inner extremity of the eyebrow, inserting opposite the middle of the orbital arch. It is attached to the frontal bone at the internal angular process and to the occipito frontalis. The corrugator supercilii draws the brows in and down. It is the “frowning” muscle and may be regarded as the principle agent in the expression of “suffering”.

The levator palpebræ is the opposing muscle to the corrugator supercilii, raising the upper eyelid and exposing the globe of the eye.

NASAL REGION

The pyramidalis nasi is another pyramidal muscle which draws down the inner angle of the eyebrows, producing transverse wrinkles over the bridge of the nose. It is attached to the occipito frontalis and disappears into the nasal bone.

The levator labii superioris alæque nasi is a thin, triangular muscle by the side of the nose which draws upwards the upper lip and wing of the nose, acting to dilate the nostrils. This muscle is the principle agent in the expression of “contempt” and “disdain”.

The dilatator naris posterior and anterior are small muscles which serve primarily to keep the nostrils from collapsing from atmospheric pressure while breathing, but can be violently active in situations such as difficult breathing and emotions like “anger”.

SUPERIOR MAXILLARY REGION

The levator labii superioris, a quadrilateral muscle, is the proper elevator of the upper lip, also moving the lip forward a small amount while elevating. It is attached to the superior maxillary bone, to the cartilage of the nose, and to the upper lip. This muscle assists in forming the naso-labial ridge, the area which runs from the side of the nose to the upper lip. The naso-labial ridge is part of the expression of “sadness”.

The zygomaticus major is a small muscle which attaches to the malar bone and descends obliquely into the angle of the mouth where it blends with the levator anguli oris, the orbicularis oris, and the depressor anguli oris. The zygomaticus major draws the angle of the mouth backwards and upwards as in laughing.

The zygomaticus minor inserts into the outer part of the upper lip rather than the angle of the mouth, and is continuous with the orbicularis oris while also attaching at the malar bone. This muscle draws the upper lip backward, upward, and outward, working with the levator labii superioris to express “sadness”.

The levator anguli oris, attaching to the superior and inferior maxillary bones and to the corners of the mouth, raises the angle of the mouth and assists the levator labii superioris in producing the naso-labial ridge.

INFERIOR MAXILLARY REGION

The levator labii inferioris is a conical fasciculus which raises the lower lip and makes it protrude forward, at the same time wrinkling the integument of the chin. This wrinkling of the chin creates an expression of “doubt” or “disdain”.

The depressor labii inferioris is a small quadrilateral muscle which draws the lower lip directly downward and slightly outward, making an “ironic” expression on the face.

The depressor anguli oris is a triangular muscle rising with a broad base from the external oblique line of the lower jaw and inserting into the lower lip. It’s function is to enlarge the mouth and depress the angle of the mouth by drawing it directly backwards.

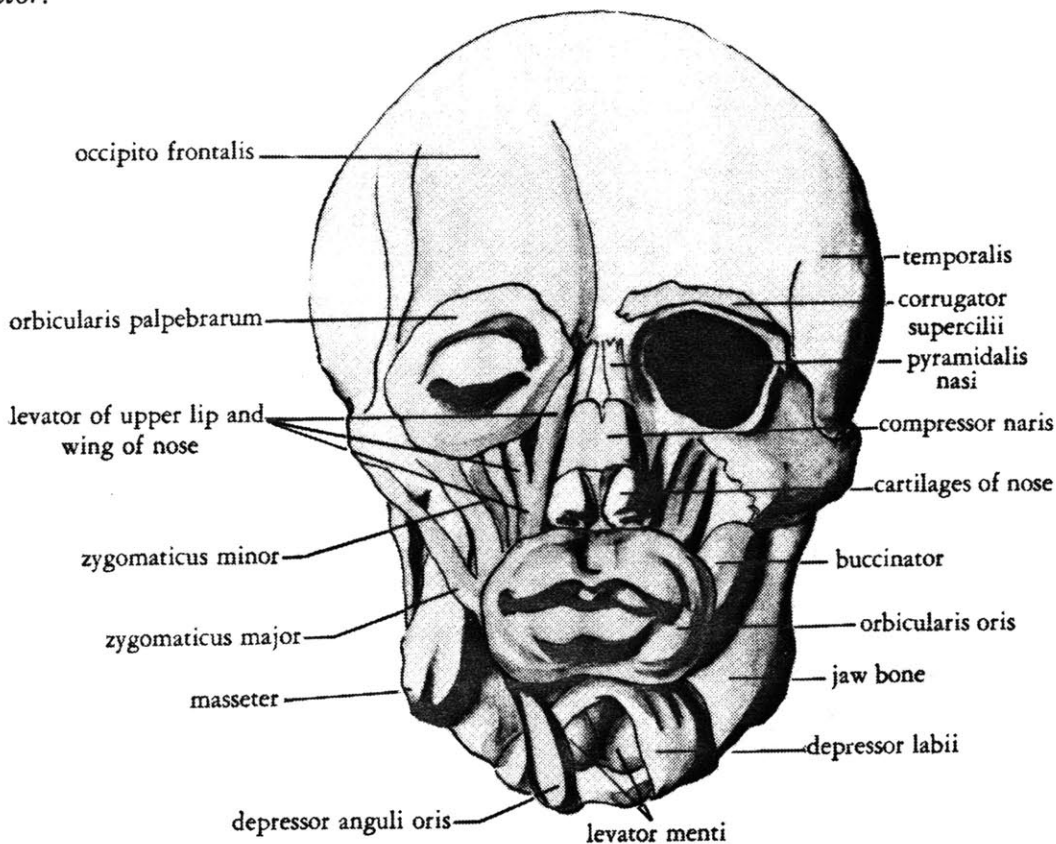
INTERMAXILLARY REGION

The orbicularis oris consists of numerous layers or “strata” [10] of muscular fibers. Many of these fibers are derived from other muscles, mingling with the orbicularis oris, including: the levator and depressor anguli oris, the buccinator, the depressor labii inferioris, the levator labii superioris alæque nasi, the levator labii superioris, and the zygomaticus muscles. These intermingle with the transverse fibers of the orbicularis oris and are primarily oblique in their direction. Those fibers which are proper to the lips themselves are also obliquely oriented. The levator and depressor anguli oris cross at the corners of the mouth, running along the lips and inserting at the median line. The depressor arises in the upper jaw, inserting into the lower lip and the levator arises in the lower jaw to insert into the upper lip. The fibers of the orbicularis oris are connected directly with the maxillary bones and the septum of the nose. The deeper stratum of this muscle is formed largely by fibers which derive from the buccinator.

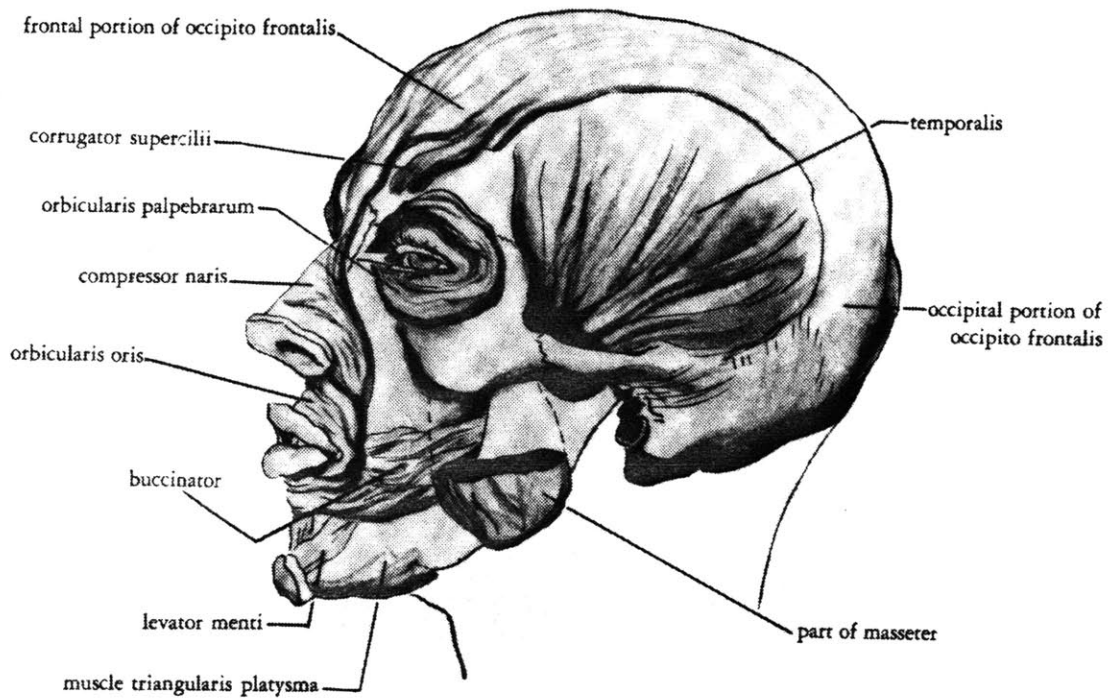
The buccinator is a broad, thin, quadrilateral muscle which is found in the interval between the jaws on the side of the face. It arises near the molars, the fibers converging towards the corners of the mouth. The central fibers intersect with each other, those from below joining the upper fibers of the orbicularis oris and those from above joining with the lower fibers. The highest and lowest fibers of the buccinator continue uninterrupted, inserting into the corresponding segments of the lips.

2.3 SUMMARY

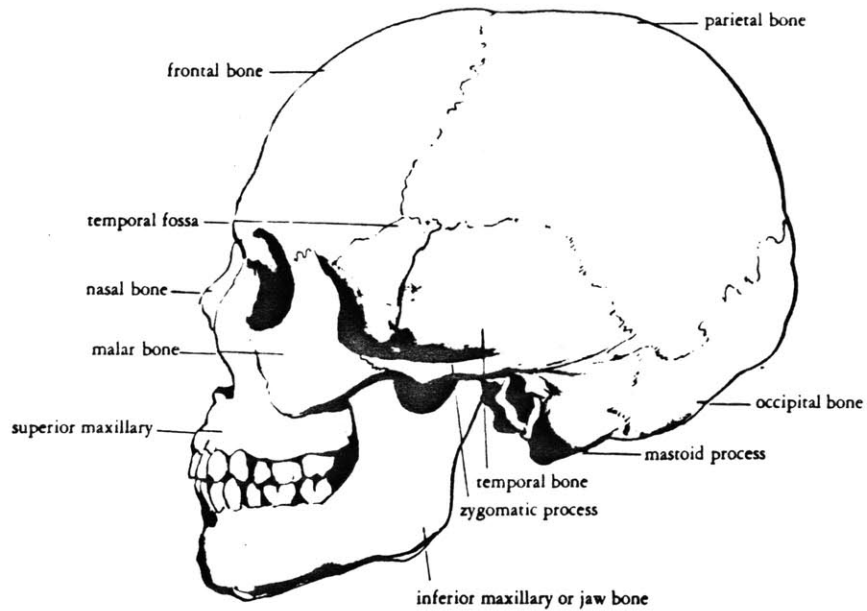
Because of the importance of the different types of muscles to developing a parametric model of FACS, a general overview of the voluntary muscles of the face has been presented. In addition, the muscles specific to the Action Units of the Facial Action Coding System, their essential features, and their relationships to one another, have been summarized. A simplified model of these muscles has been implemented as part of the *ActionUnit* structure of *face* (see Chapter Five). The orientation of the muscular fibers is especially relevant to this model. Each of these muscles has been explained in detail, including the location of its origin and insertion on the face, its fiber structure, the muscles to which it connects, and in appropriate cases, the expression for which it is most responsible. This is the information central to the emulation model, forming the anatomical basis for the *Facial Action Control Editor*.



(2.1) FACE MUSCLES, FRONT [18]



(2.2) FACE MUSCLES, PROFILE [18]



(2.3) THE BONES OF THE HEAD AND FACE [18]

CHAPTER 3:

THE ACTION UNITS OF THE FACIAL ACTION CODING SYSTEM

Once an appropriate parameterized model is developed, a wide variety of animation can be produced with relatively little effort.[25].

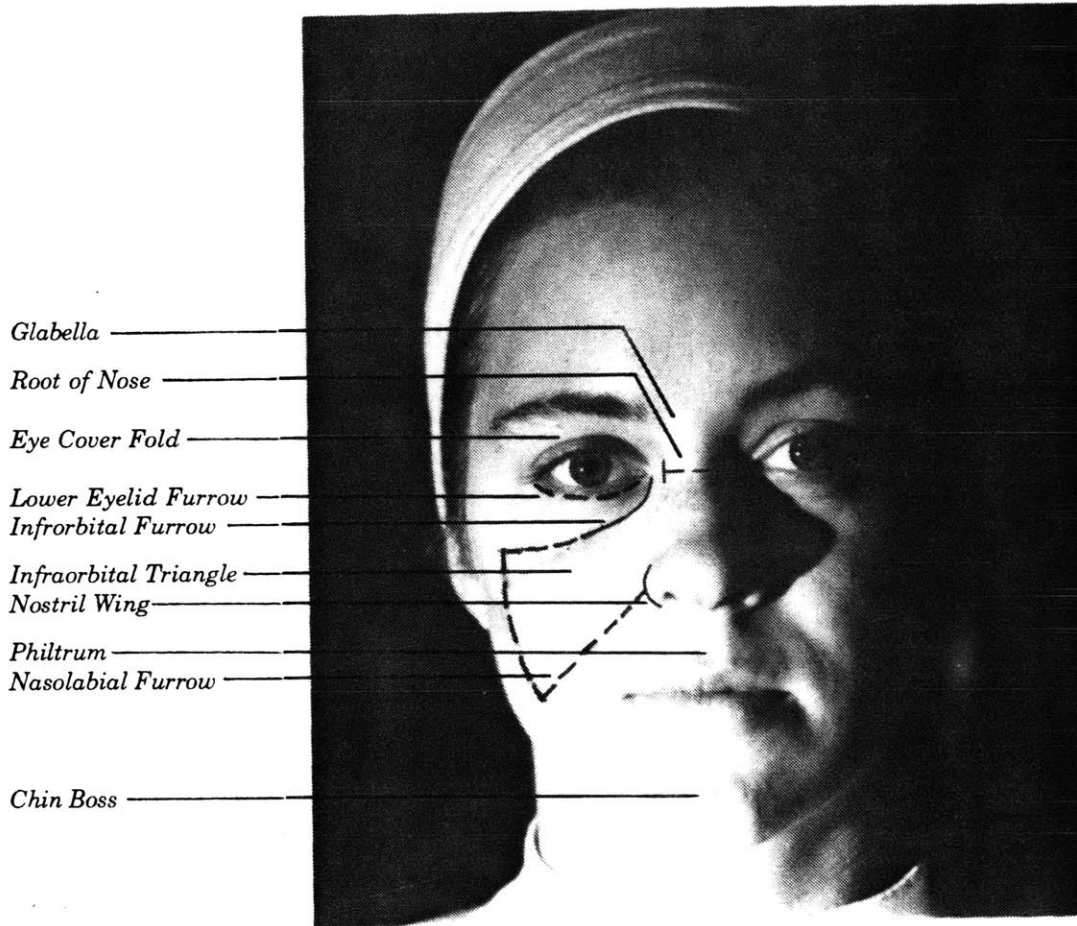
Next we examine the system developed by Ekman and Friesen, the *Facial Action Coding System*, as a simplified, quantified version of the muscles described in Chapter Two. FACS breaks the face into groups called Action Units, AU's, which are one or more muscles whose motions are used in creating facial expression: "AU's are responsible for momentary changes in facial appearance." [6]. FACS was developed as a means of recording facial expressions without imposing cultural or personal interpretation on the expressions. Using FACS, expressions are recorded as combinations of Action Units and levels of activation. Trained viewers look at a face and fill out a score sheet, rating the expression on the face. These viewers are trained from the *Facial Action Coding System Manual*, a document which was central to the development of *face*.

In the FACS Manual, the AU's are presented to be learned in groups which are broken down according to the locations of the Action Units and/or the types of actions involved. For example, the first group presented in the FACS manual is the *Upper Face*. These are the Action Units which affect the eyebrows, forehead, and eyelids. This grouping of Ekman's can be compared with the anatomical groupings in section 2.2. Ekman's *Upper Face* is equivalent to the Cranial and Palpebral regions and part of the Nasal region. The *Lower Face*, corresponding to the Superior and Inferior Maxillary regions, and the Intermaxillary region, is divided into five groups by actions: *vertical actions* (up/down), *horizontal actions*, *oblique actions* (involving upward, diagonal motions), *orbital actions*, and *miscellaneous actions*. The three other major categories in FACS are *Miscellaneous Actions*, *Head Position*, and *Eye Position*.

A summary of the Action Units relevant to this thesis has been provided in the remaining portion of the chapter, a condensed description of those Action Units which are possible with the present implementation of *face*. Other than the figures from the manual, all the available information necessary for designing a computational model of FACS has been included. Wherever possible, the Action Units have been correlated to their corresponding muscle or muscles. This correlation is helpful in the design of motivators which emulate the physical aspects of the expressive actions.

Due to the restrictions of the spline software used to create the facial "skin", eyes, eyelids, lips, and teeth are not included in the model. The Action Units which describe these areas of the face will therefore not be part of the following description. Cheek puffing and sucking and motions of the tongue have likewise been omitted due to constraints of the model. These are especially difficult areas whose emulation are still subjects of research. Action Units involving movement of the head as a whole have been left to the implementation of the articulated figure to which the face will eventually be added. Descriptions of wrinkles are generally omitted since the capability for these is also not included in the current model. Furrows and dimples have been included.

The FACS Manual terminology used to refer to certain areas of the face will also be used in the following descriptions of the Action Units. These areas, the *glabella*, the *root of nose*, the *eye cover fold*, the *lower eyelid furrow*, the *infraorbital furrow*, the *infraorbital triangle*, the *nasolabial furrow*, the *nostril wing*, the *philtrum*, and the *chin boss*, are illustrated in the diagram below.



(3.1) AREAS OF THE FACE

3.1 UPPER FACE ACTION UNITS

•*AU 1: Inner Brow Raiser*- The occipito frontalis muscle raises the eyebrows and the skin of the forehead. AU 1 is activated by the medial portion of the

occipito frontalis, pulling the inner portions of the eyebrows and the forehead upwards. The outer corner of the brows may pull inwards towards the center line of the face in the case of strong action.

•*AU 2: Outer Brow Raiser*- AU 2 is activated by the lateral portion of the occipito frontalis, pulling the brow and the skin of the lateral part of the forehead upwards. AU 2 also causes the outer portion of the eye cover fold to be stretched up towards the hairline.

•*AU 4: Brow Lower*- There are three muscle strands which underlie Action Unit 4, forming the pyramidal muscle of the corrugator supercilii. In the typical case, these three strands work together, lowering the eyebrows. The inner eyebrow alone, the inner and central eyebrow, or the entire eyebrow may be lowered by AU 4. The eye cover fold is pushed down in this action, possibly narrowing the eye opening. The eyebrows are pulled closer together and one or more horizontal wrinkles can form at the root of the nose.

•*AU 6: Cheek Raiser and Lid Compressor*- AU 6 is activated by the orbicularis palebrarum, the sphincter muscle surrounding the eye. This Action Unit involves the outer circumference of this muscle, which extends into the eyebrow and below the lower eye furrow. The skin is drawn in towards the eye from the temple and cheeks, raising the infraorbital triangle and lifting the cheeks. The skin surrounding the eye is pulled in radially, narrowing the eye opening. Other effects from AU 6 may include: lowering of the lateral portion of the eyebrows slightly, deepening of the nasolabial furrow, raising the outer portions of the upper lip slightly, and deepening of the infraorbital furrow, causing a line to run across the top of the infraorbital triangle.

•*AU 7: Lid Tightener*- AU 7 is also activated by the orbicularis palpebrarum, but is associated with the inner portion of this muscle, which circles the eye orbit. When contracted, AU 7 tightens the eyelids, pulling them towards the inner corner of the eye and narrowing the eye opening. The lower lid is usually the more affected by this motion, being raised up over the eyeball. The lower eyelid furrow may become visible or be deepened. When AU 7 is at maximum contraction, it is considered to be AU 44.

•*AU 44: Squint*- AU 44 is the narrowed and tightened eye opening of a squint as caused by AU 7. AU 5, the *Upper Lid Raiser*, may also be involved in the Squint as the antagonistic action to the eye narrowing motion. AU 44 is considered to be a separate Action Unit by FACS because the appearance is different from that of AU 7. The eye opening is very narrow and the lids are tensed with the skin below the lower eyelid pulled in towards the root of the nose. For the sake of implementation, AU 44 is simply the extreme case of AU 7.

3.2 LOWER FACE ACTION UNITS

UP/DOWN ACTIONS

•*AU 9: Nose Wrinkler*- The levator labii superioris alæque nasi, originating at the root of the nose and inserting into the upper lip adjacent to the nostril wing, is the muscle underlying AU 9. This Action Unit pulls the skin below the nostril wing upwards and usually pulls the brow downwards at the same time, as in AU 4. The infraorbital triangle is pulled upwards, deepening the infraorbital furrow and narrowing the eyes. Creases may also appear across

the root of the nose. The center of the upper lip is pulled upwards and in the case of a strong action, the lips may part. The nostril wings may be widened and raised and the nasolabial furrow deepened when the action is strong.

•*AU 10: Upper Lip Raiser*- AU 10 is one of the actions caused by the zygomaticus minor muscle which originates in the center of the infraorbital triangle and attaches at the nasolabial furrow. The skin above the upper lip and the upper lip are pulled upwards towards the cheek by this muscle. The center of the upper lip is drawn up vertically, rising higher than the outer portions of the lip, causing an angular bend to occur in the lip which forms a trapezoidal shape. The infraorbital triangle is pushed up and the infraorbital furrow is deepened. The upper portion of the nasolabial furrow is also deepened and formed into an angled, trapezoidal shape by this action. The nostril wings are raised and widened and in the case of a strong action, the lips will part.

•*AU 15: Lip Corner Depressor*- The depressor anguli oris originates from the side of the chin and inserts near the corners of the lips. This is the underlying muscle to AU 15, pulling the corners of the lips down at an angle and stretching them horizontally to a small degree. The skin below the lip corners may pouch, bag, or wrinkle, being most apparent when the action is strong. The chin boss may be flattened or bulge and a depression may appear centered below the lower lip. The nasolabial furrow is deepened and the lower portion of it is pulled downwards, appearing lengthened.

•*AU 16: Lower Lip Depressor*- AU 16 is also activated by the depressor anguli oris, which pulls the lower lip downwards towards the chin. This action stretches the lower lip, pulling it laterally and may cause it to protrude or

flatten. The skin of the chin boss is also stretched laterally and downward, possibly causing a wrinkle to appear below the lower lip.

•*AU 17: Chin Raiser*- The integument of the chin and the lower lip are pushed upwards by AU 17. A depression may appear below the center of the lower lip, and if the action is strong, the lower lip may protrude. AU 17 causes the mouth to be shaped into a downward arc as in the upper portion of a semi-circle.

•*AU 26: Jaw Drop*- AU 26 involves the lips, the lower jaw, and to a certain extent, the teeth. The latter are not present in the *face* model. The mandible is lowered in AU 26, parting the lips and showing or implying space visible between the teeth. The lips are not stretched by this motion and may even remain closed due to “lip friction” [6].

•*AU 27: Mouth Stretch*- AU 27 pulls the mouth wide open because the mandible is pulled down as opposed to the relaxed lowering of AU 26. The jaw is not dropped but pulled, stretching the lips vertically, flattening and stretching the cheeks, and changing the shape of the chin boss.

HORIZONTAL ACTIONS

•*AU 14: Dimpler*- The muscles involved in AU 14, which originate in the cheek bones and attach in the medial portions of the lips, are the buccinator and the orbicularis oris. One strand of the buccinator goes to the upper lip and the other goes to the lower lip. The action caused by this muscle pulls the skin beyond the lip corners inwards towards the lips, tightening and narrowing the corners of the mouth and pulling them in as well. A wrinkle or bulge will form at the lip corner and a “dimple-like wrinkle” [6] may form beyond the lip corners. The

angles of the lip corners during this action are usually downwards or straight, but may also be upwards in some cases. The lips are slightly stretched and flattened by this motion. The chin boss and the skin below the corners of the lips are pulled up towards the lip corners, also flattening and stretching the chin boss skin. The nasolabial furrow may be deepened during this action.

•*AU 20: Lip Stretcher*- AU 20 pulls the lips back laterally towards the ears in a horizontal motion. The lips become flattened and stretched, and the skin beyond the lip corners is also pulled flat. The lower portion of the nasolabial furrow is also affected, the skin in that area being pulled laterally. The skin over the chin is also pulled. The nostrils will be stretched by this motion.

OBLIQUE ACTIONS

•*AU 11: Nasolabial Furrow Deepener*- The upper lip is pulled upward and outward slightly by the zygomaticus minor, the pull point of AU 11 originating midway between the philtrum and the corners of the mouth. The skin below the upper part of the nasolabial furrow is pulled obliquely upwards, and the upper middle portion of this furrow is deepened. The infraorbital triangle may be puffed and the infraorbital furrow may deepen in a strong action.

•*AU 12: Lip Corner Puller*- AU 12 pulls the corners of the mouth back and up towards the cheekbones in an oblique motion, being activated by the zygomaticus major. The nasolabial furrow is deepened during this motion and the adjacent skin is raised in an upward and lateral direction. The infraorbital triangle is raised up, the nostrils are raised and widened and the skin on the chin boss is flattened and stretched during a strong action. Many of

the effects seen with AU 12 on the upper face are also produced by AU 6, the Cheek Raiser.

•*AU 13: Sharp Lip Puller*- AU 13 is similar to AU 12 in that it also pulls upwards on the corners of the mouth, but the angle of pull of AU 13 is much sharper, causing the cheeks and infraorbital triangle to “puff out”. The motion of AU 13 is up rather than oblique. The lip corners become tightened, narrowed, and sharply raised, but the lips themselves do not move. The upper and middle portions of the nasolabial furrow may deepen, and in the case of a strong action, the infraorbital furrow will deepen. A strong AU 13 will also cause many of the same changes as AU 6.

3.3 MISCELLANEOUS ACTIONS

•*AU 38: Nostril Dilator*- AU 38 flares out the nostril wings, changing the shape of the nostril opening and may cause a bulge in the nostril wing by the actions of the dilator naris posterior and anterior muscles.

•*AU 39: Nostril Compressor*- The nostril wings are flattened and may be pulled downwards due to AU 39. The nostril itself will become less wide. This AU only occurs as a motion, it cannot be maintained as a static expression.

3.4 IMPLEMENTED ACTION UNITS

Those Action Units which are currently activated in *face* include: AU1- Inner Brow Raiser, AU2- Outer Brow Raiser, AU4- Brow Lowerer, and AU12- Lip Corner Puller.

Examples of these Action Units and a detailed description of their implementations are given in Appendix A.

CHAPTER 4:

THE SKIN

There are two techniques for expressing shape currently used in computer animation. One is the technique of polyhedron approximations. The other is the technique of the composition of primitives, i.e., basic figures such as spheres, cubes, and revolving shapes. It is difficult for either of these techniques to achieve...conditions essential for a model expressing human skin.[17]

The development of the complete system which comprises the *Facial Action Control Editor* took place in three phases. The first phase involved developing a model of the face surface, the skin, and in the process building tools with which to sculpt this skin into a recognizable human form. Phase two was the development of a muscle model with which to manipulate the skin in order to create expressions. This phase also involved creating an intuitive user-interface to these manipulators. The final phase was to create realistic looking motion, facilitated by a script building tool, via an interpolation function. This chapter examines the first of these phases, designing the skin.

The skin model for the *Facial Action Control Editor* needs to both look and function like the human face in order to create an acceptable emulation of facial expression. The model used by *face* uses a bicubic B-spline to create this surface of the skin. The B-spline surface is then carved and shaped using a tool developed for this purpose, *spline-molder*. Careful restrictions had to be made on the placement of the control points when shaping the skin surface into a human facial form to facilitate the workings of the muscle model. The final face-shaped surface is called a *mask* and is the facial model used by the expression editor.

The use of the bicubic B-spline to create the surface of the skin was developed as an extension of the method described by Waters [37] to simulate facial skin, what he calls *zones of influence*. Waters' *zones of influence* operate on a polygonal model of the skin's surface. The model developed for *face* replaces this polygonal skin and the *zones of influence* in one step with the bicubic B-spline, creating a simpler, yet equally realistic and general model of the face. This model, its constraints, and the tools created to shape it, are described here in detail.

4.1 THE WATERS SPRING MODEL

The Waters' facial model is a polygonal mesh of simulated springs, interconnecting to create the skin of the face. This skin is then acted upon by muscle vectors to cause motion of the face's surface. The muscles attach to node points on this skin, pulling at the node with a force, \mathbf{F}^* . If the force \mathbf{F} is applied to a point \mathbf{p} , the new position of the point \mathbf{p} is given as

$$[4.1] \quad \mathbf{p}' = \mathbf{F}/K$$

*The bold typeface is used to denote a three-dimensional vector.

where K is the sum of all the spring constants at that point. This displacement, \mathbf{p}' , is propagated by the springs from the node through the skin mesh.

The skin mesh is defined such that each node has a finite degree of mobility. The factors Waters uses to define the nodal mobility are: the tensile strength of the muscle and skin, the proximity of the given node to the node to which the muscle is attached, the depth of the tissue at the node and its proximity to the bone, the elastic bounds of the relaxed tissue, defined as its spring constant (equation 4.1), and the interaction of other muscles with that node.

At any given point along the skin mesh, the effective proportion of the force on the point will depend on the line of the contraction relative to the skin at that point. Waters models muscle "fibers" which run from the attachment node to each skin point. The more oblique the fibers of the muscle are to the attachment node, the more dependent the force on the skin point will be on the angle θ between the fiber and the skin.

$$[4.2] \quad \mathbf{F} = l_f (\cos \theta)$$

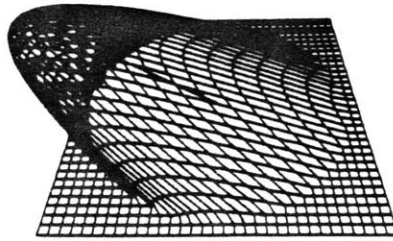
where l_f is the length of the muscle fiber.

ZONE OF INFLUENCE

Muscles, in Waters' model, have what he calls a *zone of influence* over a certain area of skin. This zone delimits the area of the polygonal surface which will be affected by pulling on a node and the way in which the points of the surface within the zone are affected. Any function can be used to describe this zone, but the simplest example is a circular zone in which the affect of the force on a given point depends not only on equation 4.2 above, but also on the radial distance of the skin point from the affected node point. The forces on points fall off as a function of the radial distance of the points from the affected node.

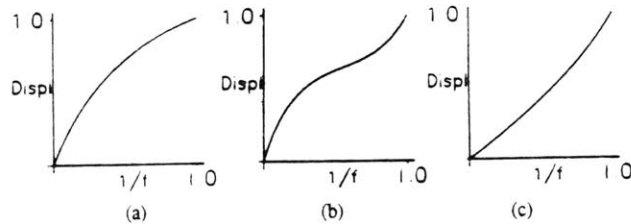
$$[4.3] \quad \mathbf{p}' \propto f(K.A.R.p)$$

where K is again the spring constant of the muscle, A is the angular displacement factor, and R is the radial displacement factor [37]. Points outside the circle of the zone are not affected by this node.



(4.1) A SURFACE DISPLACED BY A MUSCLE VECTOR USING A CIRCULAR COSINE FALLOFF [37]

This *zone of influence* can also be described using a Gaussian, or any other appropriate function.



(4.2) THREE FUNCTIONS USED TO DESCRIBE ZONE OF INFLUENCE FALLOFF [37]

4.2 BICUBIC B-SPLINE PATCHES

The model used in *face* uses a bicubic B-spline patch to represent skin rather than the polygonal spring-mesh used by Waters. "...the face is not a rigid structure but is a complex flexible surface." [24]. The B-spline has the advantage of the basis or blending functions, which are used to describe Waters' *zone of influence*. These basis functions not only provide an "automatic" zone of influence, but also serve to emulate the springy nature of the skin, propagating the movement of one point smoothly over an area of the surface. "...the

intrinsic property of mutual tension at the control points gives an elasticity to the muscles and skin of a figure in motion.” [22]

The bicubic B-spline is a parametric representation of a three-dimensional curved surface constructed from a two-dimensional array of cubic curves. Cubic curves, in comparison to quadratic or quartic curves, are useful because they provide continuity of position, slope, and curvature at the point where curve segments meet and because they are variation diminishing, a property which the higher order curves lack.

Spline curves are often described in terms of their continuity. $C(i)$ continuity means that the function and its first i derivatives are continuous [13]. A bicubic spline surface has $C(2)$ continuity; the 0th, 1st, and 2nd derivatives of the curve function are continuous, *i.e.* position, slope, and curvature are all continuous at the joining points between segments. This is an important feature in creating the skin of the *mask*. The B-spline is said to be “smoother” [1] than the other forms of cubic curves, the Hermite and Bezier functions. In addition, the B-spline is well suited for interactive manipulation because the geometry vector contains only points and not tangent vectors, which simplifies making changes to the curve.

The x component of a one-dimensional cubic curve, a function of the interpolation parameter u , is given by:

$$[4.4] \quad x(u) = UM_S G_{S_x}$$

$$[4.5] \quad U = [u^3 \quad u^2 \quad u \quad 1]$$

$$[4.6] \quad M_S = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

$$[4.7] \quad \mathbf{G}_s = \begin{bmatrix} \mathbf{P}_{i-1} \\ \mathbf{P}_i \\ \mathbf{P}_{i+1} \\ \mathbf{P}_{i+2} \end{bmatrix} \quad 2 \leq i \leq n-2$$

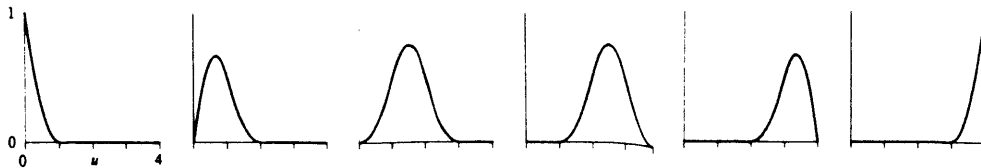
$$[4.8] \quad \mathbf{P}_i = [x_i, y_i, z_i]$$

where M_s is the basis function, \mathbf{U} is the interpolation vector, and matrix \mathbf{G} is the *geometry vector*. For each point computed along the spline curve $[x, y, z](u)$, the positions of four control points, \mathbf{P}_i , are taken into account by \mathbf{G} : \mathbf{P}_{i-1} , \mathbf{P}_i , \mathbf{P}_{i+1} , and \mathbf{P}_{i+2} , in the form of a weighted sum. The B-spline curve is a series of approximations to these control points, $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_n$, using a different geometry matrix between each pair of consecutive points. In this way the curve passes smoothly along the path laid out by the control points. This is the *blending function* affect which *face* uses to produce the equivalent of Waters' skin-springs and the *zone of influence*. It can also be seen from \mathbf{G} that each join point between segments of the parametric curve is a weighted sum of its three adjacent control points, creating the $C(2)$ continuity between curves. Figure 4.3 illustrates the weighted sum distribution curves.

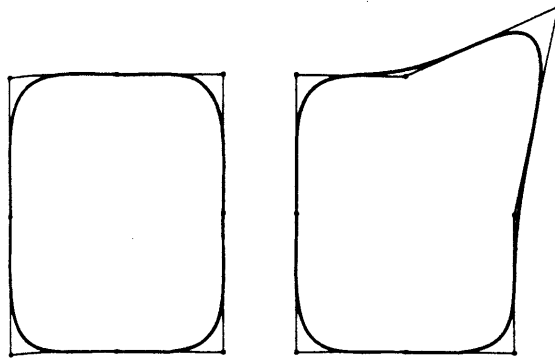
By extension, a two-dimensional parametric surface is represented as a function of u and v . The parameters u and v are generally limited to vary between 0 and 1

$$0 \leq u \leq 1.0.$$

The reader is referred to [1] for a full treatment on splines.



(4.3) B-SPLINE BLENDING (BASIS) FUNCTIONS FOR A QUARTIC SPLINE [23]



(4.4) THE B-SPLINE SMOOTHLY INTERPOLATES CHANGES TO THE CONTROL POINT POSITIONS [23]

TRIMMED CURVES

The bicubic B-spline surface described above is the foundation for creating the skin of the face model. This surface or *patch* is the stretchy, springy “stuff” which can be shaped into something that begins to look like a face. It is our clay. But a simple patch is not enough. The next issue to be addressed is the creation of orifices, or “holes”, in this skin patch. These holes will become the eyes, mouth, and nostrils of the face model, the *mask*.

Holes are created in parametric surfaces using *trimming curves* [13]. Trimming curves are closed (ends touching), one-dimensional splines which define boundaries in the parameter space of the patch. The surface is drawn inside the parametric boundaries of the trimming curve, or between the boundaries of two or more trimming curves. For example, if the patch below (fig. 4.5a) is a function of u and v ,

$$0 \leq (u,v) \leq 1.0,$$

and the trimming curve exists inside the patch, in parameter space (the two-dimensional space of the patch) from $0.2 < u < 0.8$, then the patch will be cut off outside the parametric bounds of the trimming curve (fig. 4.5b), like cutting out a cookie from a piece of dough.

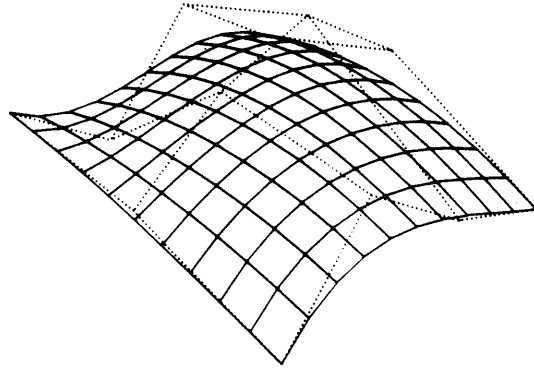
To keep the dough and throw away the cookie, a second one-dimensional spline curve needs to be defined as the outer boundary, making the original curve into the inner boundary (fig. 4.5c). The outer boundary can be as simple as a one-dimensional linear curve ($C(0)$) which traces the edges of the patch.

FURROWS

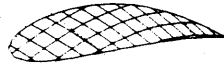
After cutting the five holes in the skin-patch of the *mask*, the next problem is to make this surface behave more like the facial skin when a force is applied to it. Among the important features of the skin of the face are the landmark furrows, creases in certain areas of the face. These furrows are the infraorbital furrow, the naso-labial furrow, and the creases along the nostril wings.

A crease is represented mathematically as a discontinuity in the curvature of the surface of the skin. In a bicubic surface, this discontinuity, a one degree factor, would decrease the continuity at the furrow point to first derivative continuity, continuity of slope and position, losing the second derivative, $C(2)$, continuity of curvature. This lowering to $C(1)$ continuity is accomplished by doubling up and tying together control points where creases are required. Several of these doubled points are lined up in a row to form a furrow.

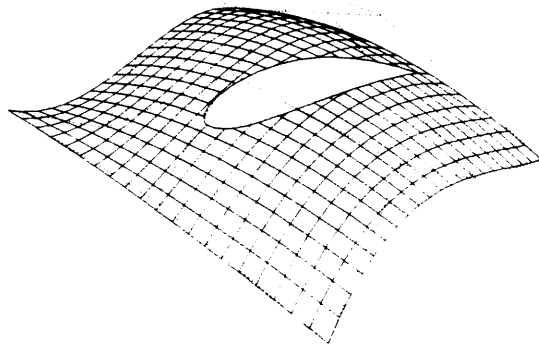
Every control point occupying the same position as another control point decreases the continuity of the curve at that point by one degree. This is because the geometry vector is effectively reduced from four to three points by this move. Creases and cusps can be formed in this way despite the blending functions. Below in figure 4.6, the skin patch, with orifices and furrows in place, is shown. This patch, the *rectilinear face patch*, is composed of one, 16x12 array bicubic B-spline patch. This is the "clay" used to create all *masks*.



(A)

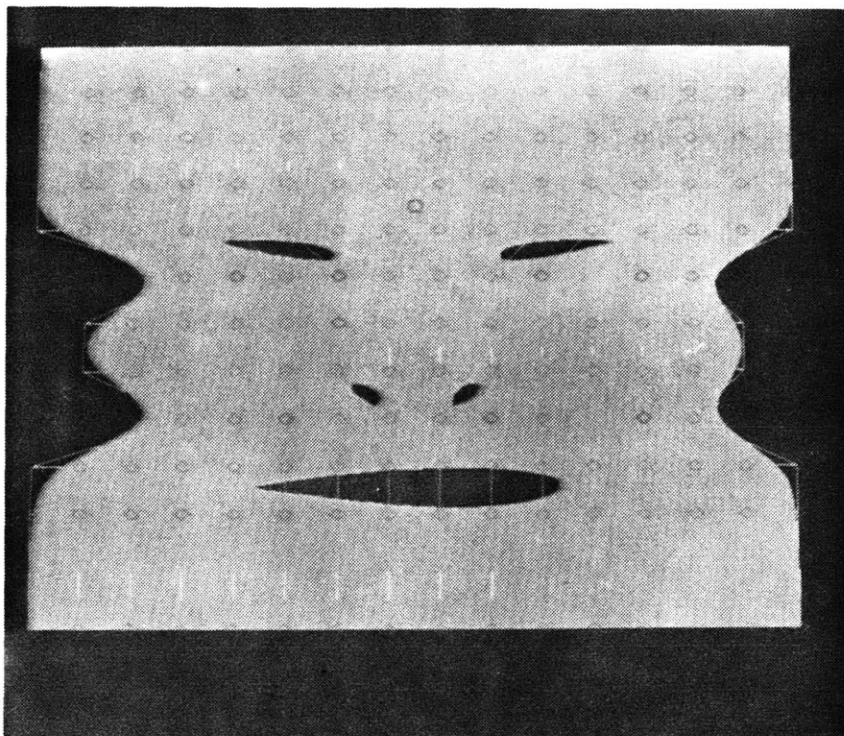


(B)



(C)

(4.5) TRIMMED B-SPLINE PATCHES[13]



(4.6) RECTILINEAR FACE PATCH

4.3 SPLINE-MOLDER: A TOOL FOR SHAPING B-SPLINE SURFACES

The grid of the *rectilinear face patch* does not look much like a human face. To shape this grid into human form, a spline surface modeling tool had to be created, *spline-molder*. Although *spline-molder* was created to shape *masks*, it functions as a general three-dimensional spline surface modeler, taking advantage of the easy interactive ability of the B-spline. Any size patch or groups of patches can be read in and modified by pulling the control points around in three dimensions. Trimming curves are specified in the data file.

Many of the tools created to mold the surfaces carried over to the expression editor, *face*. The most challenging problem encountered in designing these tools was the interface to the user — how to deal with the three dimensions when both the display and input hardware are two-dimensional devices (not, by any means, a new problem), while keeping the interface

fast, intuitive, and simple to use. A way to manipulate the data in “real” time, in three-dimensions was necessary, a 3D cursor.

THE 3D CURSOR

The *3D cursor* makes possible interactive manipulation of three-dimensional data, in real time, with visual feedback to the user. This cursor is used to move object points in space — in the space of the objects, known as *world coordinate space*. The cursor is implemented using a tablet, a two-dimensional input device, and combinations of buttons on the puck to select different functions, namely grabbing control points, moving the camera (which is bound to a sphere), and writing out the current configuration of control points to a file. The *3D cursor* allows the user to manipulate the object data, in this case B-spline control points, in “space” in a visually and kinetically interactive fashion, never having to enter numbers for the point positions. The system is fast, allowing the user to see the affect of her/his manipulations in near real time. All of these features are important in the design of the facial expression editor, in which ease of use and predictability of actions and their reactions are key. The tool is designed to be as obvious as possible so that the user can concentrate on expressiveness.

In order to see the object being manipulated in three dimensional space, the user must settle for a two dimensional perspective projection of the object onto the graphics screen. This is the nature of computer graphics and most other imaging technologies. But, the model being molded is three dimensional, so in order to properly shape it, the user must be able to work in all three dimensions. The input device, the tablet, is also limited to two dimensions, the *xy* viewing plane. So, with the *3D cursor*, the user can only move the surface in two dimensions at a time, in planes parallel to the view plane. The missing third dimension problem is dealt with by adding a camera movement function to the cursor, making it quick and simple to change the current viewing plane. With the viewing plane, the series of two-dimensional planes of manipulation are also changed. The depth of the

point is always held constant during the dragging movement. To move the point along the depth axis, the angle of the camera, and thereby the viewing plane, must be moved so that the depth axis becomes parallel with the viewing plane.

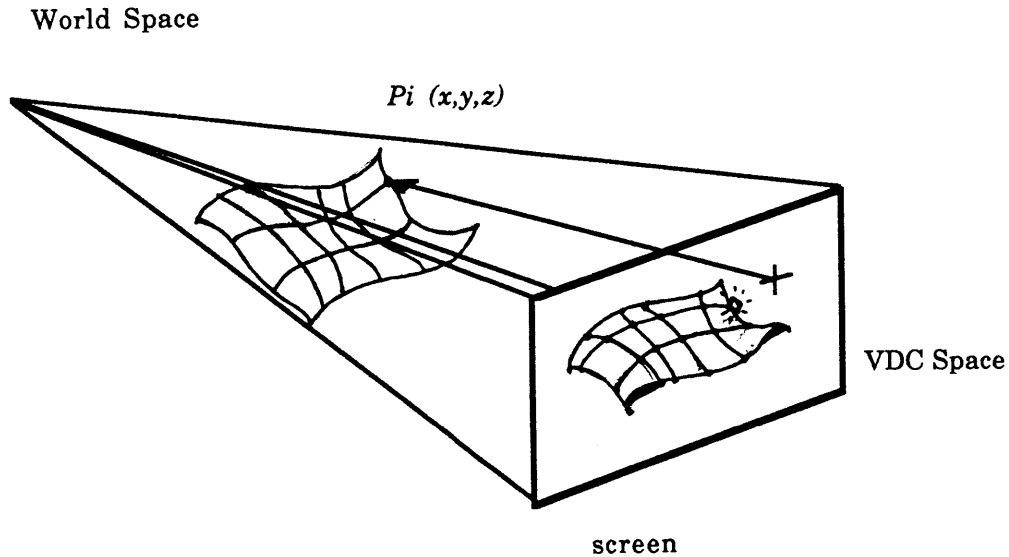
To select a point, the cursor casts a “ray” out into world space from its viewing plane position on the screen, searching for the closest point to this ray. The closest point, ignoring depth, is highlighted, and if grabbed, its (xyz) coordinate in world space is returned.

The *3D cursor* works in two different coordinate spaces, world coordinate space (or *object space*), *WC*, and *perspective space*, *VDC*, for virtual device coordinates, a *Starbase* term. The cursor reaches back and forth between these two spaces, converting a point from *WC* to *VDC*, moving it, then converting it back to the world coordinate system to become part of the object again. It is necessary to do this constant conversion in order to have true three-dimensional information which is no longer available in the *VDC* space after the perspective transformation (see [32]). The *z*-coordinate from *WC* is necessary to perform the inverse transformation on the cursor-ray, *WC* to *VDC*, standing in for the depth information which was thrown away in the perspective transformation.

4.4 CONTROL POINT CONSTRAINTS

A face/mask can now be created using as tools the trimmed bicubic B-spline skin model and *spline-molder*. Each *ActionUnit* structure in *face*, described in Chapter 5, contains a list of the control points which it affects. The number and placement of the B-spline control points is fundamental to the functioning of the *mask* model and must be shaped with the Action Units of FACS in mind.

In addition to the control points needed for the *ActionUnits*, other control points for creating special features of the face, such as the furrows, and for forming the general shape of the face/mask, must also be considered when building the model of the face. The *mask* model, with its constraints, was developed in three steps.



(4.7) THE CURSOR-RAY BETWEEN COORDINATE SPACES

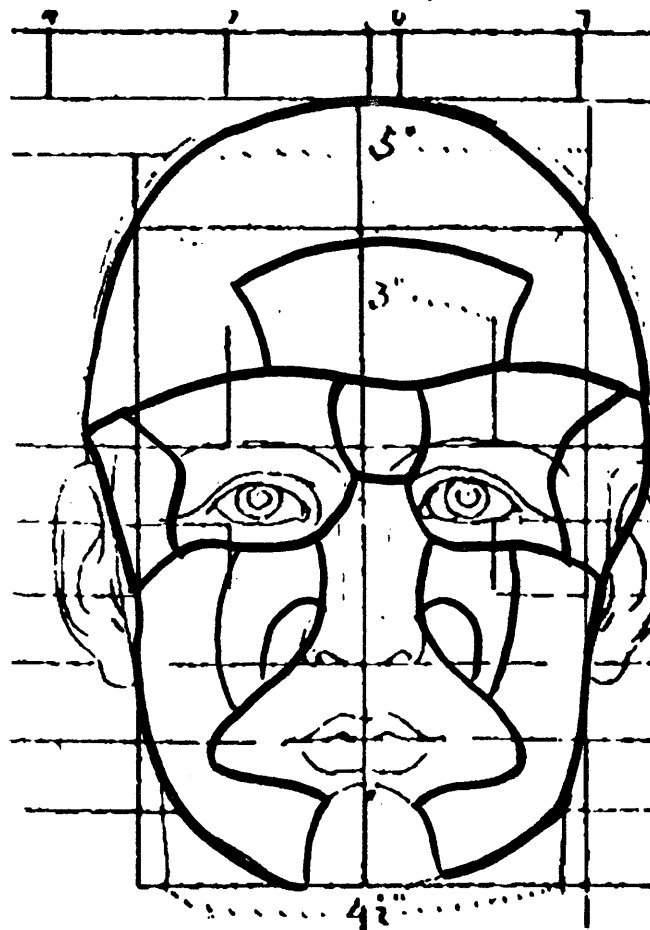
STEP I:

The design of the face model began by breaking down a canonical face into small, logical units. Platt, for OASIS/F, divides the face into area groupings called FACE-REGION's. These divisions are based on the Action Units of FACS. "The FACE-REGION structure contained all the information about a logical section of a face... the AU's are best defined in terms of the lowest sub-regions..." [29].

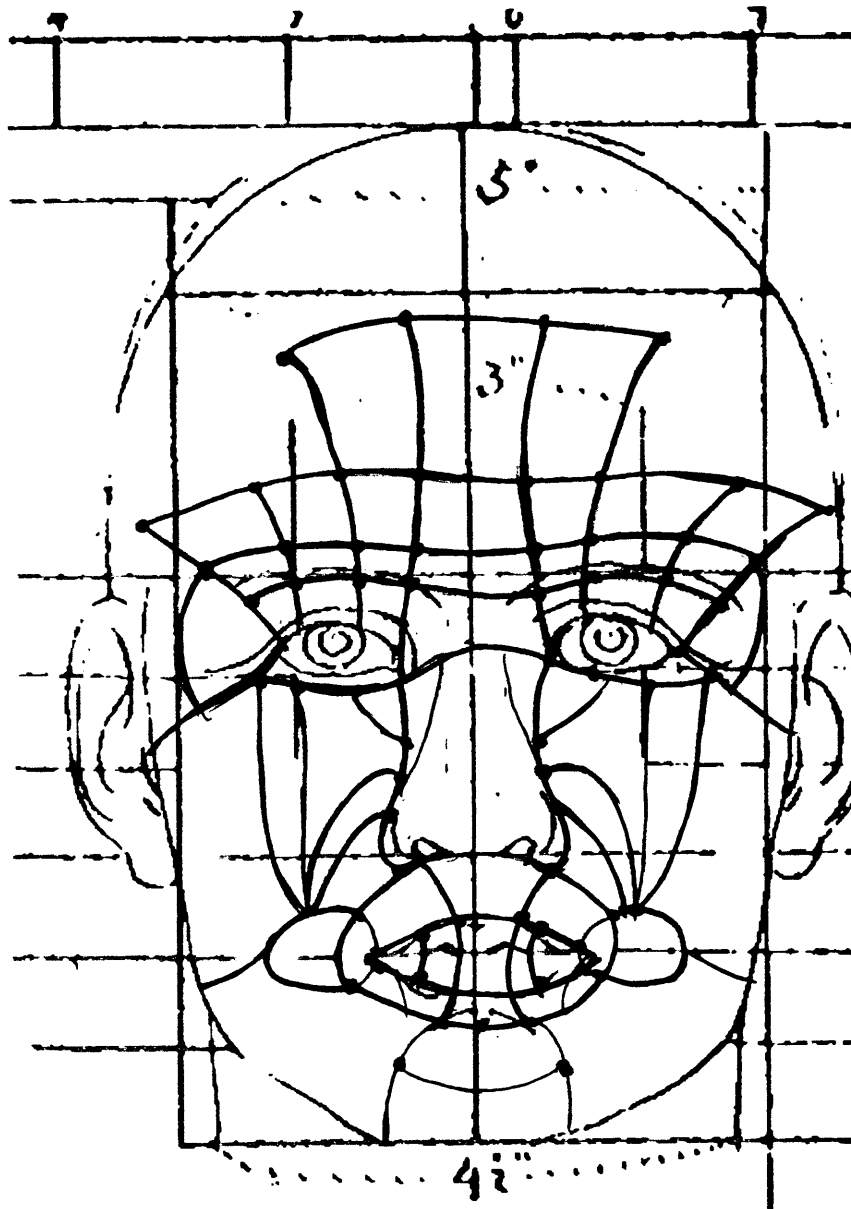
As a first visualizing step for the face model, Platt's FACE-REGION's were applied to a generic face. Working from his diagrams, the face was divided up along the FACE-REGION borders. Many of the region borders mark furrows and other areas where there are changes of continuity in the surface. The original multiple patch model of the skin used these FACE-REGIONS as borders for the individual patches, but because of the

difficulty of “tying” these patches together with the *Starbase* software, this initial approach was abandoned for the large, single patch model.

The face was next divided into “sub-regions” which represent the the areas directly affected by the different Action Units. These sub-regions were compared to the descriptions of each Action Unit, marking locations, directions of pull, and the zones which each AU affects on the face. A logical grid of points was created by overlaying these smaller divisions onto the canonical face and placing control points at each crossing. From this sketch, a first surface model could be created for experimentation.



(4.8) THE FACE-REGION'S



(4.9) FACE WITH "SUB-REGIONS" AND CONTROL POINTS

By moving a group of control points laid out for a given AU and comparing the result with one of Ekman's pictures or written description of the AU, it was possible to fine-tune this initial model, adding or subtracting control points, and to develop a sense of the

correspondence between control point placement and AU motion descriptions. This first step mapped out the general number and placement of the control points needed to represent the Action Units.

STEP II:

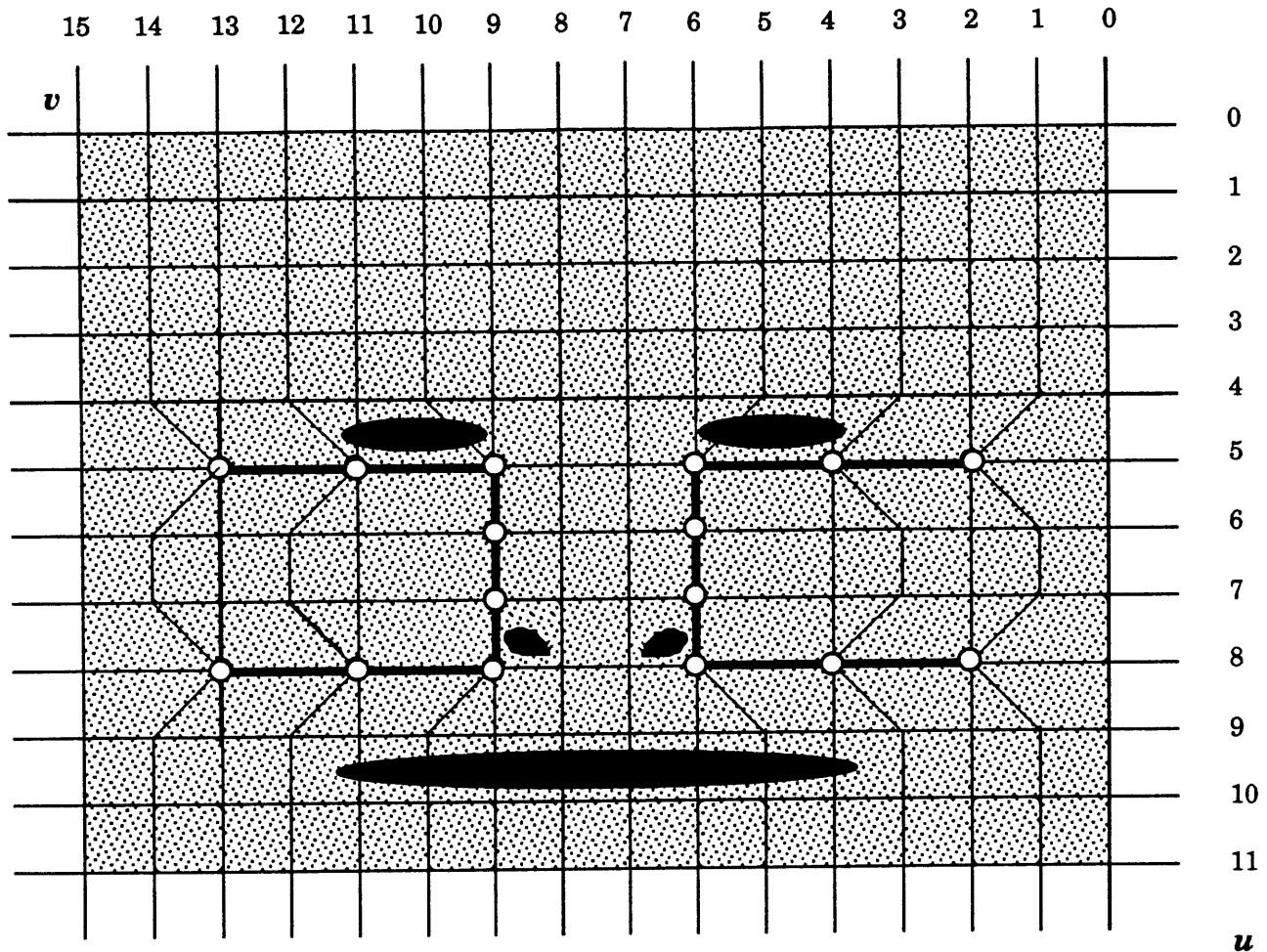
The mesh of control points for the bicubic B-spline patch was next laid out in a rectangular array. The features of the face are defined on this rectangular patch. These features are the orifices of the face — the eyes, nostrils, and mouth, and the landmark furrows and creases of the face — the infraorbital furrow, the nasolabial furrow, and the creases along the side of the nose. The eyes, nostrils and mouth are formed by trimming curves which are used to cut five holes in the patch surface. The furrows are created by doubling up control points, lowering the continuity at that location by one factor.

The minimum dimensions of this array are determined by the row and column in the model which require the greatest minimum number of control points to function. Once the control points needed for the *ActionUnits* were in place, the rest of the control points needed to define the face were considered.

In the model, the minimum number of control points required along the horizontal component of the array was determined by the rows containing furrows, furrows requiring rows of doubled control points. Furrows are created by overlaying adjacent control points along the horizontal axis of the array. In figure 4.11 below, rows 5 and 8 contain the control points for the infra-orbital furrow and the naso-labial furrow respectively, which determined the minimum horizontal value. These rows had 10 original control points, to which 6 additional doubling points were added, making 16 the minimum number of points needed.

The vertical dimension was set by the third factor in the model's layout, the number of control points necessary to give shape to the face. Columns 6 and 9 in figure 4.11 demonstrate this, containing the doubled control points required to create the creases along

the sides of the nose, plus additional points for the forehead and chin. A column of 12 control points, determined by the three factors — the points needed for the *ActionUnits*, the furrows, and the special features of the face — provided enough control points to allow for general face shaping with no additions. Hence the dimensions of the *rectilinear face patch*, a 16x12 array, were set.



rectilinear face patch

- | | | | | | |
|---|-----------------------|---|----------------------|---|---------------|
| ○ | doubled control point | — | furrow | ▒ | patch surface |
| ● | hole | + | single control point | | |

(4.11) RECTILINEAR FACE PATCH II

STEP III:

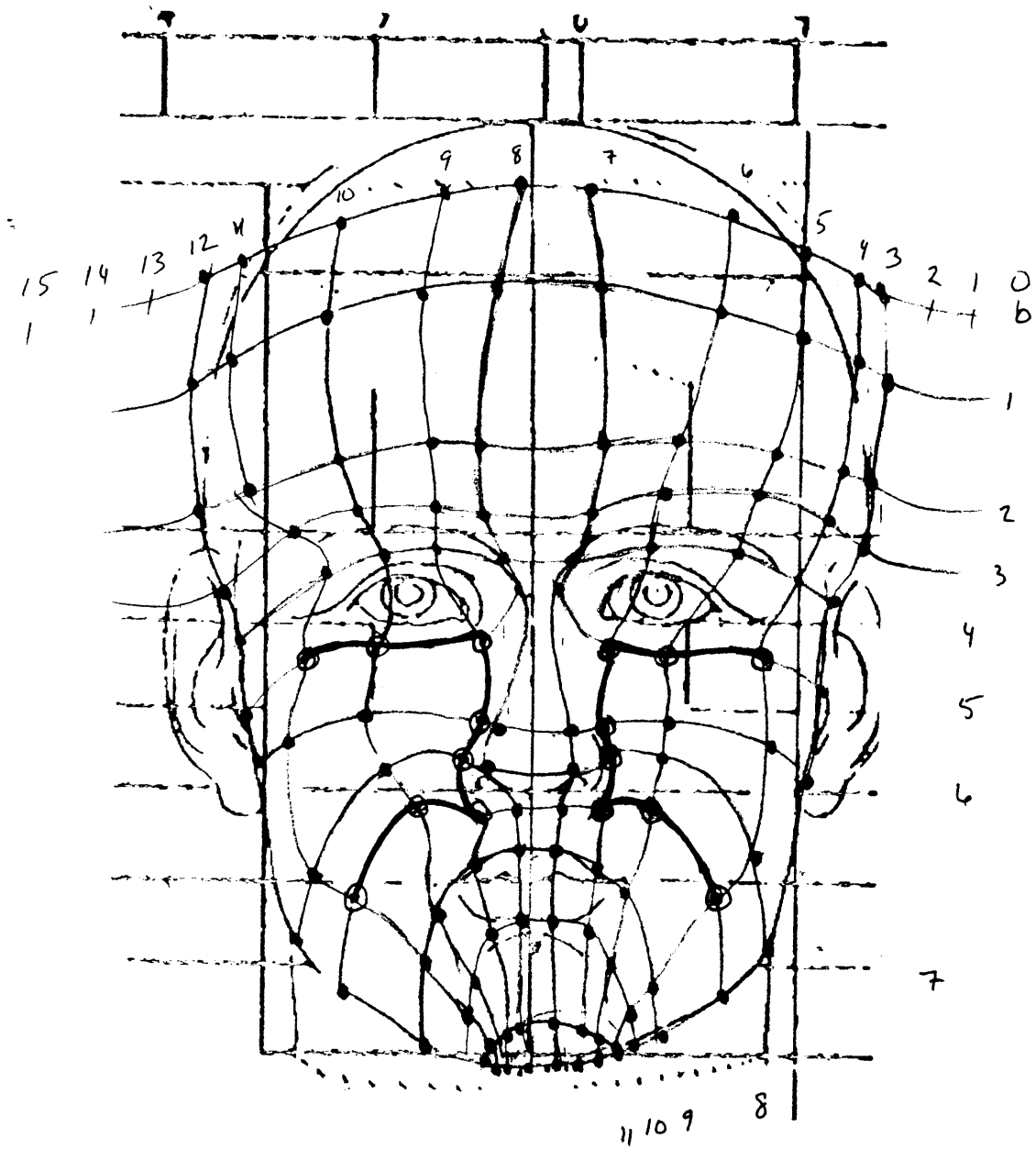
The grid of control points is next arranged to make the surface appear and work like a human face. Each control point is restricted in its placement, destined to represent a specific part of the face, irrespective of its final proportions and shape.

The construction of the *face* model must take into account both the *conformation* and the *expression* parameters needed to construct a realistic face.

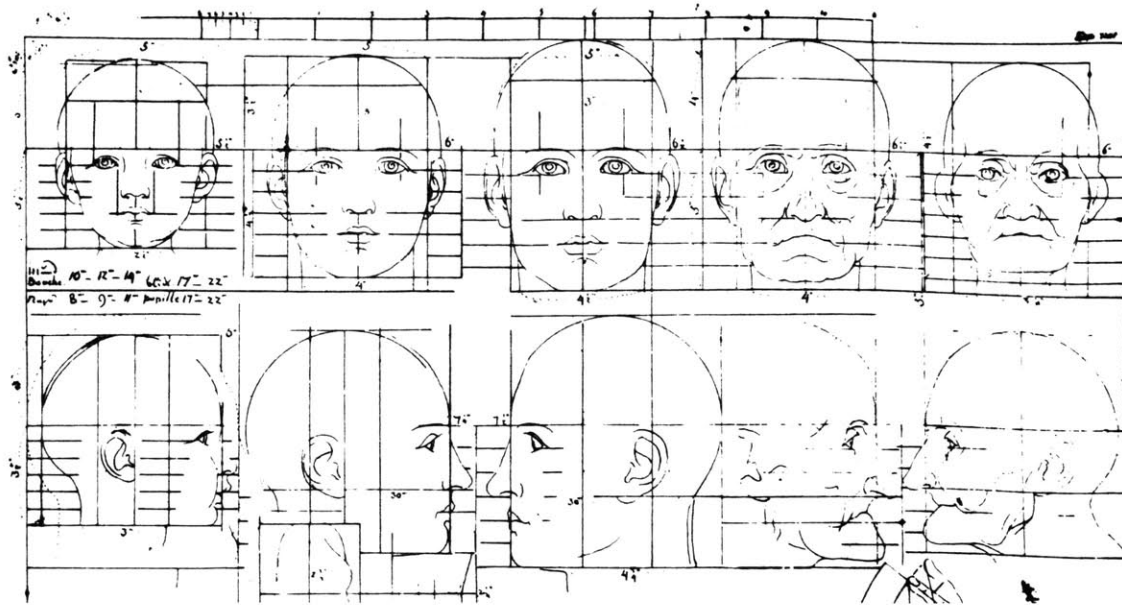
The conformation parameters are chosen when modeling the face and are concerned with measurements such as nose length, eye separation, forehead width... etc. They define the face at the "neutral" state. The expressive parameters, variable, are not concerned with the shape of the face but with its possible movements...[34].

Figure 4.12 shows a two-dimensional layout of the control points superimposed on the canonical face. Each rendition of the face should maintain this relative positioning of the points in the *xz*-plane, adjacencies being fixed. Depths and distances *between* adjacent points may vary, and indeed must in order to create different facial topologies.

The features of the face must also be restricted in their arrangement. Basic human proportions and composition need to be maintained for the face model to function properly with *face*. Though perhaps obvious, the constraints on the conformation parameters include: two eyes, one on each side of the nose, located approximately midway down the face, one nose, located in the middle of the face with the root of the nose between the eyes, one mouth, directly below the nose, two cheeks, two eyebrows, and a chin below the mouth. The eyes, for example, may be wide set or close, but they must be reasonably and realistically placed. On the other hand, horns, warts, or other anomalies would not negatively affect the model. With refinement [9], some interesting "monsters" would be possible.



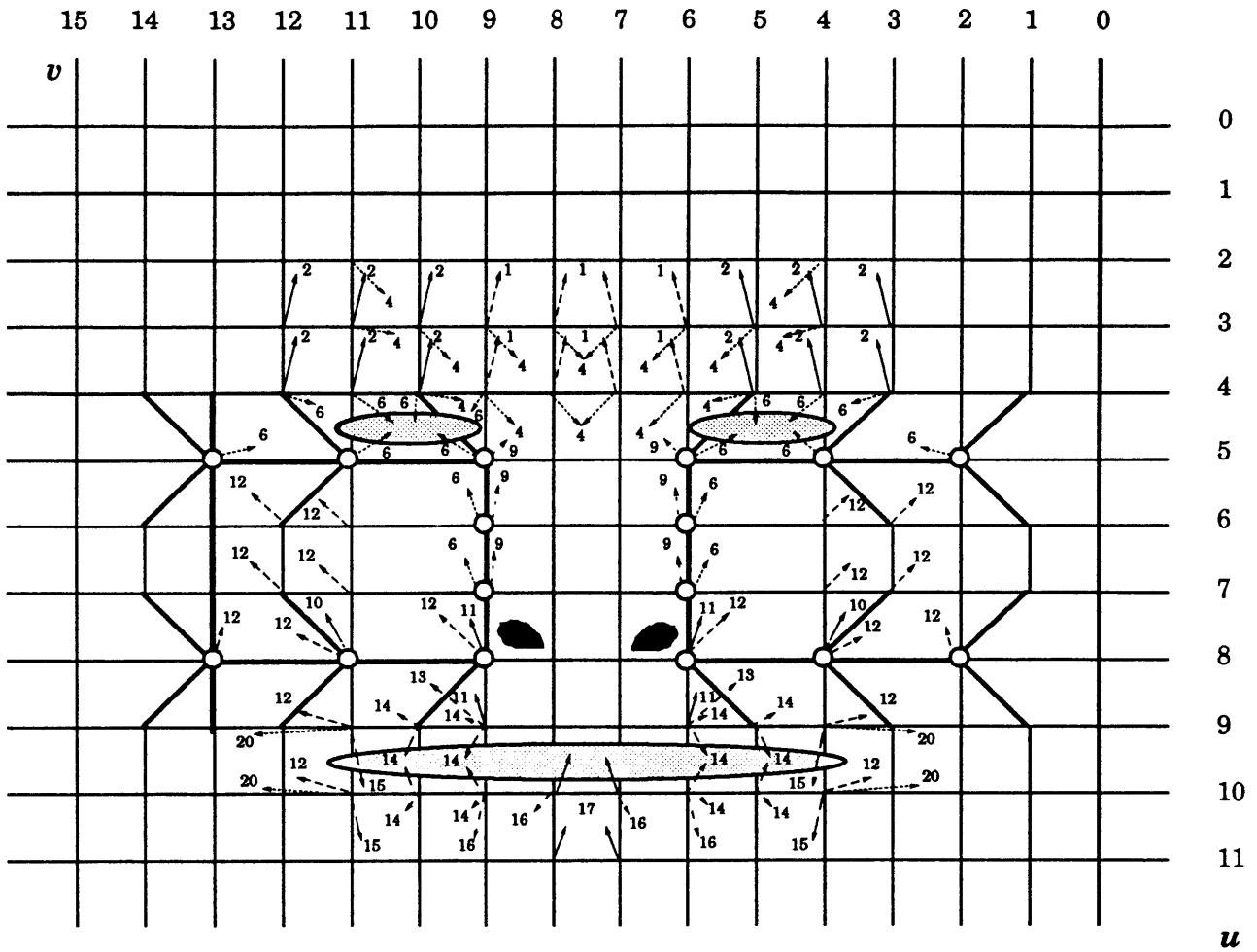
(4.12) FACE WITH CONTROL GRID



(4.13) PROPORTIONS OF THE HEAD AND FACE (JOHANN GOTTFRIED SCHADOW) [18]

The features, too, are confined to a fixed relationship of adjacent points. Most critical to the functioning of *face* is the strategic placement of the control points necessary to the AU's. These control points must function both to identify the features of the face and to realistically distort this face into expressions when moved in a constrained manner.

The most difficult area of the face is the surface surrounding the mouth. This area is particularly difficult because of the complexity and number of motions possible with the mouth. Many control points are required in this region, even with no lips. To add to the complication of this area, the hole created by the trimming curve to form the mouth does not always respond to control point motion in a predictable and desirable manner. Point placement and movement constraints therefore need to compensate for the unwanted affects that can occur. The following figures, 4.14 and 4.15 show the correspondence between the control points and the AU's. The numbers are the AU numbers from the FACS manual [6] and the arrows indicate the approximate direction in which the points will be moved by the indicated AU.

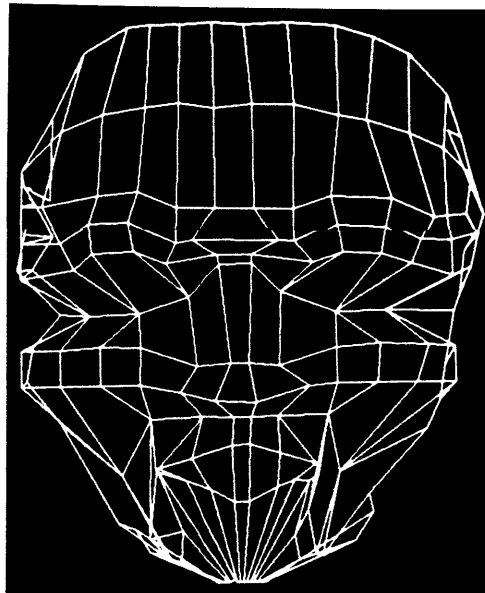


(4.14) AU/CONTROL POINT CORRESPONDENCE I

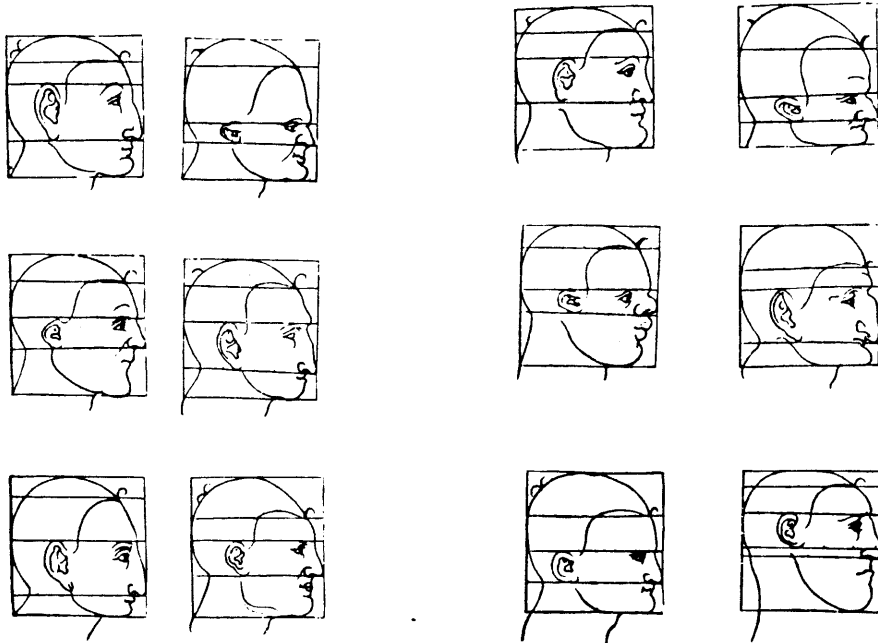
ActionUnit	Left u v	Right u v
1	(3,9) (3,8) (4,9) (4,8)	(3,7) (3,6) (4,7) (4,6)
2	(3,11) (3,10) (4,11) (4,10)	(3,5) (3,4) (4,5) (4,4)
4	(2,11) (2,8) (3,10) (3,9) (3,8) (4,9) (4,8) (3,11)	(2,7) (2,4) (3,7) (3,6) (3,5) (4,7) (4,6) (3,4)
6	(4,12) (4,11) (4,10) (4,9) (5,9) (5,11) (5,13) (6,9) (7,9)	(4,6) (4,5) (4,4) (4,3) (5,6) (5,4) (5,2) (6,6) (7,6)
9	(5,9) (6,9) (7,9)	(5,6) (6,6) (7,6)
10	(8,11)	(8,4)
11	(8,9) (9,9)	(8,6) (9,6)
12	(9,11) (10,11) (8,9) (8,11) (8,13) (7,12) (7,11) (6,11) (6,12)	(9,4) (10,4) (8,6) (8,4) (8,2) (7,4) (7,3) (7,2) (6,4) (6,3)
13	(9,9)	(9,6)
14	(9,10) (9,9) (10,10) (10,9)	(9,6) (9,5) (10,6) (10,5)
15	(9,11) (10,11)	(9,4) (10,4)
16	(10,9) (10,8)	(10,7) (10,6)
17	(MEDIAL) (10,8) (10,7) (11,8) (11,7)	
20	(9,11) (10,11)	(9,4) (10,4)

(4.15) AU/CONTROL POINT CORRESPONDENCE II

The finished, shaped surface of the face is the *mask*. The *mask* model is extremely versatile. Each control point must be placed according to its conformation and expressive parameter constraints, but again, this placement is only relative. Like the human face, the different regions can conform to a variety of shapes and proportions, potentially allowing the creation of a full spectrum of facial types at the hands of a skilled sculptor. The restrictions on the control point placements correlate directly to the restrictions and rules for the natural construction of the human face. Like the human face, all the variations should work as long as these basic rules are followed.



(4.16) THE MASK "GOODWIFE" WITH CONTROL POINT GRID



(4.17) VARIOUS FACIAL PROPORTIONS (ALBRECHT DUERER) [18]

CHAPTER 5:

THE MUSCLES (UNDER THE SKIN)

While the meaning of a gesture depends on the process that is viewing it, the hard question is to find an appropriate set of parameters for controlling a complex process, for example, facial expressions.[39].

In earlier chapters, the muscular structure of the human face was discussed, as well as the quantization of groups of muscles by FACS into Action Units. The functioning of the facial muscles is, like most biological systems, immensely complex. In order to create an articulated figure for computer animation, an exact simulation of the anatomical structure of the muscles, neurons, and joints is not necessary. What is desirable is to have a system with a few dynamic parameters that emulate the primary characteristics of the anatomical structure.

Now that the design and workings of the skin model have been described, the next step is to manipulate this skin to create expressions. The point of *face* is to facilitate these manipulations by presenting a logical analogue to the workings of the human face. This is done by creating an abstract model of the facial muscles. The muscles of the face, described

in detail in Chapter Two, are complicated in their actions and interactions. A simulation would require models of bone, SMAS [28], and the muscles in great detail, down to the myofibrils and the individual fibers [15]. We therefore return to the idea of *emulation* as the concept behind *face*.

The second phase of building the face model was the design and construction of the active components which move the face through its possible expressions. The Facial Action Coding System takes the facial muscles and abstracts them into the simplest expression elements, the Action Units. Each AU is the smallest single motivator necessary for creating an expressive change on the face. An AU may correspond to a single muscle or a group of muscles, or the same muscle may have more than one AU assigned to it. FACS strives to quantify facial expression using these AU's. Little research has been done on measuring facial movement, so in creating *face* it was necessary to use Ekman's AU's and reverse the information provided by FACS, finding parameters for expressive *motions*. From FACS, structures containing direction vectors, placements, motion limits, and other necessary information were created and filled in, becoming the basis of *face* as an expression editor.

5.1 QUANTIFYING FACIAL MOVEMENT

5.1.1 REVERSING FACS

The Facial Action Coding System manual defines and teaches FACS, going through each Action Unit, one at a time. Each AU is explained and illustrated with photographs and with drawings that show the approximate muscle locations and their directions of movement. In the introduction to the FACS manual it is stated,

The illustrations are not meant to be taken literally— they exemplify but do not typify. The exact appearance change will vary from one person to another depending on their bone structure, variations in the facial musculature, fatty

deposits, permanent wrinkles, shape of features, etc. There are common elements across people in the changes which occur and it is these elements which you will learn.[6].

The manual divides the descriptions of the AU's into: "Appearance Changes", "How to do the Action Units", "Minimum Requirements for Scoring", and for some AU'S "Optional—Intensity Scoring". The illustrations and the first section for each AU, "Appearance Changes", provide the primary information for implementing FACS in *face*. The "Minimum Requirements for Scoring" sections is also useful in some instances for quantifying the facial descriptions.

The types of descriptions given in "Appearance Changes" [6] include:

- *the parts of the face that have moved and their direction of movements*
- *the wrinkles which have appeared or deepened*
- *the alterations in the shape of the facial parts.*

As an example, below is the "Appearance Changes" section for AU 4:

- 1) *Lowers the eyebrow. In different instances it may be just the inner portion of the brow which is lowered, or it may appear to be the entire brow which is lowered.*
- 2) *Pushes the eye cover fold downwards and may narrow the eye opening.*
- 3) *Pulls the eyebrows closer together.*
- 4) *Produces deep vertical wrinkles between the eyebrows. In some people the wrinkles between the eyebrows may not be vertical but at a 45 degree angle, or both angled and vertical. May also produce one or more horizontal wrinkles at the root of the nose. If the vertical, angled, or horizontal wrinkles are permanently etched, they will deepen.*
- 5) *May produce an oblique wrinkle or muscle bulge running from the middle of the eyebrow down to the inner corner of the brow.*

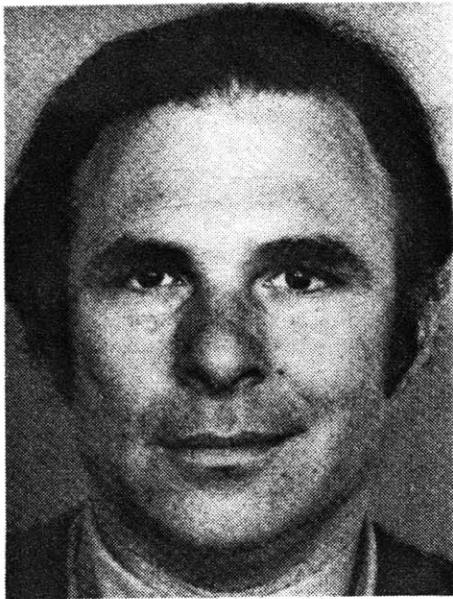
Sometimes the facial action shows appearance changes (1) and (2) and almost no evidence of appearance changes (3) and (4); or the reverse. Either way, if the brows are lowered and drawn together, score AU 4.

This excerpt gives an example of the kind of information available in the FACS manual which can be reinterpreted as movement constraints. All of this information depends heavily on the ability of the human notator to recognize these movements and landmarks, and to interpret the difference between the moved face and the *neutral face* [6] into quantities.

What FACS has tried to overcome is the cultural interpretation of these movements as actual expressions, reducing the interpretation to that of movements in degrees of AU activation. What FACS has *not* done is to measure these movements other than on a qualitative scale of relative comparison (as judged by the notator) to the neutral face. These qualitative comparisons are scored as *slight, marked, pronounced, severe, extreme, and maximum* [6].

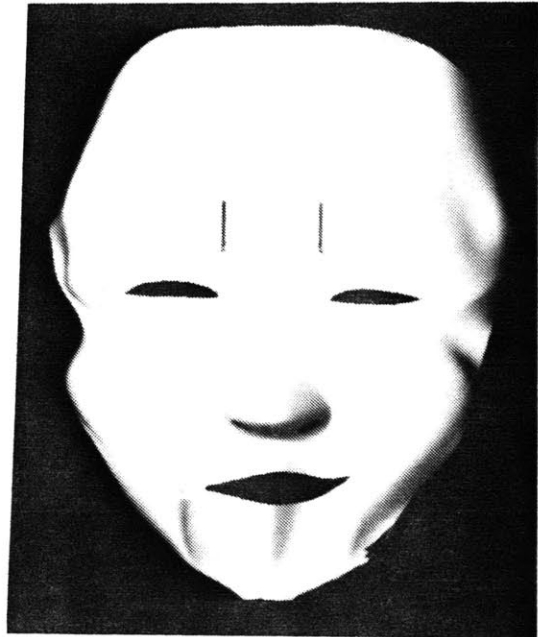
5.1.2 METHODOLOGY

It was necessary to translate FACS into some numerical form and rework the AU descriptions into causes, rather than measures, of expressive movements on a face. Lacking any numerical data on the lengths and directions of contractions of the facial muscles in creating expressions, this information had to be derived and developed empirically, based on what is available from Ekman's research, such as the "Appearance Changes" sample above. Comparing a given movement by *face* on the *mask* to the pictures illustrating an AU on a real human face in the FACS manual was one of the important tools used in finding the numerical quantities of movement which correspond to a FACS description, especially the maximum limit of movement, *lmax*, to keep the face from appearing grotesque or "exploded" .



L12X

A) FACS MANUAL: AU12L



B) MASK: AU12L

(5.1)

AU TERRITORIES

Many AU's, as they approach the upper limits of their motion extents, start to affect areas which aren't part of the AU region under less extreme circumstances. These outer regions are sometimes mentioned in an AU description, but, like FACS itself, it is often necessary to perform the AU with one's own face and to observe it on others' to get a full description of the AU, and, in the case of *face*, to properly contain the full range of movement in the numerical description. Touching the areas of one's face around the AU area during activation is very useful for determining the extent of an Action Unit's realm. AU 4 is a good example of this. The description given above does not mention that at *maximum* flexion, the skin at the outer eyebrows begins to pull in. The description given only describes the affect of this AU on the inner and middle brow regions. If you perform this motion, pulling your eyebrows in and

down as hard as you can, lightly touching your outer brow and temple, you feel this skin pull in slightly in a horizontal direction. Whereas this subtle information is not important to FACS's purpose as a scoring method, this outer limit motion is accounted for in *face* because it adds to the realism of the model.

VECTOR DIRECTIONS

Besides amplitudes and areas of motion, the direction of pull of each AU on the skin must be specified for *face* to create expressions. FACS gives some indication of direction of movement in its descriptions. "Lowers the eyebrow... Pulls the eyebrows closer together..."[6] The illustrations in the FACS manual also show the "origins" and "insertions" of the AU's. Platt's example, the OASIS/F system, defines the directions of motion of the AU's generally as *central*, *lateral*, *left*, *right*, and *medial*, also based on FACS. But, the most important system used to determine direction vectors for *face* was the actual muscles of the face.

Each *face* "muscle" has a basic direction of pull which approximates the anatomical muscle(s) it represents. This dominant pull vector is a two dimensional shift in direction. The muscle vectors are then broken down into piecewise linear approximations to the curved shape of the skull along this dominant direction, creating a three dimensional path. The linear vectors are grouped into bundles which pull the skin *along* the "bone", not through it. In certain cases, such as the cheeks during AU12 (smiling), extra pulling vectors are added to create puffing, which does not automatically occur using the spline model for the skin.

mathematical description is necessary. This is useful in cases such as the making of “unique” characteristics like extra high eyebrow raises (“stretch”^{*}), emulating some deformity or limitation, or designing Action Units for a different, non-human, shape of face. The user can alter the “realistic” facial motion descriptions for a special character or case. These new AU and muscle files would then have the special parameters designed for this character. A *mask* using these files “knows” how it moves as this other kind of face. Characteristic expressions can also be stored in scripts and be recalled repeatedly, facilitating the building of a set of consistent expressions, an important factor in character animation and sociological research. For the latter, the different cultural expression groups also be entered and stored for reference and comparison.

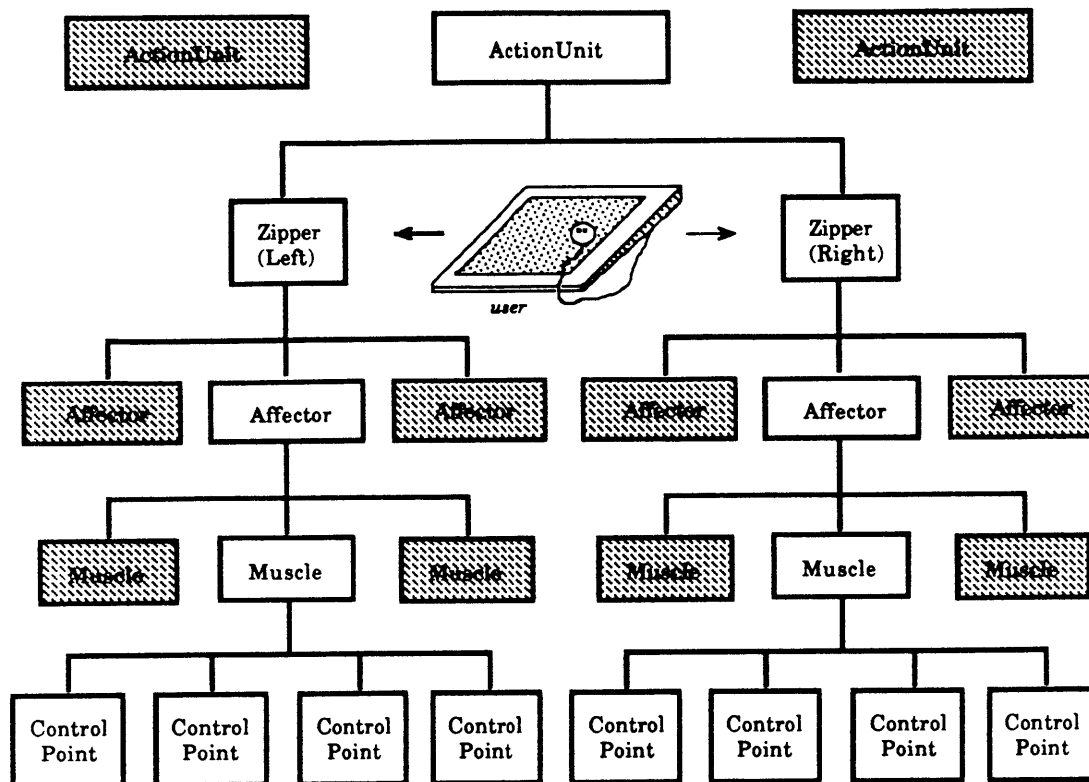
5.2 THE STRUCTURES

Computer systems can, in principle, be designed and programmed to alleviate the problems of notation, but before this can be done, the notation must be entered into the computer. [12]

Finally, in designing a computer graphics model of the face, it is required that data structures be created as a means of representing all the information in the model. *face* has been designed based on FACS, using the Action Unit as the primary data type.

The Action Unit is broken into a structure hierarchy of four levels of abstraction: the *ActionUnit*, the *Zipper*, the *Affector*, and the *Muscle*. The *ActionUnit*, the highest level structure in the chain of expression motivators, is the abstract grouping of muscles into unit motion-areas of the face. It is the object description which contains pointers to the other motion structures listed above. The *ActionUnit* structure points to a *Zipper* structure, the user input control mechanism. *Zippers* allow the user to specify which muscles are being activated and to what degree, controlling the movement the system will execute.

* as in “stretch & squash” — animation terminology for exaggerated reactions to motion.



(5.3) THE ACTIONUNIT HIERARCHY

Each *ActionUnit* in general has two *Zippers*, due to the bilateral symmetry of the face. One side of the face can often act independently of the other, or, only one side is able to perform the action. An example of this is demonstrated by “winking”, which uses only one eye, while both eyes are active during “blinking”. Both are performed using the same AU. Medial Action Units, those which are central and therefore not symmetrical about the face, will only contain one *Zipper* in the *ActionUnit* structure. The parameter settings input by the user via the *Zipper* are passed to the *Affector* structure. Each *Affector* contains a group of *Muscles* which act in unison. An *ActionUnit* will contain more than one *Affector* when multiple *Muscle* bundles are appropriate at different levels of activation.

The *Muscle* is modeled to act like the strands of an anatomical muscle, having one degree of freedom which pulls on the surface of the face. The *Muscle* is the lowest-level structure, directly moving the B-spline control points. *Affectors* group these unidirectional *Muscle* “strands” into active bundles when multi-directional muscles are called for. Figure 5.3 shows a diagram of the structure hierarchy tree.

MUSCLE

The Muscle, the lowest level structure in the Action Unit group, contains three important variables:

- *dir*- a unit vector which indicates the direction of pull of the Muscle (insert to origin),
- *mobility*- which is a scaling factor indicating what percent of the total “force” being applied to the ActionUnit will affect this Muscle, and
- *pts*- an array which is a list of pointers to the control points of the “skin”, the bicubic B-spline patch, to which the “muscle” is attached.

A single *Muscle* may move as many as ten control points. The key feature of a *Muscle* is that it has a single direction and scaling factor. An *ActionUnit* may be composed of many *Muscles* moving in different directions, but each *Muscle* is itself limited to its unique direction. The *Muscle* structure is anatomically analogous to a muscle strand rather than to a complete muscle (as implied by the name) because of its unidirectional, linear nature.

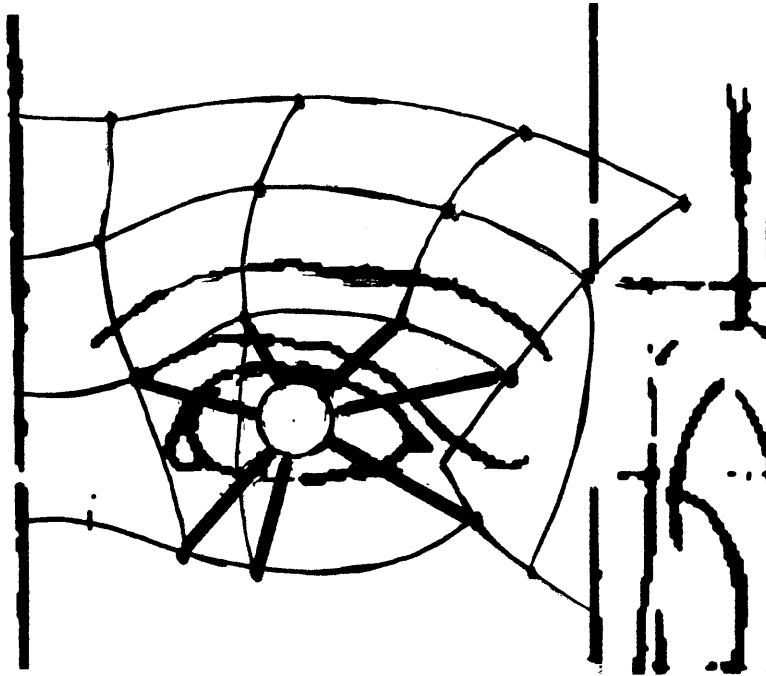
AFFECTOR

The next level up in the structure tree, the *Affector*, contains:

- *mus*- an array listing *Muscles*, and

• *lmin* and *lmax*- delimiters for its range of activation within the total range of the *ActionUnit*.

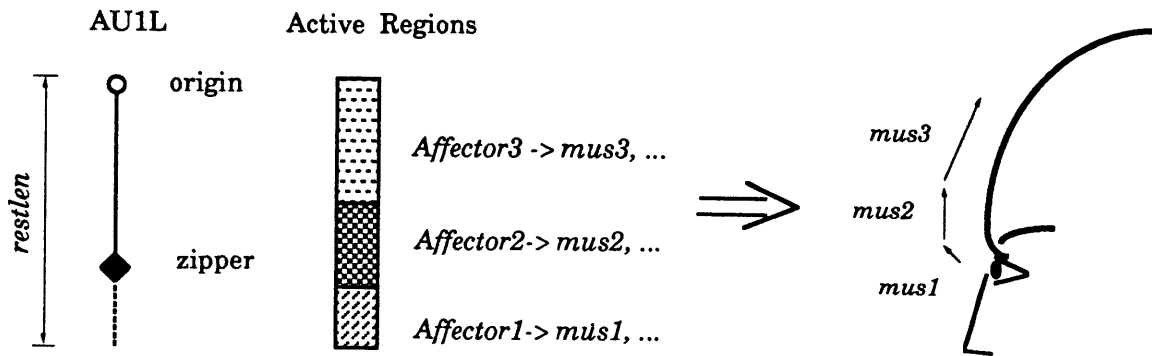
While the *Zipper* which contains this *Affector* is positioned between the delimiters of the *Affector*, ($lmin \leq zipper \leq lmax < zipmax$), the muscles listed in that *Affector* will be allowed to change in length. The *Affector* is especially important in emulating the multidirectional muscles, being used to “bundle” the muscle “strands”, such as are found in triangular or sphincter muscles.



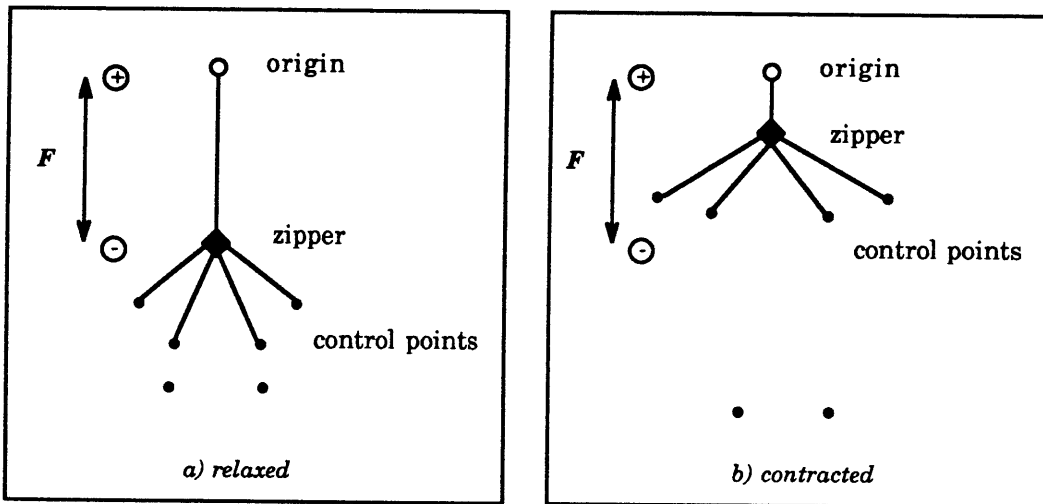
(5.4) THE MUSCLE VECTORS AND CONTROL POINTS OF AU 6, A SPHINCTER MUSCLE

The muscle vectors are grouped under one Affector, pulling in unison toward the center, like a “draw-string” bag.

Affectors are also used to activate the muscles in the outlying areas near *maximum* contraction, and to separate the *ActionUnit* into piecewise linear approximations to a curved path in space (such as the brow). Each *Affector* comes into activation when the zipper has reached the appropriate section in its path.



(5.5) THE ACTIVE RANGES OF THREE AFFECTORS AND THEIR MUSCLE-VECTORS



(5.6) THE ZIPPER

ZIPPER

The Zipper is the bridge between the user-interface of *face* and the underlying emulation structures, the *Affector* and the *Muscle*. On the display, a zipper consists of a virtual slider, a strip with a marker that can be slid up and down, and extensions which attach to the points on the mask which it affects, the control points. It works much like a clothing zipper, hence the name.

The user slides the marker up and down along the strip (the zipper has only one degree of freedom), thus indicating the amount and direction (positive or negative) of “force” to be applied to the underlying “muscles”. Quantity of force is indicated as displacement.

The *Zipper* is also used to keep track of the current state of the *Muscles*. The *Zipper* structure includes information about its current length and therefore the amount of “contraction” of all the “muscles” it controls. Three variables are used in displaying the current state of the AU to the user and in keeping track of the user’s commands to the muscles:

- origin*- a vector indicating the location of the static point of the zipper in space,
- zipper*- a vector indicating the moving part of the zipper, the marker on the screen, analogous to the insertion of a muscle except that a zipper may be attached to a combination of muscles, and thirdly,
- zip_len*- keeps track of the current length of the zipper.

A *Zipper* only affects *Muscles* on one side of the mask. This is to allow asymmetrical motions of the face. Symmetry can also be enforced in appropriate cases if desired. The symmetry flag is contained in the *ActionUnit* structure.

ACTIONUNIT

The highest level of the emulation/interface structures is the *ActionUnit*, which directly correlates to the Action Units of FACS. The *ActionUnit* contains the variables which are symmetrical about the face. These variables are:

- name*- its unique name taken directly from FACS, such as "AU4",
- zipmax*- the maximum length of its two zippers (left and right), limiting the total allowable distortion by this AU to the bicubic patch,
- symmetry*- a flag with the options ON, OFF, or MEDIAL, indicating whether pulling on one zipper will affect the corresponding muscles on the other side of the face (ON) or not (OFF), or indicating that this is a centrally located Action Unit having no symmetrical partner so that *symmetry* does not apply (MEDIAL), and
- side*- an array of two *Zippers*, one for each side of the face.

5.3 EXAMPLE: FILLING THE STRUCTURES

The following is an example of an AU description as given in the FACS manual and the corresponding values put into the *ActionUnit* structures. Action Unit 2, a relatively simple AU, has been chosen as an illustration.

First, The AU description:

Action Unit 2- Outer Brow Raiser

Figure 2-1 shows that the muscle which underlies AU 2 originates in the forehead and is attached to the skin in the area around the brows. In AU 2 the action will be upwards, pulling the brow and the adjacent skin in the lateral portion of the forehead upwards to the hairline.

A. Appearance Changes due to 2

- (1) Pulls the lateral (outer) portion of the eyebrows upwards.*
- (2) Produces a \curvearrowright \curvearrowleft shape to the eyebrows.*
- (3) Causes the lateral portion of the eye cover fold to be stretched upwards.*
- (4) In some people, causes short horizontal wrinkles to appear above the lateral portions of the eyebrows. There may be also wrinkles produced in the medial portion of the forehead, but they will not be as deep as the lateral ones.*
- (5) Don't be confused if the inner corners of the brows may move a bit. This is from the pulling of the lateral part of the brow by 2 and not due to the inner corner being pulled upwards by AU 1.*

Compare the photographs of 2 and 0 and inspect the film of AU 2. Note in both the photograph and film that 2 is often stronger on one side of the face than on the other. (the action of 2 depicted in the film is not simultaneous on both sides of the face. This occurred because of difficulty in voluntarily performing this AU.)

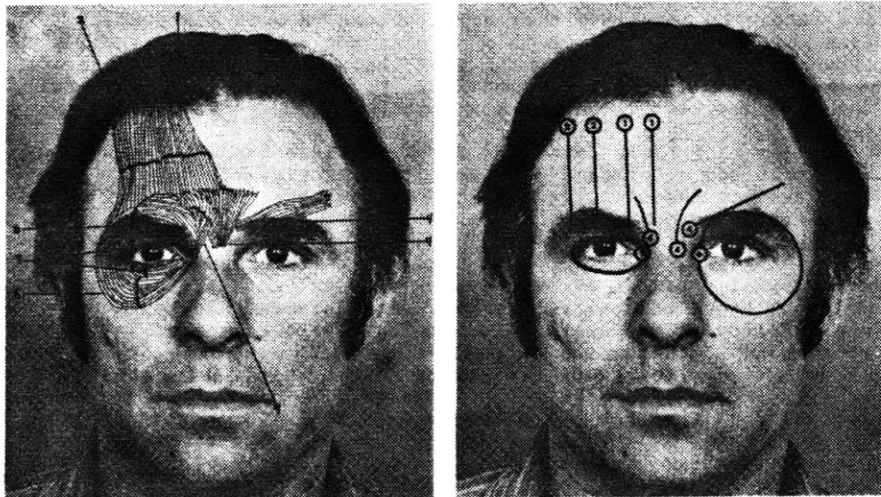


Figure 2.1

Figure 5.6 shows the file storage formats for *ActionUnits* and *Muscles*. These are the spaces which need to be filled in from the "Appearance Changes" information.

ACTION UNIT

<i>symmetry</i>	-	<symmetry> (integer- OFF = 0, ON = 1, MEDIAL = 2)
<i>r, g, b</i>	---	<zipper color> (float)
<i>zipmax</i>	-	<maximum length for zipper> (float)
<i>num_aff</i>	-	<number of effectors> (integer)

left side

<i>origin</i>	---	<origin vector> (3 floats)
<i>zipper</i>	---	<zipper vector> (3 floats)
<i>label.text</i>	-	<label text> (char)
<i>loc.x loc.y</i>	--	<label screen coordinates x, y> (float - VDC coordinates)

effectors

<i>lmin lmax</i>	--	<length limiters on this effector> (float)
<i>num_mus</i>	-	<number of muscles> (integer)
<i>mus_file</i>	-	<name of muscle files to read in> (char)

right side

<i>origin</i>	---	<origin vector> (3 floats)
<i>zipper</i>	---	<zipper vector> (3 floats)
<i>label.text</i>	-	<label text> (char)
<i>loc.x loc.y</i>	--	<label screen coordinates x, y> (float - VDC coordinates)

effectors

<i>lmin lmax</i>	--	<length limiters on this effector> (float)
<i>num_mus</i>	-	<number of muscles> (integer)
<i>mus_file</i>	-	<name of muscle files to read in> (char)

MUSCLE

<i>dir.x dir.y dir.z</i>	---	<x,y,z unit direction vector> (float)
<i>mobility</i>	-	<mobility> (float)
<i>numpts</i>	-	<number of control points attached to this muscle> (integer)
<i>patch.u patch.v</i>	--	<v,u pairs of patch control points>(integer)

(list in row, column order)

(5.7) FILE STORAGE FORMATS

Next we will take the AU description line by line and fill in values in the *ActionUnit* and *Muscle* file structures:

STEP I:

Figure 2-1 shows that the muscle which underlies AU 2 originates in the forehead and is attached to the skin in the area around the brows.

This first line, together with the illustration, provides a great deal of information for the model. First, the locations of the zipper and origin vectors can be entered by locating the AU on the unit *mask* and lining them up near the control points in that area. These values are given in world coordinate points.

<i>left side</i>	<i>right side</i>	
$(0.4, -0.116, 0.41)$	$(-0.4, -0.116, 0.41)$	<origin vector> (WC)
$(0.4, -0.12, 0.18)$	$(-0.4, -0.12, 0.18)$	<zipper vector> (WC)

The label names and locations can also be filled in now.

<i>left side</i>	<i>right side</i>	
AU2L	AU2R	<label text>
$(0.3, -0.12)$	$(-0.3, -0.12)$	<label coords> (VDC)

Figure 5.8 shows the zippers on the model.

Moving on to the next line in the description, some of the *Muscle* variables can be filled in.



(5.8) THE ZIPPERS OF AU2

STEP II:

In AU 2 the action will be upwards, pulling the brow and the adjacent skin in the lateral portion of the forehead upwards to the hairline.

From this line, the dominant direction vector can be set. The value below represents a vertical vector with a slight inward tilt, following the direction of the brow.

$(0.0, 0.008, 0.9908)$

<x,y,z unit direction vector> (WC)

This vector is the same for both sides of the face.

Item (1) of "Appearance Changes" more specifically locates the AU. Control points on the *mask* can be allocated to this *ActionUnit* based on the information given in this line. The

numbers below refer to the array index numbers which access these control points. See figure 4.11 for the location of these points on the *rectilinear face patch*.

STEP III:

(1) Pulls the lateral (outer) portion of the eyebrows upwards.

left side

3 5

3 4

right side

3 11

3 10

from *AU2R.mus*

from *AU2L.mus*

Item (3) indicates that more control points will be necessary. While the points listed in the previous step pull from above the brow, this next line indicates that the skin below the brow also needs to be pulled upward.

STEP IV:

(3) Causes the lateral portion of the eye cover fold to be stretched upwards.

left side

4 5

4 4

right side

4 11

4 10

from *AU2R.mus*

from *AU2L.mus*

One last piece of information from the "Appearance Changes" section was used to fill in the *ActionUnit* structure. From the final paragraph,

STEP V:

Note in both the photograph and film that 2 is often stronger on one side of the face than on the other. (the action of 2 depicted in the film is not simultaneous on both sides of the face.

the symmetry flag can be set to OFF.

0 <symmetry> (OFF = 0, ON = 1, MEDIAL = 2)

Looking back at figure 5.6, there are still blank spaces left to be filled in in the *ActionUnit* structure. These variables are determined by experimenting with the model. Those variables which are still blank are:

<i>num_aff</i>	<number of effectors>
<i>num_mus</i>	<number of muscles>
<i>mus_file</i>	<name of muscle files to read in>
<i>r, g, b</i>	<zipper color>
<i>zipmax</i>	<maximum length for zipper>
<i>lmin, lmax</i>	<length limiters on this effector>
<i>mobility</i>	<muscle's mobility- scaling factor between 0-1>

AU 2 is a simple Action Unit, having no unusual affects near its maximum displacement and having only a simple, unidirectional motion over a nearly planar part of the skull. "...the muscle which underlies AU 2 originates in the forehead and is attached to the skin in the area around the brows."

Because the brow is nearly planar, a piecewise linear approximation to a curved path is not necessary for AU 2. Since the motion of the AU is unidirectional, only a single *Muscle* is necessary. A straight path and a single *Muscle* mean that there will be only one *Affector*. "In AU 2 the action will be upwards, pulling the brow and the adjacent skin in the lateral portion of the forehead upwards to the hairline." [5]. The first four of the remaining

variables can now be filled in. Those variables which are symmetrically identical are not repeated.

<i>1</i>	<number of effectors>
<i>1</i>	<number of muscles>
<i>AU2L.mus</i>	<name of muscle files to read in>
<i>AU2R.mus</i>	
<i>0.2, 1.0, 0.5</i>	<zipper color>

The two file names, *AU2L.mus* and *AU2R.mus*, represent the naming convention for muscle ascii files. In the case of multiple muscle files for the same *ActionUnit*, a number is tagged onto the file name, for example, *AU4L_2.mus*. Each *ActionUnit* has a unique muscle file for either side of the face except in the case of medial Action Units.

The value for the *zipmax* variable was determined by experimentation with the *mask* model, comparing the results of different values with the FACS manual illustrations and human models.

<i>0.258</i>	<maximum length for zipper> (WC)
--------------	----------------------------------

Since this *ActionUnit* has only a single *Affector*, the limits of the *Affector* should extend the entire length of the zipper.

<i>0.0, 0.258</i>	<lmin, lmax - length limiters> (WC)
-------------------	-------------------------------------

And because there is only one *Muscle* in each *Affector*, there is no reason to scale down the “force” being transmitted by the zipper. Mobility is only an issue when there are multiple muscles which may want to receive varying degrees of the zipper force. So, the scaling factor is set to unity for AU 2.

1.0

<mobility>

Finally, the ActionUnit file, *au2.au*, and its substructures are complete. Below are the structure files in their entirety.

MUSCLE

0.0 0.008 0.9908

1.0

4

3 5

3 4

4 5

4 4

0.0 0.008 0.9908

1.0

4

3 11

3 10

4 11

4 10

AU2R.mus

AU2L.mus

ACTION UNIT

1

0.258

0.2 1.0 0.5

1

0.4 -0.116 0.41

0.4 -0.12 0.18

AU2L

0.3 -0.12

0.0 0.258

1

AU2L.mus

-0.4 -0.116 0.41

-0.4 -0.12 0.18

AU2R

-0.3 -0.12

0.0 0.258

1

AU2R.mus

au2.au

(5.9) THE FINAL ACTIONUNIT VALUES

All the files which are to be read in with the *mask* model are listed in yet another ascii file, *<name>.file*. Below is an example of the file name file used with the *mask* "goodwife".

FILE NAMES

data/goodwife.spl

<name of Spline Patch file>

4

<number of Action Unit files>

data/au4.au

<names of Action Unit files>

data/au1.au

data/au2.au

data/au12.au

f1.file

(5.10) FILE-NAME FILE

CHAPTER 6:

MOTION

In the production of character animation that treats living things moving at will, as in the case of humans and animals, it is important to express natural action...[17].

In this chapter the motion of the “muscles” is described. What is meant by motion is the interpolation between the parameter settings of the keyframes to create animated sequences. An algorithm is needed to make these motion sequences appear natural and realistic.

The parameters used by *face* to describe the changes to the muscles which result in motion are:

- which muscles to move,
- the final length of the muscles at the end of a motion, and
- the time taken to accomplish the move.

These parameters are set using the *zipper* interface, then stored in scripts as variable settings within the *ActionUnit* structure hierarchy.

To create motion from these parameters, what is needed is a function to describe the change in length of the muscles over time, breaking the changes down into the amount of displacement which occurs during a single frame's unit of time. An infinite number of mathematical descriptions can be used to implement this function, but finding a function that will interpolate the parameters in a way which closely resembles the natural movement of the muscles narrows down the choices.

6.1 THE MASS/SPRING MODEL

The movement of a muscle depends on a complex interplay of variables, resulting in a contraction. The muscles of the body are often described using the physical analogy of the mass/spring model [15], using force, mass, acceleration, and resistance to describe the response of a muscle to a "move" stimulus from the brain. The quantity of interest to the interpolation scheme is the distance traveled by the insertion point of the muscle over time. This distance is a function of the force applied by the muscle. The force is dependent on the resistance of the attachment, in this case skin, which is analogous to the *spring constant* of the mechanical model, the direction of the force, the mass load on the muscle, the length of the muscle, and the speed with which the muscle is changing in length — variables which are known as the *response* of the muscle. Equation 6.1 shows the mass/spring model representation for calculating the force exerted by the muscle:

$$[6.1]$$

where \mathbf{F} is the force of the muscle, \mathbf{V} is the normalized spring vector, giving the direction of the force, K is the spring constant, represented in the *Muscle* structure by the *mobility*

variable, and R is the response curve of the muscle, a function of l_r and l_c , the rest length and current length of the muscle, respectively.

Force in this case is equivalent to acceleration. This can be seen by the standard mechanics formula

$$[6.2] \quad \mathbf{F} = m\mathbf{a}$$

where m is the mass of the load and \mathbf{a} represents its acceleration. Since the load on a facial muscle is not likely to change, we set m to unity for the face, resulting in:

$$[6.3] \quad \mathbf{F} = \mathbf{a} = R(l)$$

From equation 6.3, the displacement function of the muscle's endpoint can be derived. The change in velocity, $d\mathbf{v}$, with respect to time is given as:

$$[6.4] \quad d\mathbf{v} = \mathbf{a}dt$$

and the current velocity is given by,

$$[6.5] \quad \mathbf{v}_{\text{cur}} = \mathbf{v}_{\text{old}} + d\mathbf{v}$$

where \mathbf{v} is velocity and dt is the change in time. The displacement, dl , is next found:

$$[6.6] \quad dl = \mathbf{v}_{\text{cur}}dt$$

and finally,

$$[6.7] \quad l_c = l_{\text{old}} + dl$$

where l_c again is the current muscle length.

Defining the time step dt to be one frame of the animation, equation 6.7 becomes the interpolation function between the animation key frames. It is $R(l)$, the response curve of the muscle, that now needs to be defined.

6.2 THE RESPONSE OF THE MUSCLE

The most complex part of the system presented is the function which represents the muscle's response in relation to the forces present. This function has as variables

- the current length of the muscle,
- the rest length of the muscle,
- the duration of the movement,
- the composition of the muscle, and
- the type of signals being sent to the muscle by the brain.

Examining the bio-physical model of the muscle, there are two main factors which determine the amount of force, or *tension* which the muscle produces. These are the number of fibers involved in the contraction and the tension produced by each fiber as it contracts [36]. Accurately simulating the contraction process of the muscle requires dealing with the additional variables involved in these two factors, including:

- The muscle fiber length.
- The number of fibers in each axon of the muscle.
- The recruitment ratio of the motor neuron units.
- The sum affect of the asynchronous activity of the neurons firing.
- The frequency of the stimuli received from the motor neurons.
- Muscle fatigue.
- The particular characteristics and chemical composition of the fibers.

These variables represent a level of detail which is too fine for the issue at hand. Returning to the last item on the previous list, some important considerations can be taken into account regarding the differences between the movements produced by the facial muscles and those produced by muscles in other parts of the body. These are:

- The distances traveled by the contracting endpoints of a facial muscle are very short, usually only a few centimeters.
- The time taken to accomplish a movement is also very short, usually on the order of tenths of a second.
- The load on the face does not vary since there is normally no change in the mass of the face.

For these reasons, modeling the intricacies of the muscle contraction listed above is in some sense a wasted effort. It is conceivable that the degree of error introduced into the system will out-weigh the degree of accuracy achieved. The true determinant of the motion achieved by the muscle is the brain, as can be seen by the number of variables above which involve the motor neurons. In essence, the brain decides how fast and how far to move a muscle and then accomplishes this move with a smooth and continuous motion. The *physical* constraints, therefore, are the only ones left to consider.

6.3 THE INTERPOLATION CURVE

Returning to the mass/spring model, there are two factors left to consider — the end conditions which need to be met by the interpolation function and the effect the function will have on a point being pulled on by more than one muscle simultaneously. The muscle can move to and from any length within its physical limits, and may take any reasonable

amount of time to accomplish the move smoothly. It is the function which describes “smoothly” which is sought.

In order to convincingly simulate the progress of facial expressions, a sinusoid has been chosen to represent the change in the length of the muscle over time during a contraction [31, 35]. The sinusoid has the desired properties for achieving a smooth animation of motion. When used to represent the change in the velocity over time, the cosine function meets the desired end conditions, *i.e.* the muscle is at rest at the start and finish of the movement, and a smooth acceleration occurs in between.

Equation 6.1 gives the relation of the force of the muscle, \mathbf{F} , to the response curve of the muscle, $R(l)$. This response curve is now replaced by the cosine function, given in equation 6.8 below.

$$[6.8] \quad \mathbf{V}KR(l) = \mathbf{a} = \alpha \cos(\beta t)$$

where α and β are constants.

We are interested in the size of the displacement step, dl , for each unit of time. In this case the unit of time, dt , is equal to one frame of animation. The total duration of the movement is defined in number of frames, n . Δl_t is the total displacement of the insertion point of the muscle. To find the displacement of the muscle at each frame:

$$[6.9] \quad \frac{dl}{dt} = \mathbf{v} = \int \mathbf{a} dt = \frac{\alpha}{\beta} \sin(\beta t).$$

To achieve natural looking motion, the velocity of the muscle should be zero at the start and end of each contraction, *i.e.* at $t = 0$ and $t = n$. Solving equation 6.9 for β :

$$[6.10] \quad \sin(\beta t) = 0, \quad t = 0, t = n$$

The sine function is equal to zero at 0 and integer multiples of π , given by the constant C .

$$[6.11] \quad \sin(C\pi) = 0, (C = 0, 1, 2 \dots)$$

Only one oscillation of the sine function is desired, so setting $C = 1$,

$$[6.12] \quad \beta = \pi/n$$

Next, α can be found by taking the integral of equation 6.9 over the total number of frames, n , equaling the total displacement of the insertion point, Δl_t .

$$[6.13] \quad \Delta l_t = \int_0^n \mathbf{v} dt = \frac{\alpha}{\beta^2} \cos(\beta t) \Big|_{t=0}^{t=n}$$

Plugging β in from equation 6.12:

$$[6.14] \quad \Delta l_t = -\frac{\alpha}{(\pi/n)^2} \cos\left(\frac{\pi}{n}t\right) \Big|_{t=0}^{t=n}$$

$$[6.15] \quad \Delta l_t = -\frac{\alpha}{(\pi/n)^2} [\cos(\pi) - \cos(0)]$$

$$[6.16] \quad \Delta l_t = -\frac{2\alpha}{(\pi/n)^2}$$

$$[6.17] \quad \alpha = -\frac{\pi^2 \Delta l_t}{2n^2}$$

The final value, dl for each dt , can now be found:

$$[6.18] \quad dl = \frac{\pi \Delta l_t}{2n} \sin\left(\frac{\pi}{n}t\right) dt$$

Finally, putting the solution for dl back into equation 6.7, the displacement of the muscles for each time step as a function of the response curve, R , is found.

$$[6.19] \quad l_c = l_{\text{old}} + \frac{\pi \Delta l_t}{2n} \sin\left(\frac{\pi}{n}t\right) dt$$

6.4 SIMULTANEOUS MUSCLE ACTION

Finally, combining the effects of different muscles acting on the same point simultaneously is considered. The muscle-vectors displace control points from their current rather than rest positions, so the effect of opposing muscles is to move the points to the positions which are the average of each muscle's displacement, dl . This occurs on a frame by frame basis.

• • •

The end result of the cosine approximation to the actual tension vs. muscle-length response curve is a motion which adequately approximates reality. The animated sequences move smoothly and naturally, with no spurious side effects. A convincing sequence of changes in expression is seen moving across the face of the *mask* model.

CHAPTER 7:

CONCLUSION

Honest labour bears a lovely face...

— *Thomas Dekker*

7.1 INTENTIONS

At the outset of this project, a list of components was made for the design of the *face* system, outlining the functions of the editor and the criteria for the model. The items in this list include the following:

- *face* will be an facial expression editor; a tool which will emulate the emotive changes of the human face using a control structure modeled after the human facial muscular anatomy.
- The editor will use a parametric control system, based on the Facial Action Coding System, and operating on the muscle model.

- The face will be modeled using a spline surface representation in comparison to earlier, polygonal models.
- The editor will incorporate a highly intuitive, interactive user interface using the notations of FACS. This interface should especially simplify working with the FACS notational system.
- The product of the editor will be expression keyframes, to be used in animating by interpolating the facial parameters over time and emulating the motion of muscles contracting and relaxing.
- The effect of the model should be a face which moves realistically, and with subtlety.
- Both the muscle control model and the facial representation model should be highly conformal.
- The muscle model should aim for portability to other facial representations.
- The editor should allow for creative expression on the part of the animator.

7.2 IMPLEMENTATION SUMMARY

face has been implemented using a four-faceted construction. These four areas are the key design elements of the system. They include:

- the muscle control model and its structural components;
- the skin model, implemented using the bicubic B-spline representation;

- the user-interface, incorporating the implementation of the FACS notation representation, and
- the parametric keyframe animation system, animating motion by non-linear interpolation of the facial muscle parameters.

The *face* muscle model is a vector representation based on the the model described by Waters [37] and the human facial anatomy. The vectors are grouped into *Affectors*, similar to the fiber bundles of the striated muscles, then arranged into hierarchical groupings based on the Facial Action Coding System [6]. *face* has empirically translated FACS into a computational model of the elemental facial expression motivators, using the Action Unit as the basis for its structural abstraction.

face has implemented the first facial model which combines a muscle control system with a spline surface representation of the skin. The muscle model operates on the facial topology by moving the B-spline control points in a constrained manner. The spline surface is molded into the *mask* model, which emulates the features of the face — the eyes, nose, and mouth — the cranial structure, and the face's landmark furrows and creases. The *mask* is molded from a rectilinear grid of B-spline control points. This grid can be shaped, using specific rules, into a variety of physiognomies. The muscle/spline combination works to create a general model of the face, conformal to diverse facial topologies.

The editor interface of the *Facial Action Control Editor* uses virtual sliders as an interactive method of inputting changes to the muscle parameters. These sliders, the *zippers*, represent the Action Units of FACS. The function of the *zipper* is made obvious by placing it directly adjacent to the area of the face which it effects, and by updating the parameter changes immediately. In this way, the animator is relieved from the task of learning the FACS notation.

The *zippers* are used to apply a contraction force to the underlying muscles, represented as linear displacement on the slider. The different parameter settings of the muscles are then stored into scripts as keyframes, to be interpolated later.

The function used to interpolate motion between keyframes is a cosine approximation to the muscle contraction curve, a non-linear distribution of tension vs. muscle-length [31, 36]. This function is applied to a simple mass/spring model of the muscles and stepped over time to animate the figure. The interpolation results in a smoothly moving image of the *mask*.

7.3 COMPARATIVE ANALYSIS

The preceding work done on parametric facial models by Parke, Platt, and Waters, and the work done by Ekman on the relationship of facial expression to emotion, has been fundamental to the development of the *face* system. *face* has taken advantage of many of their discoveries and improved on a few of their features.

THE SPLINE SKIN MODEL

face most radically differs from previous work in its application of the parametric muscle model to a spline skin model. Previous parametric models of the face have been implemented based on a polygonal representation of the face's surface, but "the face presents problems which are not handled well by a rigid model" [30]. The B-spline has several intrinsic properties which facilitate modeling the elasticity of the skin, demonstrating the following features when compared to polygonal "nets" [29]:

- the ability to affect an entire curve segment by moving only one point, meaning that changes are propagated easily;

- the ability to change the continuity of the surface by multiplying control points, making it possible to emulate furrows;
- the ability to specify a complex shape, such as the *orbicularis palebrarum*, the sphincter muscle surrounding the eye, with far fewer points than is required in the polygonal model;
- the ease with which the surface can be manipulated, allowing it to conform easily to different facial structures, and
- the smoothness of the surface achieved using a spline.

Komatsu [17], Nahas [22], and Forsey [9] have also been exploring the use of parametric curves for modeling animated figures, also with reported success:

...the use of B-spline surfaces permits more supple movements of the figure...the intrinsic property of mutual tension at the control points gives an elasticity to the muscles and skin of a figure in motion.[22].

As stated in the list above, the bicubic B-spline model is capable of emulating the furrows of the face, an area in which Waters' model failed. Waters' *zone of influence* model will also produce undesirable results when a node point is displaced beyond the boundary of the zone [27]. The basis functions of the B-spline serve the same purpose in the *face* model as Waters' *zones of influence*, but do not break down under unusual or extreme conditions.

FACS

Platt, Waters, and others have used the FACS notation to encode the actions performable on the face, but their work has centered on representing and describing motion of the surface of the skin; a *complete* control structure for facial expressions using FACS has not been reported to date. *face* has implemented a control structure using a hierarchy of

abstractions to represent the organization of Action Units, muscles, and B-spline control points. This implementation of FACS allows for easy definition and control of facial expressions through the *zipper* interface, despite the complexity of the FACS notation.

face has interpreted the qualitative descriptions of FACS into a computational model, combining the notation with actions based on the muscular anatomy of the face. *face*'s success as an implementation of the FACS Action Units is demonstrated by its ability to match *mask* expressions to Ekman's examples [6] (section 7.4).

The foremost feature of FACS is its generality; it is applicable to divergent facial types. By successfully basing its structure on FACS, *face* shares in this feature of generality.

GENERALITY OF THE MODEL

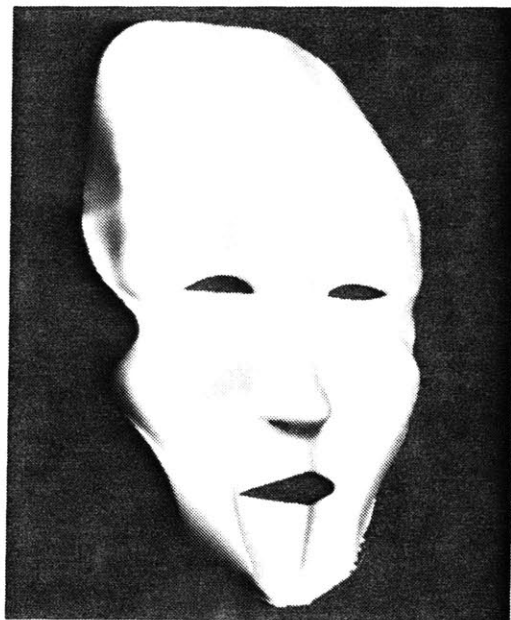
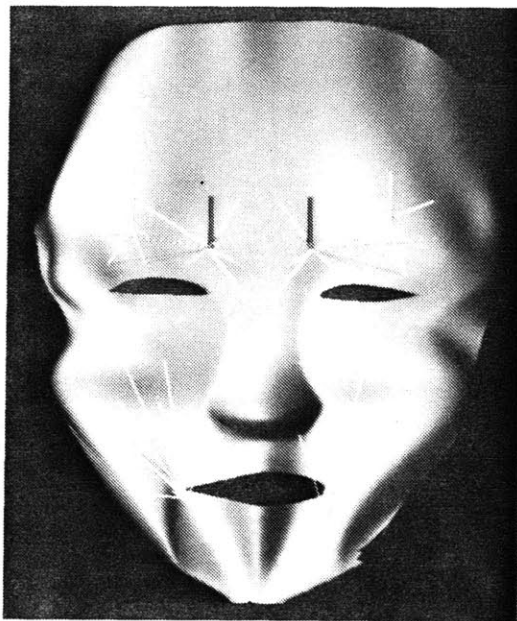
It has already been stated that the B-spline skin representation is easily adapted to different facial types, like the FACS notation. Equally general is the vector muscle representation. This model succeeds because of its independence from any underlying bone structure or specific model, as compared to Parke's [24] and Platt's [30] implementations. The vector representation is portable to a variety of facial models [37].

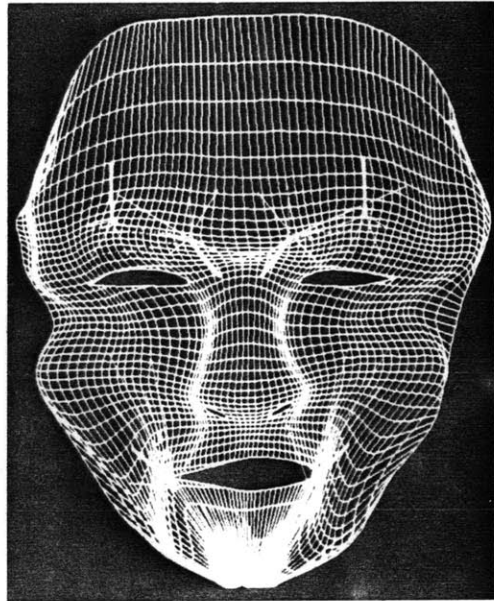
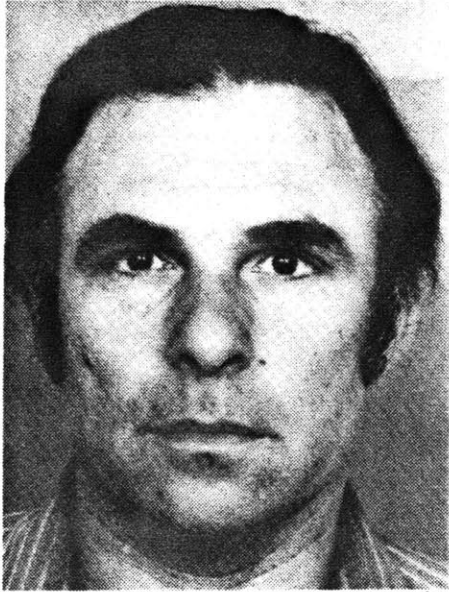
Using a control model based on the muscular structure of the face has additional advantages. The individual expressions are not explicitly encoded — it is the user who specifies the expressions via muscle movements — therefore adding additional expressions does not require modifying the parameter set. Also, unnatural configurations are not encountered when interpolating the face between keyframes. Parke's premise that the cosine function adequately approximates the motion of the facial muscles [25] is successfully demonstrated by Platt, Waters, and now the *face* model.

7.4 RESULTS

Below are pictures generated by *face*. The *mask* model is placed side by side with the AU examples from the FACS Manual [6] to aid comparison of corresponding Action Unit activations. The four implemented AU's are shown in independent and combined activations.

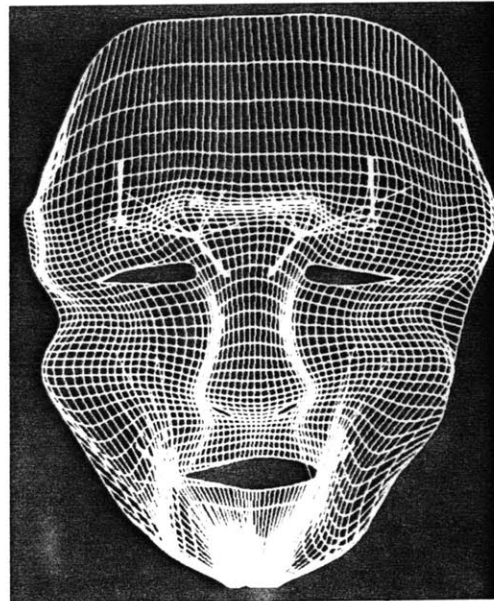
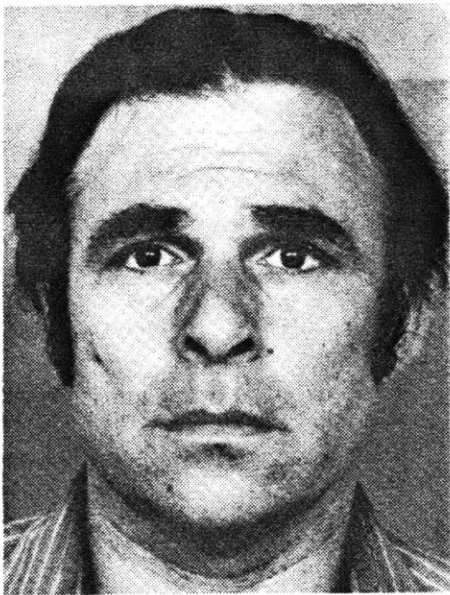
COMPARISON PICTURES





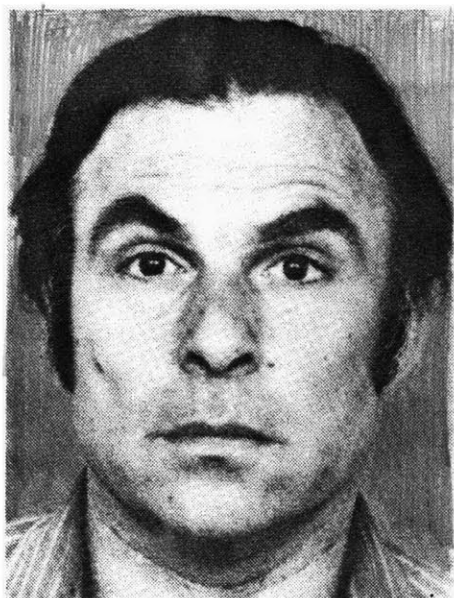
0

(7.1) AU0 - NEUTRAL FACE

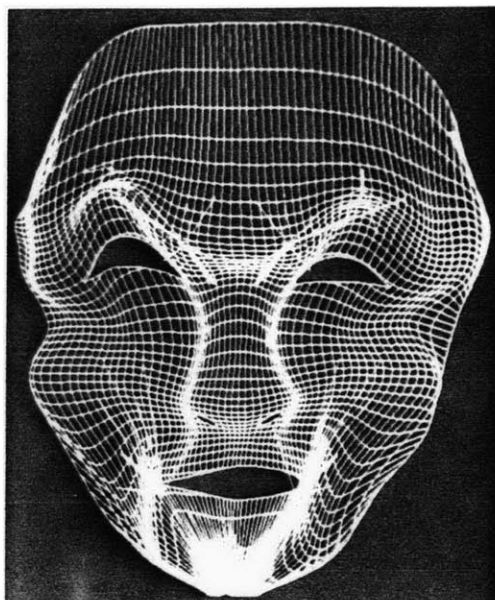


1

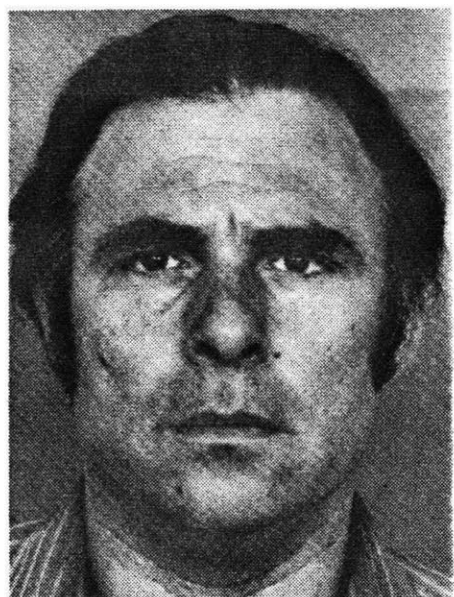
(7.2) AU1



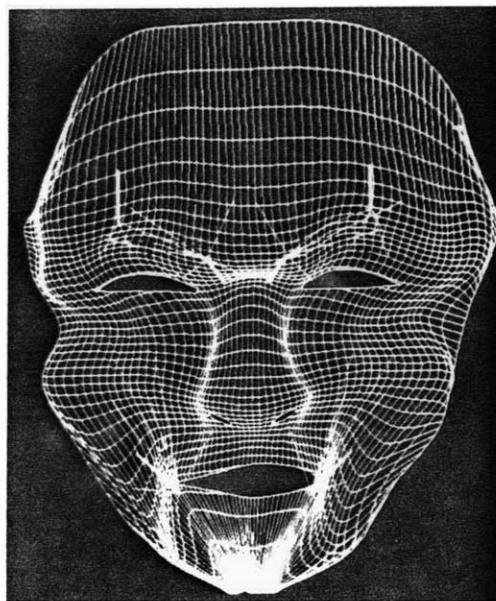
2



(7.3) AU2

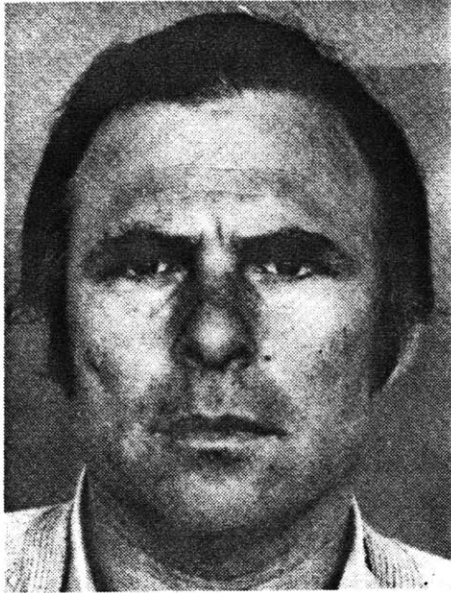


4a

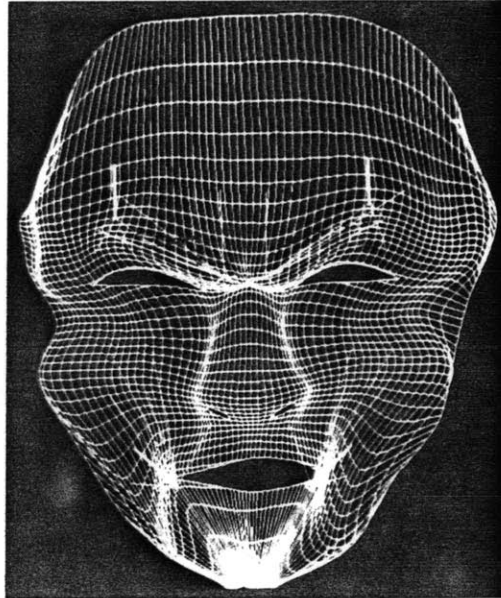


(7.4) AU4 - PARTIAL ACTIVATION

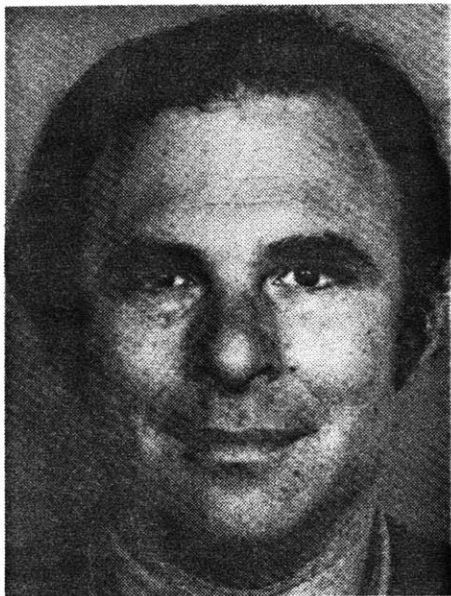
2



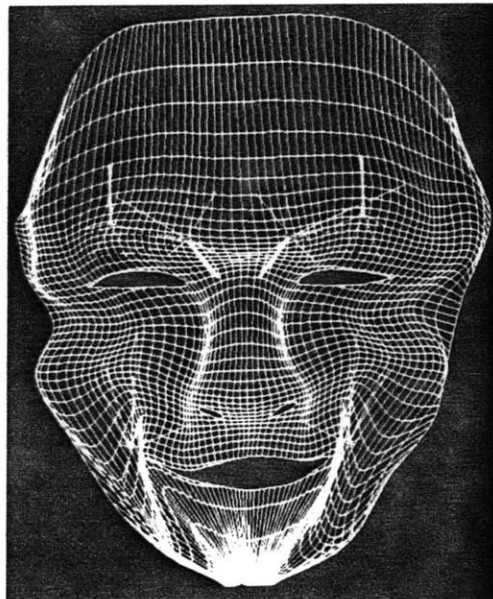
4b



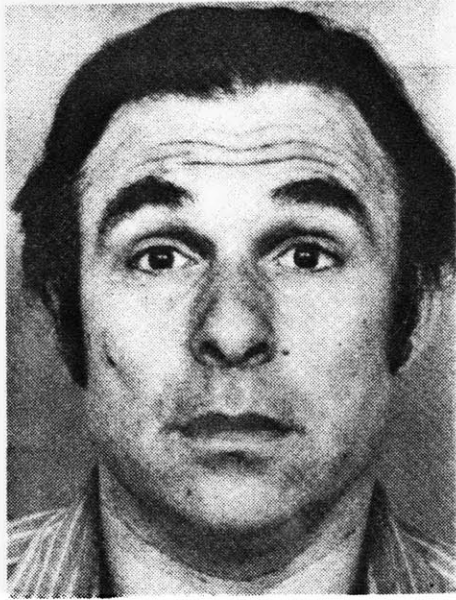
(7.5) AU4 - MAXIMAL ACTIVATION



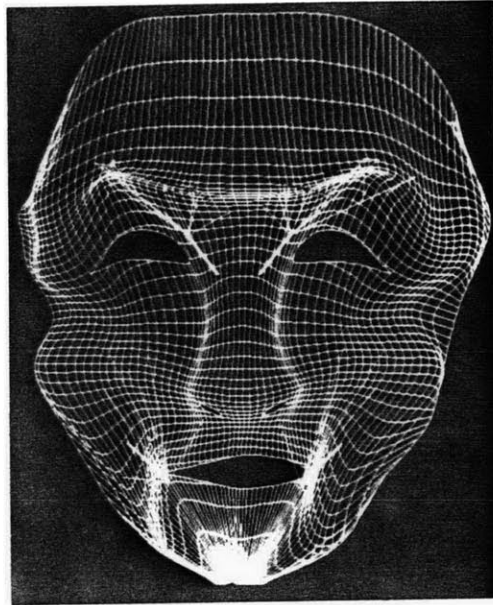
12X



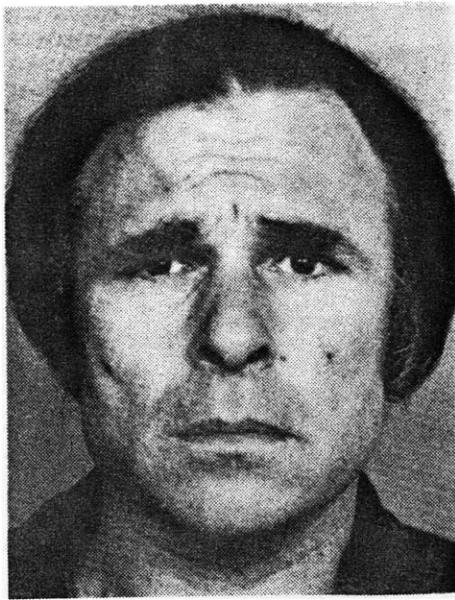
(7.6) AU12 - MAXIMAL ACTIVATION



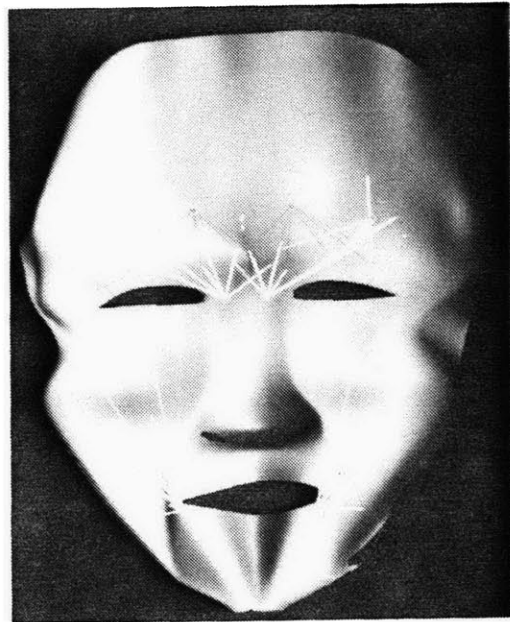
1+2



(7.7) AU1 + AU2

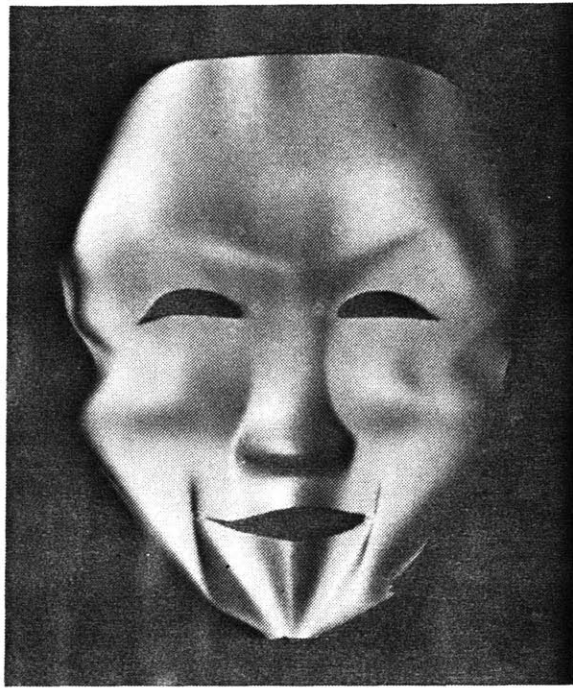


1+4b

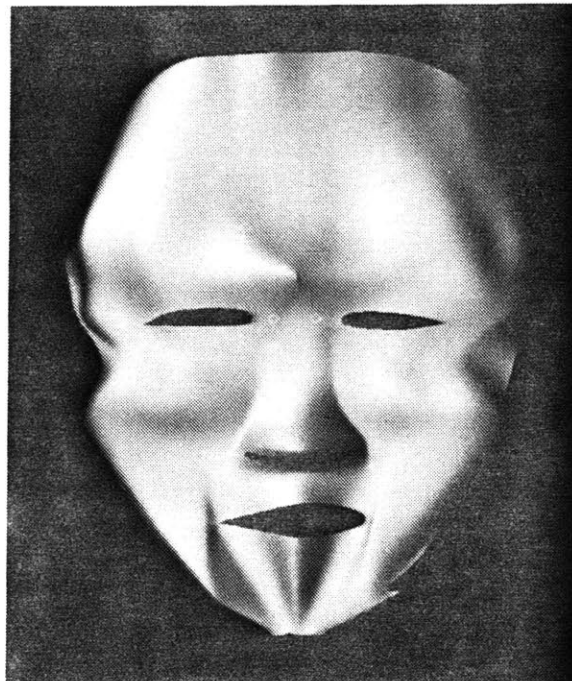


(7.8) AU1 + AU4 - MAXIMAL ACTIVATION

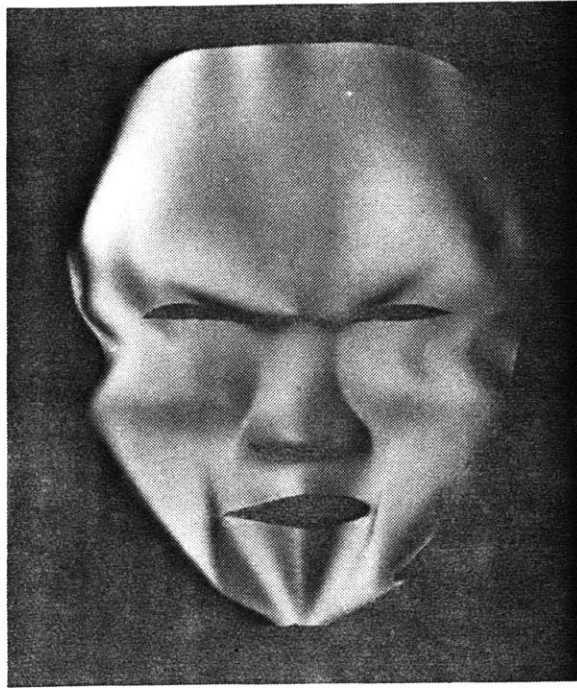
OTHER COMBINATIONS



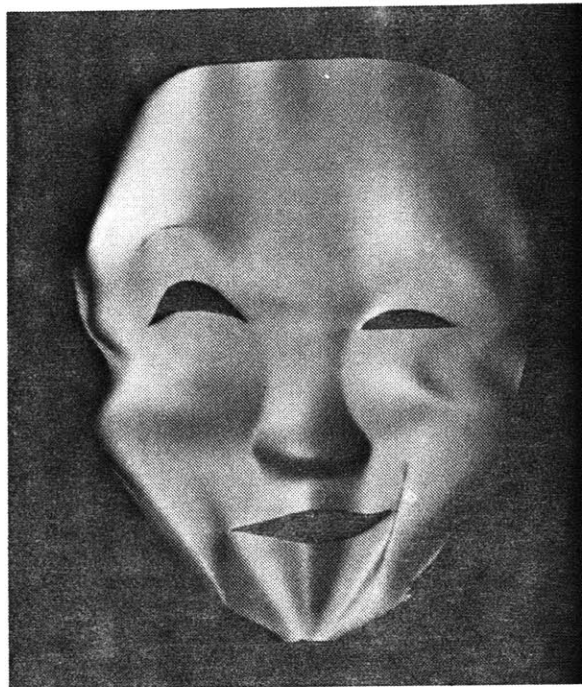
(7.9)



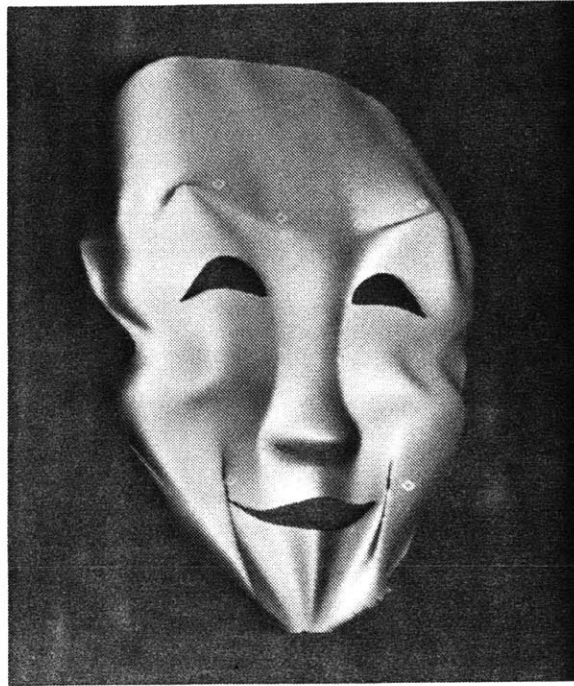
(7.10)



(7.11)



(7.12)



(7.13)

7.5 FUTURE WORK

The most important issue yet to be addressed in facial representations is the modeling of the soft tissue flowing over bone [14, 37]. The vector muscle implementation of *face* suggests some methods for emulating this effect.

SURFACE TANGENT VECTOR METHOD

The bicubic B-spline skin model can be fit over any cranial bone structure or be shaped to affect an “apparent” skull. Whether or not an underlying bone model is present, the vector muscles can be made to follow the facial surface, dragging the control points along with them to give the illusion of skin “flowing” over the surface. It is most important that the skin be prevented from passing through the virtual underlying skull. The current implementation of *face* approximates this flowing surface by gross piecewise linear

approximations to the curvature of the facial surface, preventing the skin from passing through itself. By looking specifically at the surface tangents at very small intervals along the dominant muscle direction, a much finer fit to the surface could be accomplished. Using these tangent vectors, the muscles can be made to flow directly along the surface of the skin and bone. This implementation would have the added benefit of completely generalizing the muscle model, since each muscle would automatically conform to any shape of the face.

PIEPER'S SKIN MODEL

A better way to model the effect of soft tissue flowing over bone is by attaching the muscles to a model which more precisely simulates the skin. *face* is being developed in parallel with a physically-based model of the facial tissue, capable of simulating realistic deformations to the tissue due to interactions with gravity, muscle forces, and the underlying hard tissue. This model, under development by Pieper, uses a "dynamic constraint network" [28] to represent the properties of the different layers of material present in the flesh of the face and forward dynamic simulation to calculate the changes to the network configuration.

Pieper's model handles the effects of muscles acting on the skin in a natural fashion, propagating a displacement through the flesh in a way similar to that accomplished by the basis functions of the bicubic B-spline. By combining the FACS based vector muscle model with this skin model, the most realistic model of the human face to date should be created. Flesh flowing over bone will be only one of the benefits of this combination. Pieper's skin model also effectively simulates the elastic properties of the skin, so that bulges, buckles, wrinkles, and puffing are handled correctly, effects which have been missing from all previous models.

REFINEMENT

Another issue of research in facial modeling is the *wrinkling* of the skin caused by its motion. Pieper's model presents one solution to this problem. Another method for creating wrinkles in the current spline skin model is suggested by Forsey and Bartels [9]. They have described a spline curve editing technique called *refinement* [1] which allows the overlaying of increasingly fine levels of detail onto the spline by replacing a basis function with an equivalent linear combination of new basis functions.

Refinement is usually advocated as a means of gaining finer control over a spline curve or surface during editing.... It permits the change of control vertices and subsequent editing of fine detail in one region of the curve while leaving control vertices in other regions unaffected.[9].

The refinement process can be used to make small, localized changes to the continuity of the surface, applying fine details such as wrinkles or other surface anomalies to a small area of the model without negatively affecting the functioning of the model as a whole. An alternative to this method would be to design a *mask* with a high spatial frequency distribution of control points so that "extra" points are left over for creating non-functional details on the face, but this is a more cumbersome solution.

ACCURATE CRANIAL MEASUREMENTS

Improvements planned to the *face* model involve implementing a constraint structure which will force the mask into a more accurate model of the head based on cranial measurements. The skin can be fit over an anatomically correct model of the skull, or measurement and proportion constraints [38] can be included in the *spline-molder* program.

MORE ACTION UNITS

An immediate addition planned to *face* is more Action Unit data files. The AU's listed in Chapter Three are all possible with the current model and need to be implemented. Especially interesting will be jaw rotation, which it is predicted will work correctly with this model. Jaw rotation is one of the functions cited as difficult by Parke and Platt. The work done by Nahas *et.al.* using a quadratic B-spline model of the face supports the assumption that jaw rotations will work correctly with the spline model of *face*, despite the absence of an underlying bone model:

...the difference between muscles and skeleton is not important: both kinds of actions could be reproduced by a movement of control points.[22].

Adding an *ActionUnit* editor to the *face* system would facilitate this task of AU construction.

7.6 SHORTCOMINGS OF THE *FACE* MODEL

The present implementation of *face* contains certain flaws, some of which are the effect of the graphics library used in this implementation, and others which are more design related.

STARBASE

As was mentioned in Chapter Three, the *Starbase* graphics library spline generating functions imposed certain constraints on how far the model could be developed using this code. The most glaring shortcoming of the *face* model is the missing features of the face, the lack of lips, eyelids, eye globes, teeth, and facial hair. The *mask* model would be greatly improved with these additions.

The *Starbase* implementation of the B-spline surface generating software has two problems as far as the *face* model is concerned. The first of these is an apparent “bug” in the knot-vector function. The knot-vectors only affect the endpoints of the curves instead of all the knots along the curve when so specified. This makes it impossible to reliably tie surface patches together with no gaps occurring between the adjoining patches. The lips and eyelids need to be represented as distinct patches because of their complexity. Both areas are controlled by sphincter muscles which require more control points than linear/parallel muscles in order to create the “draw-string bag” effect. The best fix for this problem is to rewrite the B-spline software so that the knot-vectors function properly.

The remaining features of the face, the eye globes, teeth, and facial hair, are missing because of a more curious aspect of the *Starbase* software. The positionings of the control points for the surface are specified and fed to the spline-generating routine, which then interpolates the surface. The location of the surface in space, for instance the coordinates of the polygon vertices used by the hardware *Phong* shader, is known to the software, but absolutely inaccessible to the programmer. The location of points actually on the surface must be known to properly add the remaining features to the face, primarily to perform collision-detection calculations between these features and the skin when the surface is moved. The eye globes and teeth are best represented by separate polygonal objects [24], and the hair by particle system strands [35]. The lips cannot be allowed to pass through the teeth nor the lids through the eyes, and eyebrow hairs which do not move along with the surface would look extremely odd. Software has been written to overcome this oversight in the *Starbase* implementation [11], but has not yet been added to the *face* package. Texture mapping onto the face surface will also be facilitated by this software addition.

MISSING AU'S

The final criticism of the *face* model involves the “missing AU’s”. Some of the Action Units denoted by Ekman [6] are not possible within *face*’s design. These are the AU’s which involve the following aspects of the facial anatomy:

- changes in air pressure inside the mouth, resulting in “puffing” and “sucking” of the cheeks;
- jaw pressure on the teeth, as in the *Bite* and *Jaw Clench* Action Units;
- tongue motion; and
- the muscles of the neck.

The necessary details in the model of the head and neck needed to perform these functions are not included in the *mask* model. Implementation of the Action Units involving these areas of the anatomy would require a substantially more complex model of the face and head than the present bicubic B-spline model — far beyond the scope of this research.

7.7 CONCLUDING REMARKS

Overall, the *Facial Action Control Editor* has proven to be a successful enterprise. *face* has succeeded in emulating human facial expressions, producing results which can be compared to human models. The spline surface has been found to model the facial properties quite well and to produce a convincing image. The full model is conformal to various physiognomies, and we hope that the muscle model will adapt well to other surface implementations. The system as a whole is easy to use and allows the animator extensive creative control over the details of the facial expressions. Most importantly, the muscle

model permits the desired subtlety to come through to the figure, creating a countenance with enormous expressive potential.

Formosa facies muta commendatio est.

“A beautiful face is a mute recommendation.”

— *Publius Syrus (Tr. Bacon)*

APPENDIX A:

STRUCTURES AND DATA FILES

The following is a listing of the data structure formats used in the *Facial Action Control Editor* subroutines and the ascii data files of the *ActionUnits* and *Muscles* implemented to date. The *face* subroutines have been programmed in *C* and use the Hewlett-Packard *Starbase* graphics library subroutines for the elementary graphics functions.

A.1 STRUCTURES

The structures listed below can be found in the header files "face.h" and "face-defs.h".

GENERAL GRAPHICS

```
typedef
    struct{ float x, y, z; } Point;

typedef
    struct{ float x, y; } ScreenPoint;
```

```

typedef int      Fildes;          /* file descriptor */
typedef float    VDC_Point[2];    /* Starbase format for screen coordinates */
typedef float    SB_point[3];     /* starbase 3d point format */

```

```

typedef
    struct{
        float    x, y, z;
    }
Vector;

```

```

typedef
    struct{
        float    r, g, b;
    }
Color;

```

SPLINE PATCH STRUCTURES

```

typedef int      Flag;
typedef int      Order;
typedef int      Index;

```

```

typedef
    struct{
        Vector    pos;          /* position vector */
        Flag      shared;      /* TRUE or FALSE is this a shared control
                                point */
        Cntrl_index tie;       /* index of point to which this point is tied */
    }
ControlPt;

```

```

typedef
    struct{
        float      u, v;
    }
Pt_uv;          /* B-spline trimming curve point */

```

```

typedef
    struct
        Index      sp_index;   /* patch number - to access an array of
                                spline patches */
        Index      spu, spv;   /* control point index */
    }
Ctrl_index;

```

```

typedef
    struct{
        Order      edge_order; /* LINEAR, QUADRATIC, or CUBIC */
        Pt_uv      edge[5];
        float      edge_knots[9];
        Order      hole_order; /* LINEAR, QUADRATIC, or CUBIC */
        Pt_uv      hole[TRIM_NUM];
    }

```

```

        float      hole_knots[TRIM_NUM + 4];
    }
Hole;                                /* trimming curve*/

typedef
    struct{
        char      name[20];
        int       u, v;                /* number of points in UV space */
        Order     u_order, v_order;    /* LINEAR, QUADRATIC, or CUBIC */
        ControlPt CtrlPts[16][16];    /* list of control points */
        ScreenPt  screen[16][16];     /* vdc coordinates of the control points */
        int       uk, vk;              /* number of knot vector */
        float     Uknots[32];         /* u knot vector */
        float     Vknots[32];         /* v knot vector */
        Flag      hole;                /* True or False */
        int       num_holes;
        Hole      h[5];                /* trimming curve structure */
    }
SplinePatch;

```

ACTION UNIT STRUCTURES

```

typedef
    struct{
        Vector     dir;                /* unit vector in direction of pull */
        float      mobility;           /* muscle's mobility. scale factor -> 0-1 */
        int        numpts;             /* index for pts */
        Ctrl_index pts[10];            /* list of control points attached to this
                                         muscle */
    }
Muscle;

typedef
    struct{
        float      lmin, lmax;         /* zipper length limits for affecting this
                                         Affector*/
        int        num_mus;           /* number of muscles on this affector */
        Muscle     mus[9];            /* list of muscles on this affector, left and
                                         right */
    }
Affector;

typedef
    struct{
        char       text[20];
        VDC_Point  loc;
    }
ScreenLabel;

```

```

typedef
    struct{
        Affector    aff[5];
        Vector      origin;          /* static point of "muscle" (action unit /
                                       zipper) */
        Vector      zipper;          /* display vector for sliding "force" along*/
        Vector      zipvdc;          /* screen (VDC) coordinates of zipper */
        Vector      zipdir;          /* unit direction vector of zipper */
        float       zip_len;          /* current length of "zipper" */
        ScreenLabel label;
    }
Zipper;          /* replaced "OneSide" */

```

```

typedef
    struct{
        int         name;
        Color       zipcolor;
        float       zipmax;          /* length limiters */
        float       dmag;           /* magnitude of current zipper
                                       displacement */
        float       displacement;    /* magnitude of muscle displacement */
        Flag        symmetry;        /* ON, OFF, MEDIAL */
        int         num_aff;         /* number of effectors (equal for left and
                                       right */
        Zipper      side[2];        /* list of all the things that are symmetrical
                                       in action units */
    }
ActionUnit;

```

```

typedef
    struct{
        int         au_num, side;
    }
AU_index;

```

SCRIPTS

```

typedef
    struct move_thing {
        char       au_name[20];
        int        side;
        float       xstart;          /* start zipper length */
        float       displacement;
        struct move_thing *next_move;
    }
Move;          /* which action unit to move &how much */

```

```

typedef
    struct scene_thing {
        int      num_moves;
        Move     *move;           /* list of moves */
        int      num_frames;
        struct scene_thing *next_scene;
    }
Scene;                          /* list of simultaneous moves in one time frame */

typedef
    struct{
        int      num_scenes;
        Scene    *scene;
    }
Script;                          /* list of scenes */

```

MENU FUNCTION DEFINITIONS

```

typedef int  (*function_ptr) ();
typedef char string_thing[81];
typedef char *function_string;
typedef char *message_string;

typedef
    struct{
        function_string  name;           /* any function call may have two names */
        function_ptr     function;     /* routine to call by 'name' */
    }
Function;

typedef
    struct{
        int      count;                 /* number of menu array slots filled */
        Function function_list[100];
    }
Menu;

```

I/O

```

typedef char Name[32];

typedef
    struct{
        Name    patch;                 /* patch or mask name */
        Name    au[20];                 /* list of ".au" files */
    }
FileNames;

```

CAMERA

```
typedef
    struct{
        camera_arg  Camera;      /* starbase data type */
        float       rho, theta, phi; /* spherical coordinates */
    }
Scamera;
```

A.2 DATA FILES

Next are the file storage formats for the *ActionUnit*, *Muscle*, *Name*, and *Script* data files, followed by a listing of all the implemented AU's and their muscles.

A.2.1 FILE FORMATS

The file format illustrations below give the order in which the different variables are listed in the ascii format files. The variable types — floating point, integer, or char — are also indicated next to the variable descriptions. The number of variables required on each line of the file is indicated by the “_” markers. Four types of files are listed: *ActionUnit* files, *Muscle* files, *Script* files, and *Name* files. *ActionUnit* files are indicated by the suffix “.au”, *Muscle* files by “.mus”, *Script* files by “.scr”, and *Name* files by “.file”.

ACTION UNIT

<i>name</i>	_	<the AU ID number> (integer)
<i>r, g, b</i>	---	<zipper color> (float)
<i>symmetry</i>	_	<symmetry> (integer- OFF = 0, ON = 1, MEDIAL = 2)
<i>zipmax</i>	_	<maximum length for zipper> (float)
<i>num_aff</i>	_	<number of affectors> (integer)
		<i>left side</i>
<i>origin</i>	---	<origin vector> (3 floats)
<i>zipper</i>	---	<zipper vector> (3 floats)
<i>label.text</i>	_	<label text> (char)

loc.x loc.y -- <label screen coordinates x, y> (float- VDC coordinates)

affectors

lmin , lmax -- <length limiters on this affector> (float)

num_mus - <number of muscles> (integer)

mus_file - <name of muscle files to read in> (char)

right side

origin --- <origin vector> (3 floats)

zipper --- <zipper vector> (3 floats)

label.text - <label text> (char)

loc.x loc.y -- <label screen coordinates x, y> (float- VDC coordinates)

affectors

lmin , lmax -- <length limiters on this affector> (float)

num_mus - <number of muscles> (integer)

mus_file - <name of muscle files to read in> (char)

MUSCLE

dir.x dir.y dir.z --- <x,y,z unit direction vector> (float)

mobility - <mobility> (float)

numpts - <number of control pts attached to this muscle> (integer)

patch.u patch.v -- <v,u pairs of patch control points>(integer)

(list in row, column order)

FILE NAMES

patch.spl - <name of Spline Patch file> (char)

num_au - <number of Action Unit files> (int)

*au*_*.au* - (_ _...)<names of Action Unit files> (char)

SCRIPTS

num_scenes - <number of scenes> (int)

first scene

<i>num_frames</i>	-	<number of frames> (int)
<i>num_moves</i>	-	<number of moves in this scene> (int)
<u><i>first move</i></u>		
<i>side</i>	-	<side of face> (int)
<i>AU*</i>	-	<AU unique name> (char[20])
<i>xstart</i>	-	<start length (before moving)> (float)
<i>displacement</i>	-	<total displacement> (float)
<u><i>next move</i></u>		
same as first move, etc.		
<u><i>next scene</i></u>		
same as first scene, etc.		

A.2.2 ACTION UNIT FILES WITH ASSOCIATED MUSCLE FILES

```

-----
                        AU 1
-----

:~::~:
au1.au
:~::~:
1
0.8  0.15  0.15
0.2
2

0.1534  -0.0986  0.3023
0.1534  -0.0982  0.1023
AU1L
    0.0    0.1

0.0  0.138
2
data/AU1L_0.mus
data/AU1L_1.mus
0.138 0.2
2
data/AU1L_1.mus
data/AU1L_2.mus

-0.1534  -0.0986  0.3023

```

* Fill in the Action Unit number

-0.1534 -0.0982 0.1023
AU1R
0.0 0.1

0.0 0.138
2
data/AU1R_0.mus
data/AU1R_1.mus
0.138 0.2
2
data/AU1R_1.mus
data/AU1R_2.mus

-0.03 -0.5 3.0
1.0
4
3 9
3 8
4 9
4 8

.....
AUIL_0.mus

.....
-0.03 -0.5 3.0
1.0
4
3 9
3 8
4 9
4 8

.....
AUIL_1.mus

.....
-3.5 0.0 0.5
0.25
4
3 11
3 10
4 11
4 10

.....
AUIL_2.mus

.....
-0.03 0.9 3.0
1.0
4
3 9
3 8
4 9
4 8

.....
AU1R.mus
.....
0.03 -0.5 3.0
1.0
4
3 7
3 6
4 7
4 6

.....
AU1R_0.mus
.....
0.03 -0.5 3.0
1.0
4
3 7
3 6
4 7
4 6

.....
AU1R_1.mus
.....
3.5 0.0 0.5
0.25
4
3 5
3 4
4 5
4 4

.....
AU1R_2.mus
.....
0.03 0.9 3.0
1.0
4
3 7
3 6
4 7
4 6

AU 2

.....
au2.au
.....
1
0.2 1.0 0.5
0.258
1
0.4 -0.116 0.41

0.4 -0.12 0.18

AU2L

0.3 -0.12

0.0 0.258

1

AU2L.mus

-0.4 -0.116 0.41

-0.4 -0.12 0.18

AU2R

-0.3 -0.12

0.0 0.258

1

AU2R.mus

.....

AU2L.mus

.....

0.0 0.008 0.9908

1.0

4

3 11

3 10

4 11

4 10

.....

AU2R.mus

.....

0.0 0.008 0.9908

1.0

4

3 5

3 4

4 5

4 4

AU 4

.....

au4.au

.....

1

1.0 1.0 0.0

0.185

3

0.066 -0.1045 -0.0548

0.1634 -0.0982 0.1023

AU4L

0.0 0.1

0.0 0.185

```

2
data/AU4L_0.mus
data/AU4L_1.mus
0.037  0.185
2
data/AU4L_2.mus
data/AU4L_3.mus
0.0925  0.185
1
data/AU4L_4.mus

```

```

-0.066  -.1045  -0.0548
-0.1634 -0.0982  0.1023
AU4R
      0.0  0.2

```

```

0.0  0.185
2
data/AU4R_0.mus
data/AU4R_1.mus
0.037  0.185
2
data/AU4R_2.mus
data/AU4R_3.mus
0.0925  0.185
1
data/AU4R_4.mus

```

```

.....:
AU4L_0.mus
.....:
-1.5 -0.9 -1.75
1.0
4
4 8
3 8
4 9
3 9

```

```

.....:
AU4L_1.mus
.....:
-1.55 -0.1 -2.5
0.4
1
3 10
.....:
AU4L_2.mus
.....:
-4.0 -0.1 -0.5
0.8
1
3 10

```

```
.....  
AU4L_3.mus  
.....  
-2.9 -0.5 -1.0  
0.5  
1  
2 8  
.....  
AU4L_4.mus  
.....  
-1.5 0.0 -1.75  
0.8  
2  
3 11  
2 11
```

```
.....  
AU4R_0.mus  
.....  
1.5 -0.9 -1.75  
1.0  
4  
4 7  
3 7  
4 6  
3 6
```

```
.....  
AU4R_1.mus  
.....  
1.55 -0.1 -2.5  
0.4  
1  
3 5
```

```
.....  
AU4R_2.mus  
.....  
4.0 -0.1 -0.5  
0.8  
1  
3 5
```

```
.....  
AU4R_3.mus  
.....  
2.9 -0.5 -1.0  
0.5  
1  
2 7
```

```
.....  
AU4R_4.mus  
.....  
1.5  0.0  -1.75  
0.8  
2  
3  4  
2  4
```

AU 12

```
.....  
au12.au  
.....  
1  
0.21  0.92  0.7  
0.2  
2  
  
0.3545  -0.0903  -0.7252  
0.2229  -0.1082  -0.8747  
AU12L  
0.0  0.1  
  
0.0  0.2  
2  
data/AU12L_0.mus  
data/AU12L_1.mus  
0.12  0.2  
2  
data/AU12L_2.mus  
data/AU12L_3.mus  
  
-0.3219  -0.0966  -0.7336  
-0.1903  -0.1145  -0.8831  
AU12R  
0.0  0.2  
  
0.0  0.2  
2  
data/AU12R_0.mus  
data/AU12R_1.mus  
0.12  0.2  
2  
data/AU12R_2.mus  
data/AU12R_3.mus
```

.....
AU12L_0.mus
.....
2.2 0.3 2.5
1.0
2
10 11
9 11

.....
AU12L_1.mus
.....
1.0 -0.8 2.0
0.4
4
8 11
7 11
7 12
6 11

.....
AU12L_2.mus
.....
-2.0 0.0 0.3
0.7
1
8 10

.....
AU12L_3.mus
.....
-1.0 -0.8 1.5
0.4
2
8 13
6 12

.....
AU12R_0.mus
.....
-2.2 0.3 2.5
1.0
2
10 4
9 4

.....
AU12R_1.mus
.....
-1.0 -0.8 2.0
0.4
4
8 4
7 4

7 3
6 4

.....

AU12R_2.mus

.....

2.0 0.0 0.3

0.7

1

8 6

.....

AU12R_3.mus

.....

1.0 -0.8 1.5

0.4

2

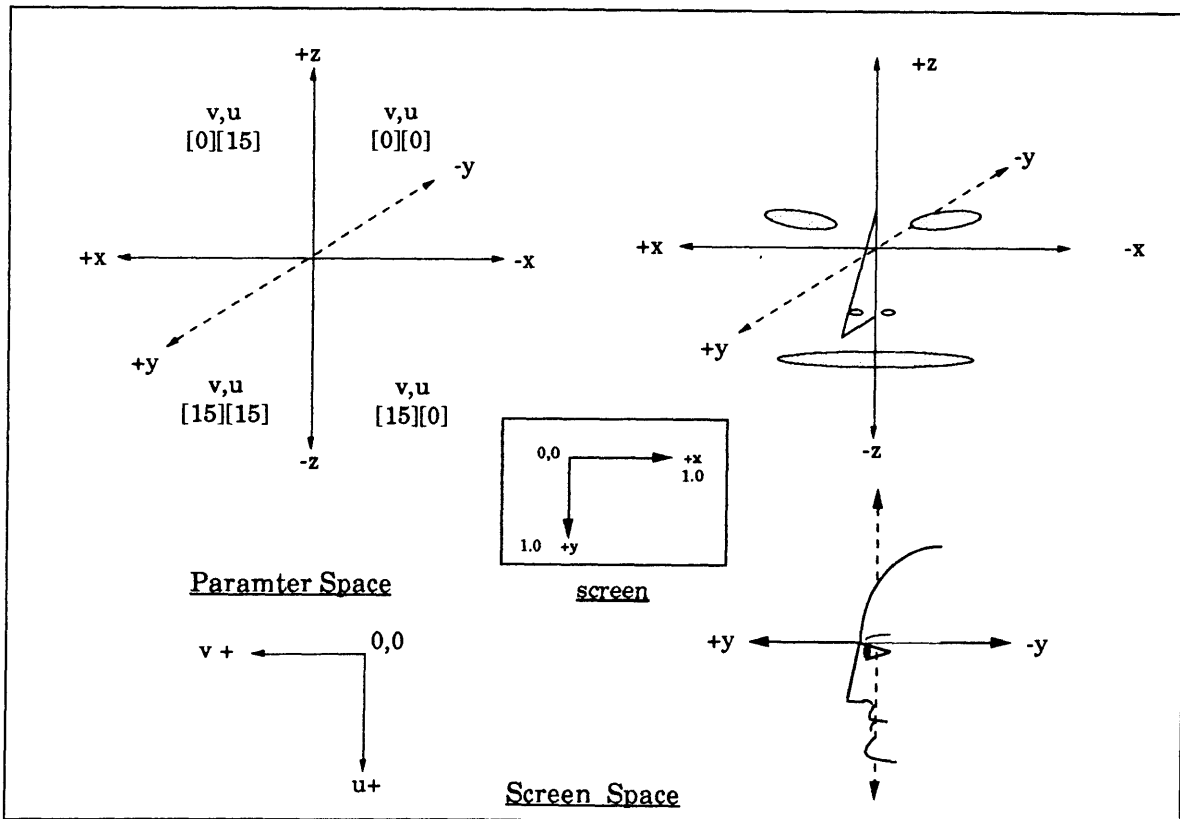
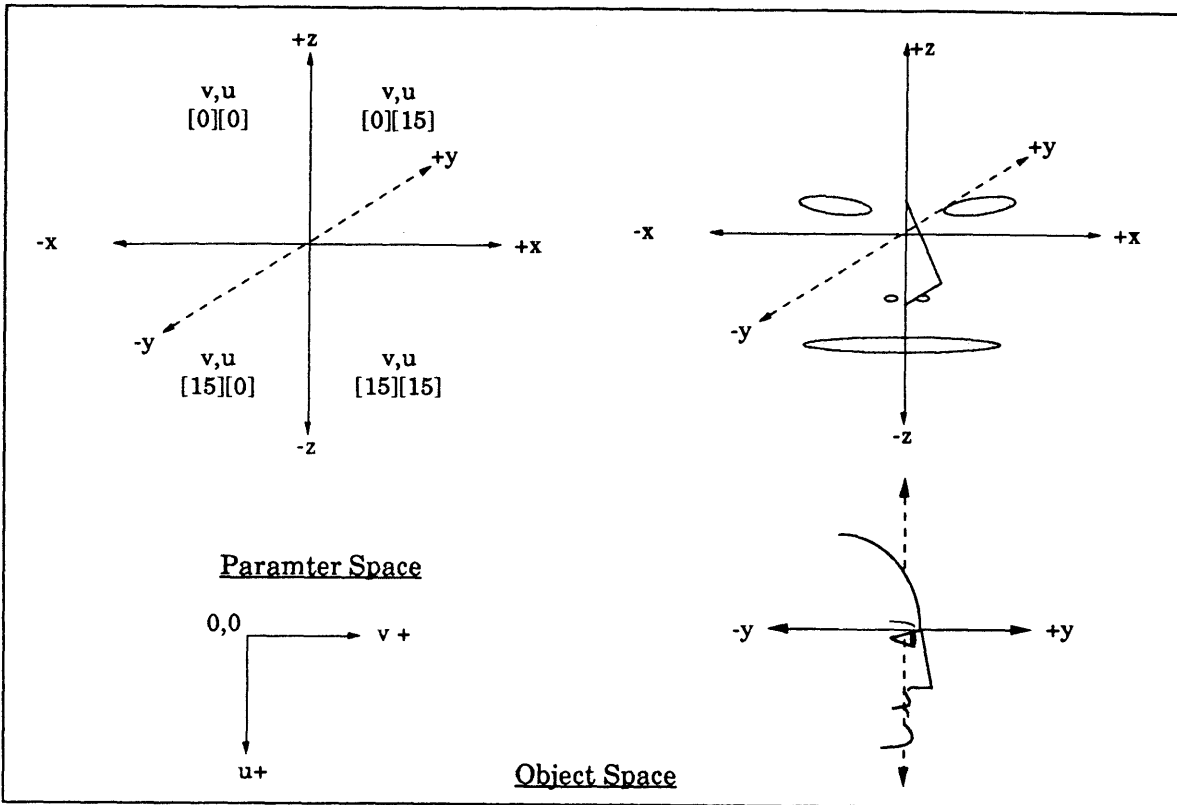
8 2

6 3

APPENDIX B:

SPACE TRANSFORMATIONS

Below are two diagrams which show the three dimensional vector and the parameter array orientations, first in *object space*, and then in *screen space*. Object space is dealt with when creating data files, such as trimming curves and *Muscles*. When viewed on the screen, these object coordinates undergo a series of transformations [33] before being rendered onto the screen. The coordinates are in essence “flipped” and can be difficult to keep straight while switching back and forth between viewing and editing files. The diagrams below are helpful for avoiding confusion.



BIBLIOGRAPHY AND REFERENCES

- 1) Bartels, Richard, John Beatty, and Brian Barsky; *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Inc. Los Altos, CA. Copyright 1987.
- 2) Boucher, Jerry D., and Paul Ekman; "Facial Areas and Emotional Information" *Journal of Communication*, 1975, Volume 25, No. 2, pp. 21-29.
- 3) Chen, M., S.J. Mountford, and A. Sellen; "A Study in Interactive 3-D Rotation Using 2-D Control Devices" *Computer Graphics*, 1988, Volume 22, No. 4, pp. 121-129.
- 4) Cutting, C., F.L. Bookstein, B. Grayson, L. Fellingham, and J.G. McCarthy; "Three-Dimensional Computer-Assisted Design of Craniofacial Surgical Procedures: Optimization and Interaction with Cephalometric and CT-Based Models" *Plastic and Reconstructive Surgery*, 1986, Volume 77, No. 6, pp. 877-885.
- 5) Ekman, Paul, and Harriet Oster; "Facial Expressions of Emotion" *Annual Review of Psychology*, 1979.
- 6) Ekman, P., and W. Friesen; *Facial Action Coding System*. Consulting Psychologists Press. Palo Alto, CA. Copyright 1978.
- 7) Finch, Christopher; *The Art of Walt Disney*. Abradale Press, Harry N. Abrams Inc. NY. Copyright 1973.
- 8) Foley, J. D., and A. Van Dam; *Fundamentals of Interactive Computer Graphics*. Addison-Wesley. Copyright 1982.
- 9) Forsey, David R., and Richard Bartels; "Hierarchical B-Spline Refinement" *Computer Graphics*, 1988, Volume 22, No. 4, pp. 205-212.
- 10) Gray, Henry; *Anatomy, Descriptive and Surgical*. Bounty Books, Crown Publishers Inc. NY. Copyright 1977.
- 11) Halle, Michael; Personal Communication. 1988.
- 12) Herbison-Evans, Don; "Dance, Video, Notation, and Computers" *Leonardo*, 1988, Volume 21, No. 1, pp. 45-50.
- 13) Hewlett-Packard; *Starbase Graphics Techniques HP-UX Concepts and Tutorials*,. Volume 2 HP 900 Series 300 Computers. Hewlett-Packard Company. Fort Collins, CO. Copyright 1986.
- 14) Hill, D.R., A. Pearce, and B. Wyvill; "Animating Speech: An Automated Approach Using Speech Synthesis By Rules" *Visual Computer*, 1988, Volume 3, pp. 277-289.
- 15) Jacob, S., C. Ashworth- Francone, and W. Lossow; *Structure and Function in Man*. 5th Edition. W.B. Saunders Company. Copyright 1982.

- 16) Kochanek, D. H. U., and R. H. Bartels; "Interpolating Splines with Local Tension, Continuity, and Bias Control" *Computer Graphics*, 1984, Volume 18, No. 3, pp. 33-41.
- 17) Komatsu, Koji; *Human Skin Model Capable of Natural Shape Variation Laboratories Note 329*. NHK Laboratories Note, Serial No. 329. NHK Science and Technical Research Laboratories. Tokyo. Copyright 1986.
- 18) Lensesen, Heide; *Art and Anatomy*. Barnes and Noble Inc., J.J. Augustin Inc. Copyright 1946.
- 19) Magnenat-Thalmann, N., E. Primeau, and D. Thalmann; "Abstract Muscle Action Procedures for Human Face Animation" *Visual Computer*, 1988, Volume 3, pp. 290-297.
- 20) Magnenat-Thalmann, Nadia, and Daniel Thalmann; "The Direction of Synthetic Actors in the Film *Rendez-vous a Montreal*" *IEEE Computer Graphics and Applications*, December, 1987, pp. 9-19.
- 21) Mase, Kenji, Yashito Suenaga, and Taka-aki Akimoto; "Head Reader: A Head Motion Understanding System for Better Man-Machine Interaction" *IEEE System, Man, and Cybernetics Conference - Preliminary*, 1987.
- 22) Nahas, Monique, Herve Huitric, and Michel Saintourens; "Animation of a B-Spline Figure" *Visual Computer*, 1988, Volume 3, pp. 272-276.
- 23) Newman, W., and R. Sproull; *Principles of Interactive Computer Graphics*, 2nd Edition. McGraw-Hill. New York. Copyright 1979.
- 24) Parke, Frederic I.; "Computer Generated Animation of Faces" *Proceedings of the ACM*, Copyright 1972 Association for Computing Machinery, pp. 451-457.
- 25) Parke, F. I.; "Parameterized Models for Facial Animation" *IEEE Computer Graphics and Applications*, November, 1982, Volume 2, No. 9, pp. 61-68.
- 26) Pentland, Alex P.; "Recognition by Parts" *Proceeding of the First International Conference on Computer Vision*, 1987, pp. 612-620.
- 27) Perlin, Kenneth; Personal communication. 1987.
- 28) Pieper, Steven; *More Than Skin Deep: Physical Modeling of Facial Tissue*. Master's Thesis, Massachusetts Institute of Technology. Copyright 1989.
- 29) Platt, Stephen M.; *A Structural Model of the Human Face*. Ph.D. Thesis. Dept. of Computer and Information Science, University of Pennsylvania. Copyright 1985.
- 30) Platt, S. M., and N. I. Badler; "Animating Facial Expressions" *Computer Graphics*, August, 1981, Volume 15, No. 3, pp. 245-252.
- 31) Rohan, Ellen; Personal Communication. 1988.
- 32) Smith, Alvy Ray; "The Viewing Transformation" *Technical Memo No. 84*, Lucasfilm Ltd., June 24, 1983, Revised May 4, 1984.

- 33) Stephenson, Ralph; *The Animated Film*. The International Film Guide Series. A.S. Barnes & Co., Inc. San Diego, NY. Copyright 1981.
- 34) Tost, D. , and X. Pueyo; "Human Body Animation: A Survey" *Visual Computer*, 1988, Volume 3, pp. 254-264.
- 35) Underkoffler, John S.; Personal Communication. Xx, 1988.
- 36) Vander, Arthur J., James H. Sherman, and Dorothy S. Luciano; *Human Physiology - The Mechanics of Body Function*. Second Edition. McGraw-Hill Book Co. New York. Copyright 1975.
- 37) Waters, Keith; "A Muscle Model for Animating Three-Dimensional Facial Expression" *Computer Graphics*, July, 1987, Volume 21, No. 4,
- 38) Wood, R.S.; "Orthognathic Surgical Evaluation Summary"
- 39) Zeltzer, David; "Animation and Learning: A Exploration of Three Dimensional, Realtime Microworlds with 3D Display as a Learning Environment for Children" *A Proposal to the Institute for Computer Literacy*.
- 40) Zeltzer, D.; "Towards an Integrated View of 3-D Computer Animation" *The Visual Computer*, December, 1985, Volume 1, No. 4, pp. 249-259.

ACKNOWLEDGEMENTS

Thanks to the Computer Graphics and Animation Group of the Media Lab, every one of whom it's been a pleasure to work with. Thanks especially to David Zeltzer for ideas, advice, and his library; Jim Puccio for teaching me how to *hack*; Steve Pieper for "making faces" with me; David Sturman for always being on the lookout for someone or something that could be of help to one of us and for giving me his nice, quiet office to work in that last dreadful week; to Fran Taylor for being so amazingly efficient, and to NHK for supporting this research. Thanks also to Mike Halle for his spline-to-polygon conversion code and the beautiful *X-moon* which lit my windowless summer, and to the entire Lab for being an endless resource.

Very special thanks to the people who helped in more subtle ways: to Mario, for being the "wife" and friend every grad-student needs — for pushing me, for putting up with me, and for breaking my code from time to time; to Angela, for being my best friend these last twelve years — nothing gets done without her influence; to John, for deriving the gaussian contractions with me, latexing my equations, custom anti-aliased zippers, dry-ice and o.j., the tesla-coil, posing for the *mask* with his amazing eyebrows, and for being inspired by *everything*; to Paul, for being a great officemate — always ready with a good idea and a stocked larder; and to Bill, for many *early* breakfasts, typesetting advice, doing the halftones at the last minute and for just being wonderfully weird at me.

There are many more people to thank — Frank, Ellen, Yas, Kenji, Film/Video, especially Stuart Cody, Lisa Monroe, my fellow vampires — everyone who's lent a hand in the last two years. But, it's time to finish up and hand this thing in!!

*There will be time, there will be time
To prepare a face to meet the faces that you meet;
And time for all the works and days of hands
That lift and drop a question on your plate.*

— T.S. Eliot

THANKYOU ALL!