University of Kentucky
UKnowledge

University of Kentucky Doctoral Dissertations | Graduate School

2011

THREE DIMENSIONAL MODELING AND ANIMATION OF FACIAL EXPRESSIONS

Alice J. Lin
University of Kentucky, alicejie2lin@gmail.com

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation
Lin, Alice J., "THREE DIMENSIONAL MODELING AND ANIMATION OF FACIAL EXPRESSIONS" (2011). *University of Kentucky Doctoral Dissertations*. 841.
https://uknowledge.uky.edu/gradschool_diss/841

This Dissertation is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Doctoral Dissertations by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained and attached hereto needed written permission statements(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine).

I hereby grant to The University of Kentucky and its agents the non-exclusive license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless a preapproved embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's dissertation including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Alice J. Lin, Student

Dr. Fuhua (Frank) Cheng, Major Professor

Dr. Raphael A. Finkel, Director of Graduate Studies

THREE DIMENSIONAL MODELING AND ANIMATION OF FACIAL EXPRESSIONS

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering at the
University of Kentucky

By
Alice J. Lin
Lexington, Kentucky

Director: Dr. Fuhua (Frank) Cheng, Professor of Computer Science
Lexington, Kentucky
2011

Copyright © Alice J. Lin 2011

ABSTRACT OF DISSERTATION

THREE DIMENSIONAL MODELING AND ANIMATION
OF FACIAL EXPRESSIONS

Facial expression and animation are important aspects of the 3D environment featuring human characters. These animations are frequently used in many kinds of applications and there have been many efforts to increase the realism. Three aspects are still stimulating active research: the detailed subtle facial expressions, the process of rigging a face, and the transfer of an expression from one person to another. This dissertation focuses on the above three aspects.

A system for freely designing and creating detailed, dynamic, and animated facial expressions is developed. The presented pattern functions produce detailed and animated facial expressions. The system produces realistic results with fast performance, and allows users to directly manipulate it and see immediate results.

Two unique methods for generating real-time, vivid, and animated tears have been developed and implemented. One method is for generating a teardrop that continually changes its shape as the tear drips down the face. The other is for generating a shedding tear, which is a kind of tear that seamlessly connects with the skin as it flows along the surface of the face, but remains an individual object. The methods both broaden CG and increase the realism of facial expressions.

A new method to automatically set the bones on facial/head models to speed up the rigging process of a human face is also developed. To accomplish this, vertices that describe the face/head as well as relationships between each part of the face/head are grouped. The average distance between pairs of vertices is used to place the head bones. To set the bones in the face with multi-density, the mean value of the vertices in a group is measured. The time saved with this method is significant.

A novel method to produce realistic expressions and animations by transferring an existing expression to a new facial model is developed. The approach is to transform the source model into the target model, which then has the same topology as the source

model. The displacement vectors are calculated. Each vertex in the source model is mapped to the target model. The spatial relationships of each mapped vertex are constrained.

KEYWORDS: Facial Animation, Facial Expression, Facial Rigging, Topology, Transferring Facial Expression

Alice J. Lin

November 11, 2011

THREE DIMENSIONAL MODELING AND ANIMATION
OF FACIAL EXPRESSIONS

By

Alice J. Lin

Dr. Fuhua (Frank) Cheng
Director of Dissertation

Dr. Raphael A. Finkel
Director of Graduate Studies

November 11, 2011

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Fuhua (Frank) Cheng, who has guided me throughout this study. This dissertation would not have been possible without his constant guidance, inspiration, advice and invaluable comments. I am deeply appreciative of his support.

Second, I would like to thank the other members of my PhD Advisory Committee, Dr. Jun Zhang, Dr. Ruigan Yang and Dr. Chen H. Chung. Their insightful comments and valuable suggestions helped me reach this stage. I would also like to thank the Outside Examiner, Dr. J. Robert Heath, and the Director of Graduate Studies, Dr. Raphael A. Finkel, for reviewing my dissertation and giving helpful comments.

Finally, I would like to thank my family for their unconditional love, encouragement, and support to make this research possible.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ---------- iii
LIST OF TABLES ---------- vi
LIST OF FIGURES ---------- vii
Chapter 1 Introduction ---------- 1
1.1 Milestones ---------- 2
1.2 Facial Expressions of Emotions ---------- 5
1.3 Facial Animation Techniques ---------- 7
1.4 Modeling Faces ---------- 10
1.5 Applications ---------- 13
1.6 Dissertation Contributions ---------- 16
1.7 Dissertation Organization ---------- 20
Chapter 2 Creation of Detailed, Dynamic, and Animated Facial Expressions ---------- 22
2.1 Introduction ---------- 22
2.2 Related Work ---------- 23
2.3 System Overview ---------- 26
2.3.1 Pattern Function ---------- 27
2.3.2 Setting Valid Vertices for Dynamic Animations ---------- 38
2.4 Summary ---------- 39
Chapter 3 Modeling and Animating Tears ---------- 42
3.1 Introduction ---------- 42
3.2 Background ---------- 43
3.2.1 Particle Systems ---------- 43
3.2.2 Related Work ---------- 45
3.3 Generating Animated Tears ---------- 46
3.3.1 Dripping tears ---------- 46
3.3.2 Shedding tears ---------- 47
3.4 Summary ---------- 52
Chapter 4 Automatically Placing Bones ---------- 55
4.1 Introduction ---------- 55
4.1.1 Facial Rigging ---------- 58
4.1.2 Facial Expressions ---------- 60
4.1.3 Rigging and Skinning ---------- 63
4.1.4 Motion Capture ---------- 64
4.2 Related Work ---------- 65
4.3 Automatic Placement of Bones on Face/Head ---------- 68
4.3.1 Placing the Head Bones ---------- 69
4.3.2 Multi-density Setting of the Facial Bones ---------- 72
4.4 Summary ---------- 78
Chapter 5 Copying Facial Expressions ---------- 86
5.1 Introduction ---------- 86
5.2 Background ---------- 87
5.2.1 Deforming Objects ---------- 87
5.2.2 Shape Interpolation ---------- 88
5.2.4 Performance Animation ---------- 89

5.3 Related Work---------- 94
5.4 Generating Identical Topologies ---------- 95
5.5 Summary---------- 99
Chapter 6 Conclusion and Future Work ---------- 106
Bibliography---------- 110
Vita---------- 118

LIST OF TABLES

4.1 Euclidean distances between five points. ---------- 72

LIST OF FIGURES

2.1 Six frames of a running cycle. 28
2.2 (a) Neutral facial expression. (b) Application of forehead wrinkles, eye corner wrinkles and cheek wrinkles. 29
2.3 Motion directions of the pattern functions. 31
2.4 Created pattern functions. (a) Eye corner wrinkles (b) Frown wrinkles (c) Cheek wrinkles 31
2.5 Function direction for cheeks. 32
2.6 Expression wrinkles for cheeks. 33
2.7 Expression wrinkles for the forehead. 33
2.8 (a) Neutral facial expression. (b) Dynamic forehead wrinkles. (c) Frown. 34
2.9 Created patterns. (a) Dimple (b) Eye pouch effects. 36
2.10 (a) Neutral expression. (b) Applied dimple function on cheeks. 36
2.11 Eye pouch effects. 37
2.12 Chewing food in the mouth. 41
3.1 Six selected frames of a running cycle of a teardrop changing shape. 47
3.2 Generating shedding tears. 48
3.3 Constructed object (represents the tears). 49
3.4 Making the head of the tear. 49
3.5 Four selected frames of a running cycle (dripping tears). 53
3.6 Four selected frames of a running cycle (shedding tears). 54
4.1 (a) Sadness (b) Anger (c) Joy (d) Fear (e) Disgust (f) Surprise 57
4.2 Grouping the head vertices. 70
4.3 Setting head bones. 70
4.4 The average pairwise proximities of all pairs of vertices from different groups. 71
4.5 The results of the grouping approach for the set of five points. 73
4.6 Sparse bone placements. 75
4.7 Dense bone placements. 75
4.8 (a) Relationship of head bone with facial bones. (b)The head bone controls the head rotation. 77
4.9 The operation of identifying the natural groups of vertices. 78
4.10 Facial expressions. 81
4.11 Selected frames from an animation of chewing. 83
4.12 Selected frames of a running cycle of chewing. 84
4.13 Selected frames of a running cycle as the eyes close. 85
5.1 Eight facial expressions. 100
5.2 (a) Template model. (b) Target model. (c) New mesh representation created for the target model. 101
5.3 (a) Original front view. (b) Original back view. (c) Front view of the processed mouth. (d) Back view of the processed mouth. 102
5.4 Spatial relationship between neighboring vertices. (a) Vertex relationships on the template model. (b) Correct relationship on the generated mesh. (c) Incorrect relationship on the generated mesh. 102
5.5 Examples of facial expressions and copied facial expressions. 105

Chapter 1 Introduction

Creating realistic human face models and facial animations has been a difficult challenge in Computer Animation. As facial animation often reflects a person's emotion, it has drawn researchers' interests for many years. In recent years, there has been dramatically increased interest in computer-based three-dimensional facial character animation. Facial animation is not a new endeavor. However, the recent explosion of activity in character animation has promoted a concurrent interest in facial animation. The success of a character's animation relies on two crucial parts, realistic animation of body movement and facial expression. Human facial expression has been the subject of much investigation by the scientific community. Character animators pay a great deal of attention to the face. Realistic facial animation is one of the most difficult tasks that a computer animator can be asked to do.

The face model is extremely difficult to model with a detailed geometry and natural-looking skin. Facial expression involves psychology and physiology. Adding the complex interactions between bones and muscles makes realistic facial animation exceptionally difficult. As facial expression is one of the major ways we communicate with each other, people are very sensitive to the details of facial expressions. Even a subtle change in facial expression can strongly draw the viewers' attention. The face is an important component in modeling a figure because it is the main instrument for communication and for defining a figure's mood and personality. A good facial model should be capable of geometrically representing a specific person.

Facial models can be used for cartooning, realistic character animation, telecommunications, and human-computer interaction. In cartooning, facial animations have to be able to convey exaggerated expressions and personality. In realistic character animation, the geometry and movement of the face must adhere to the constraints of realistic human anatomy. Telecommunications and human-computer interactions have the added requirement that the facial model and motion must be computationally efficient. One of the things that make human motion so challenging is that no single motion can be identified as correct human motion. Human motion varies from individual to individual for a number of reasons, but it is still recognizable as reasonable human motion, and slight variations can seem odd or awkward.

1.1 Milestones

In 1971, Chernoff [1] published his work using computer-generated two dimensional face drawings to represent a *k*-dimensional space. By using a simple graphical representation of the face, an elaborate encoding scheme was derived. Also in 1973, Gillenson [2] reported his work on an interactive system to assemble and edit two-dimensional line-drawn facial images, with the goal of creating a computerized photo identi-kit system. From 1974 through 1978, three-dimensional facial animation development was essentially dormant. However, during this period the development of two-dimensional computer-assisted animation systems continued at the New York Institute of Technology, Cornell University, and later at Hanna-Barbera. These systems supported two-dimensional cartoon animation, including facial animation. By 1974, motivated by the

desire to quickly produce facial animation, Parke [3] completed the parameterized three-dimensional face model.

In 1980, Platt [4] published a physically based muscle-controlled facial expression model. In the mid-eighties, developments in facial animation took off once more. An animated short film, Tony de Peltrie, produced by Bergeron and Lachapelle [5] in 1985, was a landmark for facial animation. This was the first computer-animated short film where three-dimensional facial expression and speech were fundamental elements of telling the story.

In 1987, Waters [6] reported a new muscle model approach to facial expression animation. This approach allowed a variety of facial expressions to be created by controlling the underlying musculature of the face. In 1988, Magnenat-Thalmann et al. [7] also described an abstract muscle action model. Another groundbreaking animation film was Tin Toy, which received an Academy Award. Produced by Pixar, Tin Toy was an example of the capabilities of computer facial animation. In particular, a muscle model was used to articulate the facial geometry of the baby into a variety of expressions [8].

The development of optical range scanners, such as the Cyberware™ optical laser scanner, provides a new wealth of data for facial animation. In 1990, Williams [9] reported the use of registered facial image texture maps as a means for 3D facial expression animation. By the late 1990s, large sets of high quality laser scanned data were being used to create detailed, morphable facial models by Blanz and Vetter [10].

The new wave of enhanced image processing and scanning technology promised to usher in a new style of facial animation. In 1993, Lee et al. [11] described techniques to map individuals into a canonical representation of the face that has known physically based motion attributes.

The late 1990s and early 2000s became a threshold for high-fidelity face capture and rendering for the film industry. Landmark films such as The Lord of the Rings (New Line Cinema 2002), The Matrix Reloaded (Warner Bros. 2003), The Polar Express (Warner Bros. 2004), and Monster House (Sony Pictures 2006) required face motion capture sessions of actors using markers and head gear. The capture sessions resulted in very large datasets, which had to be processed and rendered. Such techniques are referred to as data-driven facial animation and demand blending between more established modeling, rendering, animation techniques, and alternative approaches [12].

In the more recent past, the ability to create visual surrogates that are authentic enough to deceive observers into thinking that they are real people is close at hand. Such techniques will most likely blend animation, modeling, and control with live capture data. How such surrogates will be used is speculative; however, the film, games, medicine, and virtual online media are likely to be the first beneficiaries. The future is indeed bright for computer facial animation [13].

1.2 Facial Expressions of Emotions

Faces provide a wide range of information about a person's identity, race, sex, age, and emotional state. Numerous factors affect the perception of facial expression. It is essential to delineate the psychological space used to represent emotions. Ekman et al. [14, 15] identified six universal emotional expressions: anger, sadness, fear, surprise, happiness, and disgust. Ekman [16] extended his earlier work by compiling a list of basic emotions including amusement, anger, contempt, contentment, disgust, embarrassment, excitement, fear, guilt, pride in achievement, relief, sadness/distress, satisfaction, sensory pleasure, and shame. According to Ekman, these emotions all share basic characteristics. They are distinctive universal signals with specific physiological concomitants. They provide automatic appraisal, in particular, to distinctive universals in antecedent events. Emotions typically have a quick onset which is essential for their adaptive value. It is also adaptive for the response changes to be of brief duration unless the emotion is evoked again.

Emotions provide an important evaluative mechanism that has evolved to alert the individual to adaptationally relevant circumstances. Facial characteristics play a key part in the perception and expression of emotion and intent. As such, facial expressions represent an essential element in social interactions. Schmidt and Cohn [17] considered human facial expressions to be biological adaptations that confer a fitness advantage in the highly social environment common to humans.

In an effort to quantify facial expressions, Ekman, and Friesen [14] developed the Facial Action Coding System (FACS). Facial expressions were extensively examined and

their component motions were determined. Forty-four separate action units (AUs) were identified with five levels of intensity ranging from A to E. Thirty AUs are related to contraction of specific facial muscles. Eight AUs describe different movements related to head orientation, while four AUs are used to describe eye direction. While the use of FACS allows a high degree of refinement in classifying facial expressions, Ekman's approach to emotion and its expression remains categorical.

Facial expressions are controlled by both pyramidal and extra-pyramidal tracts which provide voluntary and automatic control, respectively. Voluntary control over facial muscles is considered a hallmark of human nonverbal expression and is probably due to the articulatory demands of human language [17]. However, there are notable differences between posed (social) and spontaneous expressions. Differences in the dynamic features of social and spontaneous smiles were investigated by Cohn and Schmidt [18]. Spontaneous smiles exhibit characteristics of automatic movement. Automatic movements are thought to be pre-programmed and are characterized by a consistent relationship between maximum duration and amplitude of movement. Posed smiles exhibit a far less consistent relationship between duration and amplitude.

In contrast to the dynamics of expression described above, several static facial features are thought to have an effect on the perception of faces. These include: shape and symmetry, size and spacing of the eyes, nasal width, fullness of the lips, hair color, skin complexion, and the width of the cheekbones. Attractiveness has been associated with the averageness and symmetry of the face [19].

1.3 Facial Animation Techniques

Animating every subtle facial action is an extremely tedious task and requires the skill of a good animator. Facial animation research falls into three major categories, those based on image manipulations, those based on geometric manipulations, and those based on performance-driven facial animation. Image manipulations comprise several subcategories which include texture manipulations [20] [21], image blending [22], and image morphing between two photographic images [23]. Geometric manipulations include key-framing, geometric interpolations [24] [25], parameterizations [26-28], finite element methods [29-31], and physics-based muscle modeling [6, 32-34]. Performance-driven facial animation [35-38] uses captured subtle facial movements and alters to the animator's desired motion.

Geometric Manipulation Techniques

The interaction between various layers of the face generates the complexity of facial animations. Hence, it is difficult to isolate representative methods from animation techniques. As a result, geometric modeling of facial animation in general deals with the whole geometry processing pipeline, which covers the range from surface generation and optimization up to surface manipulation and deformation. Geometric manipulations cover several subcategories which include interpolations, parameterizations, muscle-based modeling, and free-form deformation.

Interpolation

Interpolation is a process of generating a value based on its neighbors. Depending on the

interpolation approach, neighboring values usually contribute a certain weight to the value being interpolated. This weight is often inversely proportional to the distance at which the neighbor is located. Therefore, interpolation can provide a user-desired smooth transition between neighbors. Interpolation techniques offer not only a simple but also an intuitive approach to computer animation. In facial animation, typically, an interpolation function is used to specify a smooth transition between two facial expressions over a normalized time interval.

Interpolation can be performed in one-dimensional, two-dimensional, or three-dimensional space. Interpolation approaches in computer animation usually fall into two categories: linear interpolation and cubic interpolation. For simplicity, linear interpolation is commonly used to generate facial animation. The pioneering work of Parke [24] introduced simple geometric interpolation between face models to generate animated sequences of a human face changing expression. It proves that facial animations can be achieved through a simple face model with a polygonal skin containing approximately 300 polygons defined by about 400 vertices, and a cosine interpolation scheme to fill in the intermediate frames between expressions. With bilinear interpolation, Arai et al. [25] proposed a method to generate facial animations in which the facial expression and shape can be changed simultaneously in real time. In their model, facial expressions are superimposed onto the face shape through the 2D parameter space independents of the face shape.

Parameterization

Compared to geometric interpolation, which directly moves positions of the face mesh vertices, parameter interpolation controls functions that indirectly manipulate the vertices. Using these parametric functions, the animator can create facial images by specifying the appropriate set of facial parameters.

The pioneering parametric models [3] proposed by Parke are based on the concepts of simple parameterization and image synthesis. Parameterization includes choosing the relevant facial parameters based on exterior observation or on underlying structures that cause the specific expression [27].

Further work by Parke and Walters [13] continued to reduce the parameters and used what they call direct parameterization to animate facial expressions. In their method, the face is still represented by polygonal meshes, but the animation can be simulated by a far smaller set of parameters, though the underlying animation approaches are still based on key-framing and interpolation. Compared to interpolation approaches, direct parameterization provides more intuitive ways to animate facial expressions. However, this method has its own problems. The set of parameters is not universal for all faces, but is bound to a certain facial topology. To create a different face, the animator will have to reset the parameters. Furthermore, when there is conflict between parameters, the resulting facial expressions will look unnatural. To avoid this undesired effect, parameters are set to only affect specific facial regions. However, this often results in noticeable motion discontinuity, especially in the boundaries of the region.

This early model was implemented with relatively primitive techniques. While current techniques allow much more sophistication, this model still has relevance. Direct derivatives of this model are still being used, and many of the techniques embodied in it are useful in simple, low-complexity face models, such as those used in computer games and avatars.

The ideal parameterized model would be the one that allows any possible face with any possible expression to be specified by merely selecting the appropriate parameter value set. A parameterization that enables all possible individual faces and all possible expressions and expression transitions is referred to as a complete or universal parameterization. The ad hoc parameterized face model described here certainly falls far short of this ideal, but it does allow a wide range of expressions for a fairly wide range of individual facial conformations.

1.4 Modeling Faces

Developing a facial model involves determining geometric descriptions and animation capabilities that represent the faces of interest. It also involves the representation of additional attributes, such as surface colors and textures. For our purposes, static models are not useful. Facial models must be constructed in ways that support animation.

The face has a very complex, flexible, three-dimensional surface. It has color and texture variation and usually contains creases and wrinkles. The detailed anatomy of the head and face is a complex, dynamic assembly of bones, cartilage, muscles, nerves, blood

vessels, glands, fatty tissue, connective tissue, and skin. To date, no facial animation models that represent and simulate this complete, detailed anatomy have been reported. For some applications, such as medical visualization and surgical planning, complete detailed models are the ultimate goal. Fortunately, a number of useful applications, such as character animation, can be accomplished with facial models that approximate only some visual and behavioral aspects of the complete facial anatomy.

The modeled facial geometry and its animation potential are inseparably intertwined. The structure of the model determines its animation potential. The actions a model must perform determine how the model should be constructed. Choices made early in the modeling process determine its animation capabilities. The mechanics of the face and head are extremely important when constructing these models. The jaw needs to work, the eyelids need to open and close, the eyelids need to stretch over the eyeballs, and the cheeks need to puff and stretch. There are subtle points to be considered, as well. When the eyes move from one side to the other, how much of the flesh surrounding the eyes should be moved? When the mouth opens and closes, how much of the neck is influenced? When the mouth opens, how much do the cheeks move, and how far up the cheeks will there be an effect? How is the nose affected? Such details can be the key to a character's age, personality and believability. The eyes and mouth are the most expressive areas of the face. They communicate the most information and elicit the greatest emotional responses. Care must be exercised with these regions, so that the necessary details and flexibility are included in the model.

Even though most faces have similar structures and the same feature set, obviously there is considerable variation from one individual face to the next. This subtle variation is exactly what makes individual faces recognizable. One of the challenges of facial animation is to develop models that support and allow these variations. Artists and illustrators have developed many approaches to depicting the human face. These range from sketches and realistic renderings to cartoons and abstract forms to surrealism. Almost any of these artistic styles might be used as the basis for computer models of the face.

One popular form of facial representation is the caricature. Caricatures typically involve distorting or exaggerating the most recognizable features of a specific face. This often is done in editorial cartoons of famous people. Caricatures are formed by exaggerating the differences of individual faces from the computed normal face. A similar approach can be applied to three-dimensional faces. In animation, we are usually concerned with telling a story, which means getting the audience involved with the characters. This involvement implies that we must develop our facial models so that the audience can establish an emotional response to the character. We want the audience to hate the villain, love the heroine, laugh with the comic, and feel sorry for the unfortunate victim. Developing such empathetic characters is one of the goals of facial modeling [13].

Contour reconstruction is a technique of reconstructing 3D surfaces using multiple 2D data points and analysis of the positions of the object to be modeled. An algorithm for automatically creating such polygon skins was first described by Fuchs et al [39].

Contour reconstruction techniques are better suited to modeling of the softer tissue, cross-sections of which are easier to obtain via magnetic resonance imaging (MRI) and computer tomography (CT) scans [40]. The popular laser-based scanning system captures both surface and color information simultaneously. The whole process is automated and controlled by the software and hardware supplied. Edges of the polygons of scanned data are not necessarily consistent with creases on a face, as desirable by the modeling requirements. Neither are they consistent with the requirements of division of polygons in such a way that surfaces are not deformed during animation. Several methods exist for matching scanned data and desired facial topology, namely interactive fitting [13], topology adaptation [34, 41, 42] and adaptive meshes [41].

1.5 Applications

By far the largest motivator, developer, and consumer of three-dimensional facial character animation is the animation industry itself. While the animation studios continue to shape how computers are used in animation, other emerging areas that influence animation are briefly mentioned below.

Games Industry

The games industry has experienced rapidly recent development, due in part to increasing processor performance, coupled with more and more powerful graphics coprocessors. Such hardware advances have opened the door to more sophisticated real-time three dimensional animation software that is different from the techniques employed in film production animation. High quality real-time rendering is now commonplace.

While a maturing games industry has benefited from real-time performance, it also plays an important role in film production, where animated scenes can be blocked in and combined with live action well before final rendering is required. Within the context of facial animation, real-time sequencing and synchronization within the scene can save enormous amounts of time in film production.

It is clear that a number of opportunities for the next-generation facial animation techniques will be at the intersection of real-time performance and off-line, non-real-time production rendering. For the games industry, the challenges will be to combine lifelike performances with which players can interact. For the face in particular, it will be increasingly important for the characters to engage in non-verbal communication similar to what we experience in the real world.

Video Teleconferencing

The ability to transmit and receive facial images is at the heart of video teleconferencing. Despite the rapid growth of available communication bandwidth, there remains a need for compression algorithms. One active research area is in model-based coding schemes and, in particular, algorithms applied to facial images [43].

For a very low-bandwidth face video conferencing system, each captured frame from the video is analyzed by the encoder, with the assumption that the principal object in the scene is a human face. Computer vision algorithms are then used to extract and parameterize properties such as the shape, orientation, and motion of the head and face

features. These few parameters are compressed and transmitted to the decoder, where a three-dimensional model of the human head is synthesized to create a visual surrogate of the individual. As the head moves from frame to frame, new parameters are transmitted to the receiver and subsequently synthesized. This procedure is in contrast to existing video teleconferencing compression techniques that deal exclusively with compression and transmission of pixel-based image data.

While teleconferencing, there remain a number of key initialization procedures for the encoder and decoder. As part of the initialization procedure for the encoder, features of the face must be accurately aligned, such that the mouth and eyes open in the correct location with respect to the image texture map. One of the by-products of mapping images of individuals to canonical representations of the face is that any image can be used to create a novel character.

Social Agents and Avatars

A developing area for facial animation is in user interfaces that have characters or agents. The principle of social agents lies in the ability of an agent to interact directly with the user. This ability can be as simple as a reactive behavior to a simple action, such as searching for a file, or as complex as an embodiment or characterization of a personal assistant capable of navigating the Internet under voice commands and responding audibly and visually with a resulting find. Some themes include characters that display their activity state through facial expressions.

Ultimately, these agents will understand spoken requests, speak to the user, behave in real time, and respond with uncanny realism. These interfaces often are referred to as social user interfaces and are designed to supplement graphical user interfaces. For example, a character will appear to assist you when you start a new application. If you hesitate or ask for help, the agent will reappear to provide you with further guidance. In many instances, these characters will be seen as active collaborators, with personalities of their own.

Robots present a new frontier for experiments to understand what makes us human. Not only is it possible to mimic human responses and behaviors, but new types of robots can serve as human surrogates. Unlike a computer-generated character that is constrained to a two-dimensional display, a physical embodiment of a robot has to move in the real world. This presents engineers with significant additional challenges. Nevertheless, the development of robot agents shares many of the underlying concepts developed for computer-generated three-dimensional characters [13].

1.6 Dissertation Contributions

Facial expressions and animations are important aspects of the 3D environment featuring human characters. It is in high demand by many kinds of applications, such as 3D games, interactive human/computer software, and movies. However, a good facial expression or animation requires a lot of time for a skilled artist. The complexity of the human faces may provide subtle expressions, and each spectator is able to know instinctively if an expression or animation is correct or not.

Since the pioneering work of Parke [24], many efforts were provided to increase the realism and satisfy the critical eyes of humans. Nevertheless, three aspects of facial animation are still stimulating active research: the detailed subtle facial expressions, the process of rigging a face, and the transfer of an expression from one face model to another. Our research work for the dissertation is focused on the above three aspects. In this section, we summarize the contributions of the dissertation research work as follows.

(1) We developed a system for freely designing and creating detailed, dynamic, and animated facial expressions. We employed the pattern functions to generate detailed, animated facial expressions. We created wrinkle patterns (which include forehead wrinkle pattern, eye corner wrinkle pattern, cheek wrinkle pattern and frown wrinkle pattern), dimple pattern and eye pouch effect pattern. The pattern function controls the polygon and directly controls the displacements of the mesh vertices. Displacing the vertices changes the original position of the vertex, based on the calculated value of the pattern function. To produce dynamic animations, the pattern function is not applied to each vertex of the selected region. For a selected region we only set certain vertices as valid and keep the remaining vertices as null. We only compute the deformation of the selected surface at each vertex that is influenced by pattern functions. The valid vertices are chosen during each animation period. The valid vertices are randomly chosen and cover all areas in the selected region. Our method has produced realistic results with fast performance, and the process for generating facial animations requires less human intervention or tedious tuning. The tool is intuitive and easy to use, allowing users to directly manipulate it and see immediate results. The system also allows the expressive

features to be applied to any existing animation or model. The location and style can be specified, thus allowing the dynamic detailed facial expressions to be modeled as desired. It can also be applied to a variety of CG human body skins and animal faces. The system is suitable not only for professional production of facial expressions, but also for entertaining people at home.

(2) We developed methods for generating realistic, animated tears. We developed a formula for generating teardrops, which continually change its shape as the tear drips down the face. We also developed a method for generating shedding tears. This kind of tear flows along the surface of the skin on the face. The tear travels over the skin, but remains an individual object. The tear seamlessly connects with the skin and will roll down as time passes. We show the method for generating shedding tears by using the following steps. We select the areas that the tear will pass through on the face, and then extract this area as an individual surface. From the outline of the base surface, we construct the object that the tear will wrap around on the base surface. We make the head of the tear when the tear is rolling down the face. We randomly add noise to the tear surface to simulate the roughness of the skin. We randomly choose vertices to rise from the top surface, and perform moving least squares to deform the top surface to achieve simulation of roughness on the skin. For the animation of shedding tears, we select a certain portion of the vertices on the top side of the object where the tear starts. As time passes, the selected portion of the object increases in size. For each time point, there is a corresponding set of vertices. The simulation shows real-time vivid rolling of the tears down the face. Our methods both broaden CG and increase the realism of facial

expressions. Our methods can also be applied to CG human body skins and other surfaces to simulate flowing water.

(3) We developed a method to automatically set the bones on facial/head models to speed up the rigging process of a human face. To accomplish this, we group vertices based on information found in the vertices that describe the face/head as well as relationships between each part of the face/head. To place the head bones, we start with individual vertices as groups by placing each vertex into its own group, then successively merging the two closest groups into larger and larger groups until the termination conditions are satisfied. The average distance between pairs of vertices is used as the distance between two groups. The head bone passes the mean in the height of head direction. The neck bone passes the mean and parallels the neck direction. We place two bones connected together through these two means to control the head animation. To set the bones in the face with multi-density, we take the input parameter and partition a set of vertices into groups so that the resulting intragroup similarity is high but the intergroup similarity is low. Group similarity is measured in regard to the mean value of the vertices in a group, which can be viewed as the group's centroid. The bone is placed in the centroid to control the group's vertices. The resulting facial deformations and animations are realistic and expressive. The method works on a particular person's face/head model quickly and effectively, and provides a powerful means of generating facial expressions and animations. The time saved with this method is significant. Although we have focused entirely on the human face in this work, we believe the concepts presented here are flexible enough to be used on non-human characters. Our method, in conjunction with

existing techniques, allows users to go from a static mesh to an animated character quickly and effortlessly.

(4) We developed a method to produce realistic expressions and animations by transferring an existing expression to a new facial model, instead of generating one from scratch. Our novel approach is to transform the template (source) model into the target model, which then has the same topology as the template model. We first search for corresponding facial feature points in both models. We mark the eye corners, upper and lower eyelids, mouth corners, uppermost and lowermost parts of the lips, nose tip and edges, earlobes, etc. We position these two models and try to superpose them according to correspondence between features. We calculate the displacement vectors for the facial feature points, and non-facial feature points. When specific facial feature points are matched between the two models, the morphing is performed on the mesh of the template model. Once that is done, we map each vertex in the template model to the surface of the target model by determining the minimum distance between two vertices. We constrain each mapped vertex to the same spatial relationships with its neighboring vertices as in the template (source) model. Once the target and template models have the same mesh structures, we transfer facial expressions from the template model by using its motion data.

1.7 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 presents a new system for generating detailed, dynamic, and animated 3D facial expressions. Chapter 3 describes a novel method for modeling and animating tears. Chapter 4 introduces a new

method for automatic placement of bones on the face for 3D facial expressions and animations. Chapter 5 presents a method for copying facial expressions. We conclude the dissertation in Chapter 6 and point out directions for future work.

Copyright © Alice J. Lin 2011

Chapter 2 Creation of Detailed, Dynamic, and Animated Facial Expressions

2.1 Introduction

Generating compelling dynamic animated facial expressions is an extremely important and challenging aspect of computer graphics. Humans depicted by computer graphics are not fully satisfying because they lack visually important details, despite the fact that they have been well designed and animated in recent 3D games, virtual reality, computer vision and animation movies. Their skin remains unrealistically smooth independently of their facial animations or bodily deformations. The important aspect of character animations is the character's ability to use facial expressions. We often depend on the subtleties of facial expressions to give us important contextual cues about what someone is saying to us.

Detailed surface geometry contributes greatly to the visual realism of 3D face models. The more the detailed facial expressions are enhanced, the better the visual reality will be. However, it is often tedious and expensive to model facial details which are authentic and real-time. Our system is a simple and functional approach for modeling dynamic detailed facial expressions. Humans are very accustomed to looking at real faces and can easily spot artifacts in computer-generated models. Ideally, a face model should be indistinguishable from a real face. Recently, the increasing appearance of virtual characters in computer games, commercials, and movies makes the detailed facial expressions even more important.

Computer graphics researchers have been greatly interested in this subject during the last decade. There have been many methods to produce detailed facial expressions, but the cost of production was quite high, often requiring professional skill, special devices and time consuming. The goal of our system is to create a viable alternative simulation which could be used for creating detailed animated facial expressions. In our system, we can create these facial expressions in a series of animations. The interactive tool is intuitive and easy for users in any age, allowing them to directly manipulate it and see immediate results. It is not only suitable for professional production of facial expressions, but also suitable for use by lay people to entertain themselves at home.

2.2 Related Work

Facial animation approaches could be grouped into two groups, those based on geometry manipulation and those based on image manipulation [44, 45]. Geometry manipulation refers to manipulation of 3D models that consist of vertices in space forming polygons and thus represent the surface of the model. Image manipulation refers to 2D images or photos that are morphed from one to another in order to achieve a desired animation effect. It is generally agreed that one could divide the geometry manipulation methods roughly into interpolation [46], finite element methods [47, 48], parameterization [26], pseudo-muscle methods [49], and physics-based methods [6, 34, 50, 51]. The image manipulation methods include morphing and blendshaping [52]. The performance-driven animation (also referred to as expression mapping) assumes that a performer makes appropriate use of both geometry-based and image-based techniques to do the animation [53].

The basic idea of Marker-Based Motion Capture is combining 3D facial geometry with marker-based motion capture data [9]. The commercial markets for marker-based facial-capture systems acquire data with excellent temporal resolution, but due to their low spatial resolution, they are not capable of capturing expression wrinkles. Structured light techniques are capable of capturing models of dynamic faces in real time. However, structured light systems cannot match the spatial resolution of high-quality static face scans [54, 55] or the acquisition speed of marker-based systems. They also have difficulties in dealing with the concavities and self-shadowing that is typical for expression wrinkles [56].

Blinn introduced bump mapping [57], which consists of modifying a surface's normal before lighting calculations to give a visual impression of wrinkles, without deforming the geometry. This idea has been extensively used since then, especially for facial wrinkles. However, this technique suffers several drawbacks. For example, the geometry used in this technique remains undeformed and the silhouette of the objects is visually incorrect (noticeable in close views). Oat [58] presented a technique that involves compositing multiple wrinkle maps and artist animated weights to create a final wrinkled normal map that is used to render a human face. This method allows blending multiple wrinkle maps across several regions of a textured mesh. Wrinkle maps are bump maps that get added on top of a base normal map. Zhang et al. [59] demonstrated a method for reconstructing animated, anatomy-based facial models of individuals from ranged data. A prototype model with a multi-layer skin-muscle-skull structure serves as the starting point for the method. After the global adaptation, this skin mesh of the

prototype model is deformed to fit scanned data according to internal force stemming from the elastic properties of the surface and external forces produced from the scanned data points and features.

Pighin et al. [60] presented techniques for creating photorealistic textured 3D facial models from photographs of a human subject, and for creating smooth transitions between different facial expressions by morphing between these different models. Bickel et al. [56] presented a multi-scale representation and acquisition method for the animation of facial geometry and wrinkles. They first acquired a static scan of the face, and then augmented a traditional marker-based facial motion-capture system by two synchronized video cameras to track expression wrinkles. The resulting model consists of motion-capture data and expression wrinkles in 2D parametric form. This combination represents the facial shape and its salient features at multiple scales. The lack of a strict vertex correspondence between wrinkles and mesh over time could potentially lead to minor drifts of the wrinkles. Zhang et al. [61] presented a geometry-driven facial expression system, which includes a framework for synthesizing facial expression textures from facial feature point motions by using an example-based approach. Generating character animations and 3D facial expressions were through sketching [62-66]. Chang and Jenkins [64] presented a 2D sketch interface for posing 3D faces. In their work, users can intuitively draw 2D strokes that are used to search for the optimal face pose. The work of Davis et al. [62, 65] presented sketching interfaces for efficiently prototyping articulated character animations.

2.3 System Overview

As subtle facial expressions are induced by facial animation, they are defined according to muscle anatomy, contraction and innervations. Texture mapping offers a good simulation of static facial expression, but it is difficult to mimic the expression deformation during animation by a purely geometric model with texture mapping. The simulation of subtle facial expressions is achieved by changing the geometric structure of a region of the 3D face model.

The system allows users to create realistic, compelling dynamic detailed facial expressions. The user can manipulate the shape and style, position and orient the detailed expressions on a facial mesh, and control how the facial expressions interact with the face animation. The expressive features can be added to any existing animation. When the local resolution of the mesh is not sufficient, it can be refined according to the finest expressed feature. The user can alter parameters of patterns to adjust and create facial expressions and repeat the process until the desired results are achieved.

Our system has two main phases to create a series of facial expressions. We select the region that we want to simulate. For the selected region, we subdivide the vertices (or polygons) to make the meshes fine enough for detailed simulation and smoothness. For the dynamic animation we need to select the percentage of valid vertices in the selected area. In the emotion expression creation phase, we interactively deform the face using deformation methods to generate expressions, such as smiles and surprises. In the

detailed expression phase, we apply an animated pattern function to acquire the desire results.

2.3.1 Pattern Function

We defined the pattern function as a combination of primary function P and damping. Primary function $P = p_1 + p_2 + ...,$ can be only one basic pattern function p_1 or several basic pattern functions p_i, $i = 1, 2, 3, ...$. The pattern function is layered on top of the selected area. It describes the surface of the skin change. Since the pattern function adjusts the vertices sequentially, the form and shape in the selected area change over time. Adjusting the parameters of the pattern functions precisely controls the shape of the expression. Moving the layer of pattern functions refines the expression location. The detailed movement of the expression is a function of the time parameter t in each time interval.

Damp function influences the smoothness around the selected area where we applied expression patterns. The damp function dictates expression patterns to decrease and then vanish toward the borders of the selected region.

2.3.1.1 Wrinkles

Research in facial expression has concluded that there are six *universal* categories of facial expressions that are recognized across cultures [67]. These categories are sadness,

anger, joy, fear, disgust, and surprise. Faigin [68] presented an excellent discussion that associated with facial wrinkles, from an artist's point of view.

The wrinkles associated with sadness include horizontal folds across the brow, trace vertical lines between the eyebrows, oblique folds above the upper eyelid, and a smile-shaped fold under the lower lip. The wrinkles for anger include horizontal folds above the

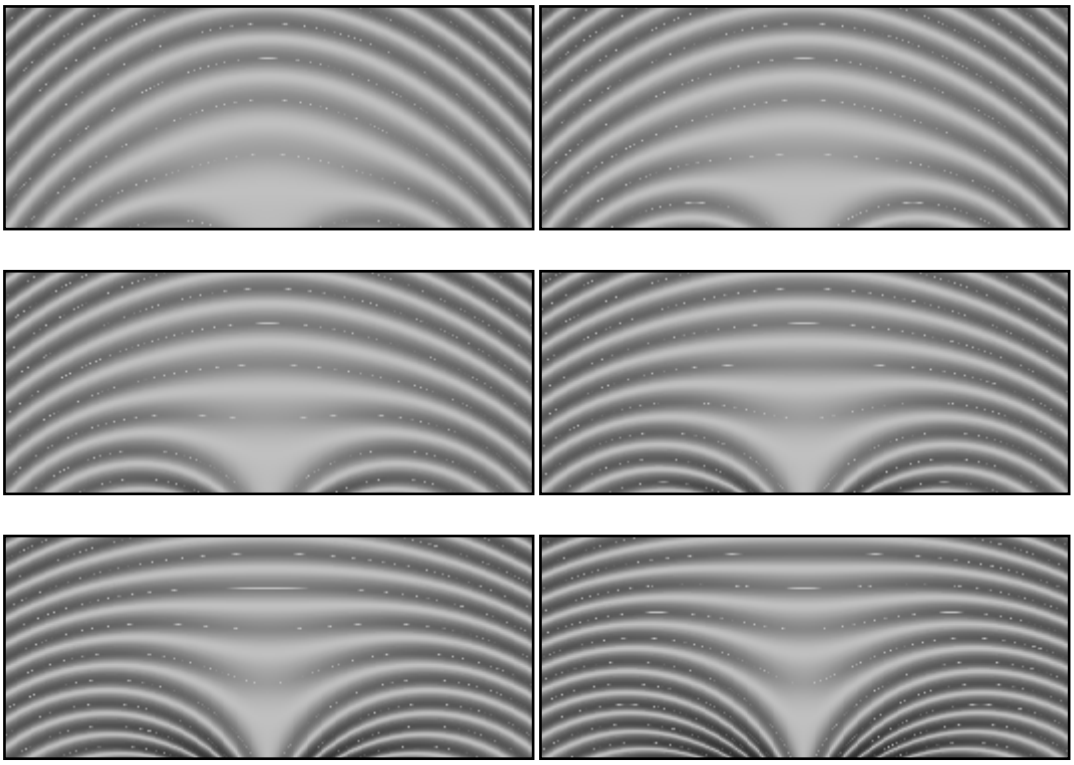


Figure 2.1: Six frames of a running cycle.

upper eyelids and vertical lines between the eyebrows. For joy, the wrinkles create "crow's feet" at the corners of the eyes, a smile-shaped fold under the lower eyelid, dimples, and a deep nasolabial fold from nose to chin. The wrinkles associated with fear include horizontal brow folds, vertical lines between the eyebrows, dimples above the

eyebrows, and oblique folds above the upper eyelids. In disgust, there are vertical lines between the brows, crow's feet and lower eyelid creases, wrinkles from the inner corner of the eye across the bridge of the nose, and a chin bulge. In surprise, horizontal folds are formed across the brow.

(a) (b)

Figure 2.2: (a) Neutral facial expression. (b) Application of forehead wrinkles, eye corner wrinkles and cheek wrinkles.

For human faces, wrinkles can roughly be classified into two different types: aging wrinkles and expressive wrinkles. Wrinkles are formed due to compression of the skin surface caused by movement of facial muscle or aging. Since skin itself is incompressible, when the skin surface shrinks, the excess skin buckles and forms wrinkles. These become gradually more prominent with expression and aging. Aging wrinkles are permanent and

once they appear on face, they become gradually more pronounced with every passing day. Expressive wrinkles are temporary wrinkles that appear on the face during expressions at all ages. They can appear anywhere on the face during expressions, but for most people, the expressive wrinkles mainly come forth on several regions of the face such as the cheeks, the forehead and eye corners. From these effects, we have developed a set of basic pattern functions.

Forehead wrinkles:

The pattern function of forehead wrinkles is shown below. Figure 2.1 shows six frames of a running cycle of this pattern. We applied this pattern function to the forehead surface (Figure 2.2(a)). The facial model with wrinkles in forehead is shown in Figure 2.2 (b). For the forehead, when a person makes an expression such as opening the eyes widely, the pattern function starts at the points indicated by the arrows and propagates in the directions of the arrows (Figure 2.3 (a)). When a person reverses the expression, such as by relaxing the eyes, the function starts in the opposite directions to erase the pattern (Figure 2.3 (b)).

$$p = e^{\alpha \times \left| \sin((x^2+y^2)-(t \times \sin(\arctan(\frac{y}{x}))-\sin(\beta \times \arctan(\frac{y}{x})))^2) \right|} - z$$

α and β are parameters. t is time.

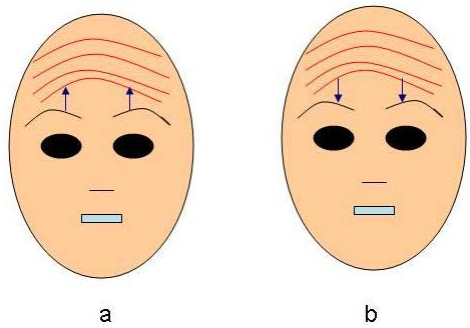


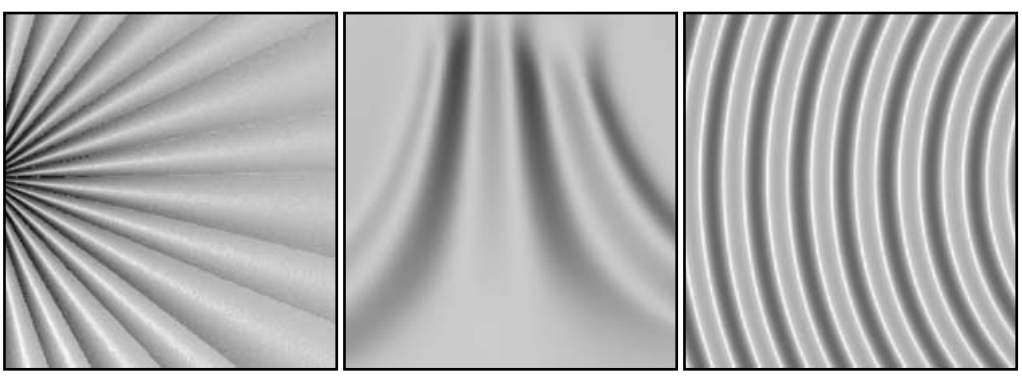
Figure 2.3: Motion directions of the pattern functions.

Eye corner wrinkles:

For eye corner wrinkles, we developed the pattern function, which is shown below. Figure 2.4(a) is the pattern for eye corner wrinkles. We applied the pattern to both eye corners (Figure 2.2 (a)). The result is shown in Figure 2.2 (b).

$$p = \alpha \times e^{\beta \times \left| \sqrt{x^2+y^2} - t \times \sin(\gamma \times \arctan(y/x)) \right|} - z$$

α, β and γ are parameters. t is time.



(a) (b) (c)

Figure 2.4: Created pattern functions. (a) Eye corner wrinkles (b) Frown wrinkles (c) Cheek wrinkles

Cheek wrinkles:

The position of the wrinkle is vertical to the muscle movement direction. Due to the mouth animation, the cheeks usually have muscle contractions and generate expression wrinkles. The pattern function we developed for cheek wrinkles is shown below. Figure 2.4(c) is the pattern for cheek wrinkles. We add pattern functions to cheeks to improve mouth animations and make the face look alive. This achieves the realism of cheek skin. When the mouth opens bigger, the directions of the pattern function applied to cheek skin are indicated as arrow directions (see Figure 2.5). When the mouth becomes smaller, the function is applied in the opposite directions. Examples are shown in Figure 2.2 (b) and Figure 2.6.

$$p = \alpha \times \sin(\beta \times \sqrt{x^2 + y^2} + t) - z$$

α, β are parameters. t is time.

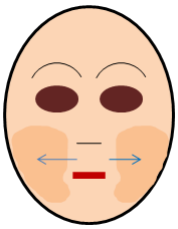


Figure 2.5: Function direction for cheeks.

Figure 2.6: Expression wrinkles for cheeks.

Figure 2.7: Expression wrinkles for the forehead.

Frown wrinkles:

The equation below and Figure 2.4 (b) is the pattern function we created for the frown. An example of applying this function to Figure 2.8(a) is shown in Figure 2.8 (c).

$$p = \alpha \times \sin(x \times y / (\beta + t)) - z$$

α, β are parameters. t is time.

(a) (b) (c)

Figure 2.8: (a) Neutral facial expression. (b) Dynamic forehead wrinkles. (c) Frown.

2.3.1.2 Dimple and Eye Pouch Effects

Dimple:

Dimples are visible indentations of the skin. In most cases, facial dimples appear on the cheeks, and they are typically not visible until someone smiles. The changes in the face

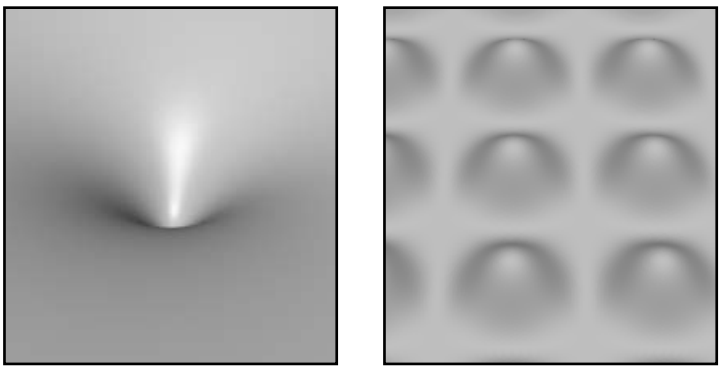
caused by a smile will bring out the dimple. However, some people only have a dimple on one side; this physical trait can actually be rather endearing in many cultures. The look of dimples can also vary; as an inherited trait, unusual dimples can be passed on through multiple generations of a family.

Most dimples are actually caused by a birth defect, which just goes to show you that not all “malformations” are actually bad. The most common cause of dimples is a shortened muscle, which explains why dimples are not always apparent at rest, since muscles are typically in their shortened state at rest. In the face, shorter face muscles pull at the skin, especially when someone smiles, creating a classic dimple. Over time, the muscles of the face can slowly stretch out, which is why some people have dimples when they are young, but lose them as they age.

Because dimples can fade with time as someone's facial muscles stretch out, many people associate them with youth. Children are referred to as having dimpled cheeks, for example, and someone who has dimples may be thought of as youthful or baby faced. The pattern function we developed for a dimple is shown below. Figure 2.9 (a) is the pattern for dimple. We applied the dimple function on Figure 2.10 (a), and generated the result in Figure 2.10 (b).

$$p = \log(x^2 + y^2 + t) - z$$

t is time.



(a) (b)

Figure 2.9: Created patterns. (a) Dimple (b) Eye pouch effects.

(a) (b)

Figure 2.10: (a) Neutral expression. (b) Applied dimple function on cheeks.

Eye Pouch Effects:

In 3D facial expressions, people usually either exaggerate the emotion or make a straight face. Emotions, such as anger or sadness, are usually expressed as opening the mouth and

eyes widely and furrowing the eyebrows or deforming the face. Subtle facial expressions also express a person's emotion. To show an angry or sad person, for instance, we can use facial skin trembling on different regions, such as the eye pouches, chin, cheeks or temple. A pattern function that we have developed for eye pouch effects is shown below:

$$p = \frac{\left|(\cos(x + \alpha \times t) + \cos(y + \beta \times t))^{\gamma}\right|}{t} - z$$

Where α, β and γ are parameters and t is time. The degree of impact can be manipulated by adjusting parameters of the equation. The pattern is shown in Figure 2.9(b). Figure 2.11 shows a frame of a running cycle.

Figure 2.11: Eye pouch effects.

2.3.2 Setting Valid Vertices for Dynamic Animations

The pattern function controls the polygon and directly controls the displacements of the mesh vertices. Displacing the vertices changes the original position of the vertex, based on the calculated value of the pattern function. To produce dynamic animations, the pattern function is not applied to each vertex of the selected region. For a selected region with N vertices, we only set certain n vertices as valid and keep the remaining N - n vertices as null. We only compute the deformation of the selected surface at each vertex that is influenced by pattern functions. The valid vertices are chosen during each animation period. The valid vertices should be randomly chosen and cover all areas in the selected region.

We know all the vertices and their indices in the selected region. Suppose n vertices are to be selected as a list. The vertices may be selected randomly, in increasing/decreasing index order or in cyclically increasing/decreasing index order. We select an interval k and pick vertices at equal intervals along the list.

$k = N / n$:

For a small N, we randomly select one vertex from the first k vertices in the vertex list and every k^{th} vertex thereafter is selected. For a large N, we select vertices in the same manner. Suppose we repeatedly select r times: $k_r = rk$. We select r random vertices between the first and the k_r^{th} vertices. Then, the constant k_r is added to each of these random start points in order to obtain r vertices between the k_r^{th} and $2k_r^{th}$ vertices; the process of adding the constant is continued until r times of each n/r vertices are obtained. If the selected vertices are in random order, the result is equivalent to random

selection. If selected vertices are in periodic order, using this method will hit both the peaks and valleys of a cyclical pattern. Thus, it also achieves the random result. We alter the parameters - the percentage of vertices in the area for the animation to get different dynamic results.

To generate dynamic expression wrinkles, for instance, we continuously change the number and position/shape of the wrinkles during animations by randomly choosing valid vertices. Figure 2.7 and Figure 2.8 (b) are examples. An example of chewing food in the mouth is shown in Figure 2.12.

2.4 Summary

We have presented a system for generating detailed dynamic facial expressions. Despite the complexity of human facial anatomy and people's inherent sensitivity to facial appearances, we have created a real-time system that generates realistic subtle facial expressions and adapts to individual faces. Our method has produced realistic results with fast performance, and the process for generating facial animation requires less human intervention or tedious tuning. The tool is intuitive and easy to use, allowing users to directly manipulate it and see immediate results. The system also allows the expressive features to be applied to any existing animation or model. The location and style can be specified, thus allowing the dynamic detailed facial expressions to be modeled as desired. It also can be applied to a variety of CG human body skins and animal faces.

In the future, we will extend our research to mouth animation. The inside parts of the mouth play important roles in facial animation, especially in speech simulation. In most facial animation, the teeth, gums and tongue are often omitted or oversimplified. When modeled, the mouth is usually represented as a simple parallelepiped. Although only a small portion inside the mouth is visible during normal speech, its simulation is important for realistic synthesized mouth animation.

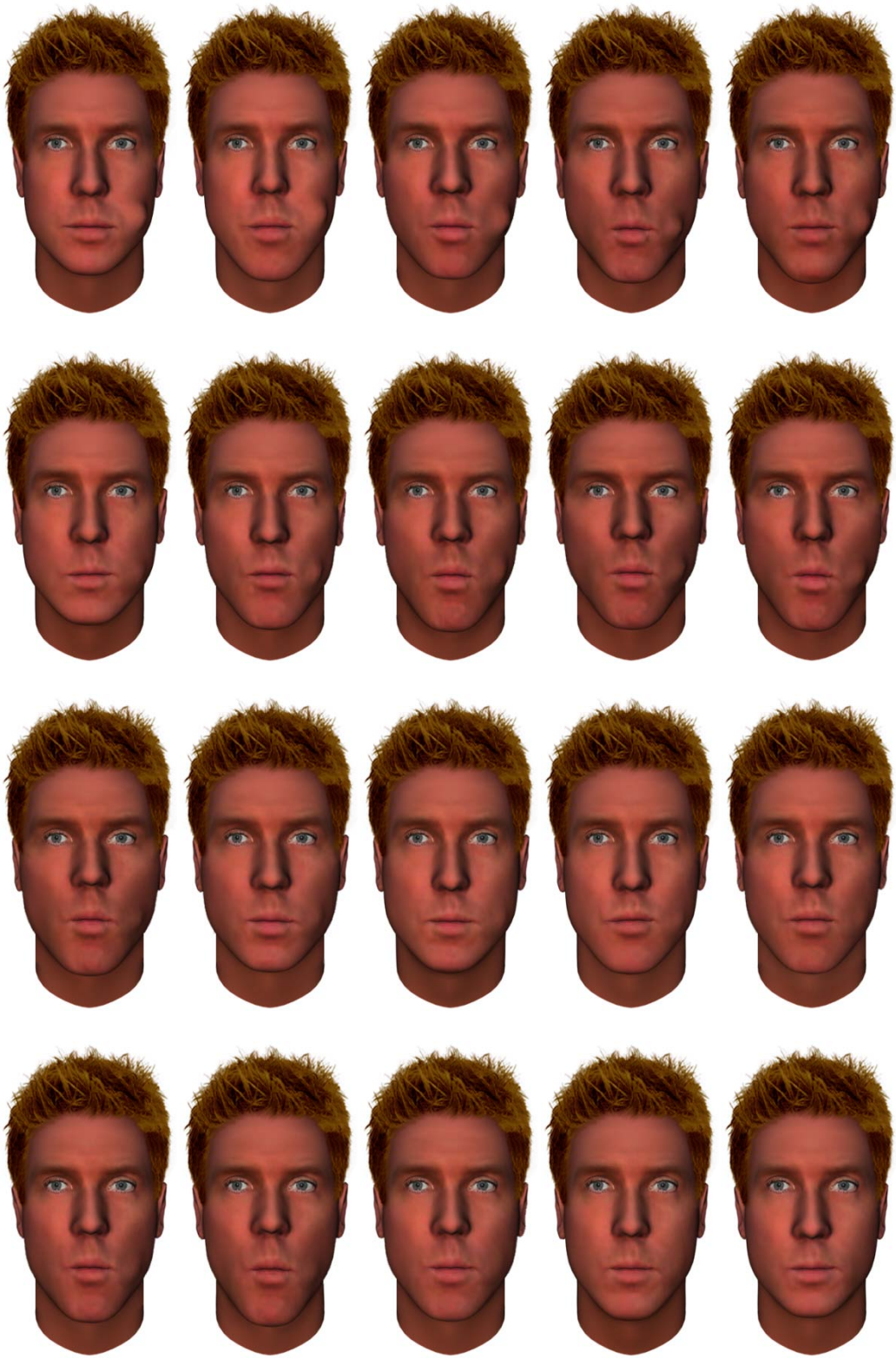


Figure 2.12: Chewing food in the mouth.

Copyright © Alice J. Lin 2011

Chapter 3 Modeling and Animating Tears

3.1 Introduction

Humans are very accustomed to looking at real faces and can easily spot artifacts in computer-generated models. Ideally, a facial model should be indistinguishable from a real face. Generating compelling animated facial expressions is an extremely important and challenging aspect of computer graphics. The most important aspect of character animations is the character's ability to use facial expressions. Although well designed and animated in recent 3D games, virtual reality, computer vision or animation movies, humans depicted by computer graphics are not fully satisfying since they lack important expressions - emotions.

Emotions play a critical role in creating engaging and believable characters. Facial movements and expressions are therefore the most effective and accurate ways of communicating one's emotions. Emotions and feelings can be difficult at times to put into words but our behavioral reactions to these feelings are universally accepted as being similar and easily noticeable. Although weeping and tears are a common concomitant of sad expressions, tears are not indicative of any particular emotion, as in tears of joy. A happy face with tears appears more joyous, or something in between, perhaps described as bittersweet. As far as scientists can tell, no other creature cries emotional tears the way humans do. The tear effect appears unique to our species. With tears, the face increases the richness as an instrument for communication. The way humans read other people's emotions is markedly impacted by the presence (or absence) of tears.

Provine et al. [69, 70] provided the first experimental demonstration that tears are a visual signal of sadness by contrasting the perceived sadness of human facial images with tears against copies of those images that had the tears digitally removed. Tear removal produced faces rated as less sad. The research subjects suggest further that after tear removal, the faces' emotional content showed uncertainty, ambiguity, awe, concern, or puzzlement, and not just a decrease in sadness. Sometimes, after the removal of the tears, the faces do not look sad at all and instead look neutral. Recently, the increasing appearance of virtual characters in computer games, commercials, and movies makes the facial expressions even more important. Computer graphics researchers have been greatly interested in this subject and we have seen considerable innovation in 3D facial expressions during the last decade. However, there have been no theories and methods of making dramatic or realistic animated 3D tears. In the past, the artist usually used various textures to accentuate the static tears. Since tears, which consist mostly of water, has high specular reflection and transparency, those antiquated texture tears cannot achieve this degree of realism. We thus present the methods to simulate real-time shedding of tears.

3.2 Background

3.2.1 Particle Systems

A particle system is a collection of a large number of point-like elements. Particle systems are often animated as a simple physical simulation. Because of the large number of elements typically found in a particle system, simplifying assumptions are used in both the rendering and the calculation of their motions. Various implementations require

different simplifying assumptions. Some of the common assumptions are: particles do not collide with other particles; particles do not cast shadows, except in an aggregate sense; particles only cast shadows on the rest of the environment, not on each other; particles do not reflect light - they are each modeled as point light sources.

Particles are often modeled as having a finite life span, so that during an animation there may be hundreds of thousands of particles used but only thousands are active at any one time. Randomness is introduced into most of the modeling and processing of particles to avoid excessive orderliness. In computing a frame of motion for a particle system, the following steps are taken: 1. Any new particles that are born during this frame are generated. 2. Each new particle is assigned attributes. 3. Any particles that have exceeded their allocated life span are terminated. 4. The remaining particles are animated and their shading parameters change according to the controlling processes. 5. The particles are rendered. The steps are then repeated for each frame of the animation. If the user can enforce the constraint that there are a maximum number of particles active at any one time, then the data structure can be static and particle data structures can be reused as one dies and another is created in its place.

Particles are typically generated according to a controlled stochastic process. For each frame, a random number of particles are generated using some user-specified distribution centered at the desired average number of particles per frame. The distribution could be uniform, Gaussian or anything else the animator wants. Splashing, foaming, and breaking waves are complex processes best modeled by particle systems

and volumetric techniques, but these techniques are inefficient in nonturbulent situations [71].

3.2.2 Related Work

Simulation of water in real-time, particle-based methods has become popular in recent years. Chentanez et al. [72] combined grid based and particle based approaches to demonstrate a hybrid water simulation, which included a large flowing river along a valley with beaches, big rocks, steep cliffs and waterfalls. Pfaff et al. [73] presented an algorithm to control the detail of the simulation by adjusting the number of particles without changing the large scale behavior. This is highly suitable for massively parallel architectures, and was able to generate turbulent simulations with millions of particles. Thürey et al. [74] presented a shallow water based particle model that is coupled with a smoothed particle hydrodynamics simulation to demonstrate the real-time simulations of bubble and foam. This bubble simulation interacted with a smoothed particle hydrodynamics simulation including surface tension to handle foam on the fluid surface. Yuksel et al. [75] introduced a new method for real-time simulation of fluid surface waves and their interactions with floating objects that was based on the concept of wave particles. This method was appropriate for most fluid simulation situations that did not involve significant global flow.

For the simulation of splashing fluid, jets of water, etc, using particles is likely the optimal choice. These particles are either created or initialized with meaningful positions and velocities before the simulation starts or generated during the simulation by emitters.

Particle systems simulate certain fuzzy phenomena. The parameters in the simulation are all or mostly fuzzy values. The animated tears as a part of the facial expressions need precise numeric values for its parameters. Otherwise the tears will not look realistic at all.

3.3 Generating Animated Tears

3.3.1 Dripping tears

There are two kinds of tear animations. One is the dripping tear (Figure 3.5), where the tear falls quickly to the ground, rather than falling slowly down the skin. The teardrop continually changes shape and falls as time increases. The formula we have developed to generate this kind of tear is:

$$\begin{cases} x = \dfrac{\alpha \times \sin(u)\cos(v)}{t \times \cos(u)} \\ y = \dfrac{\alpha \times \sin(u)\sin(v)}{t \times \cos(u)} \\ z = \beta \times t \times \cos(u) + \lambda \times t \end{cases}$$

α , β and λ are preset parameters. α and β describe the sizes of the tear, λ describes the speed of the tear dropping. t represents the time of animation. u and v are parameters. Figure 3.1 presents a visual representation of this phenomenon. When the teardrop is falling, time increases and the teardrop's shape continually change from left to right (Figure 3.1). Figure 3.5 is an example of dripping tears on a child's face. The example shows four frames of a running cycle.

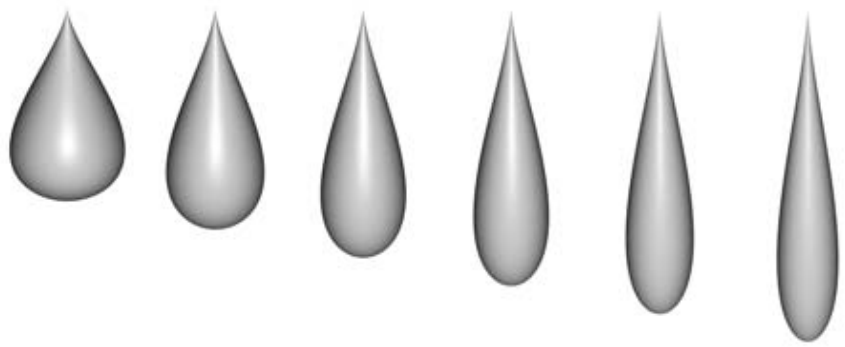


Figure 3.1: Six selected frames of a running cycle of a teardrop changing shape.

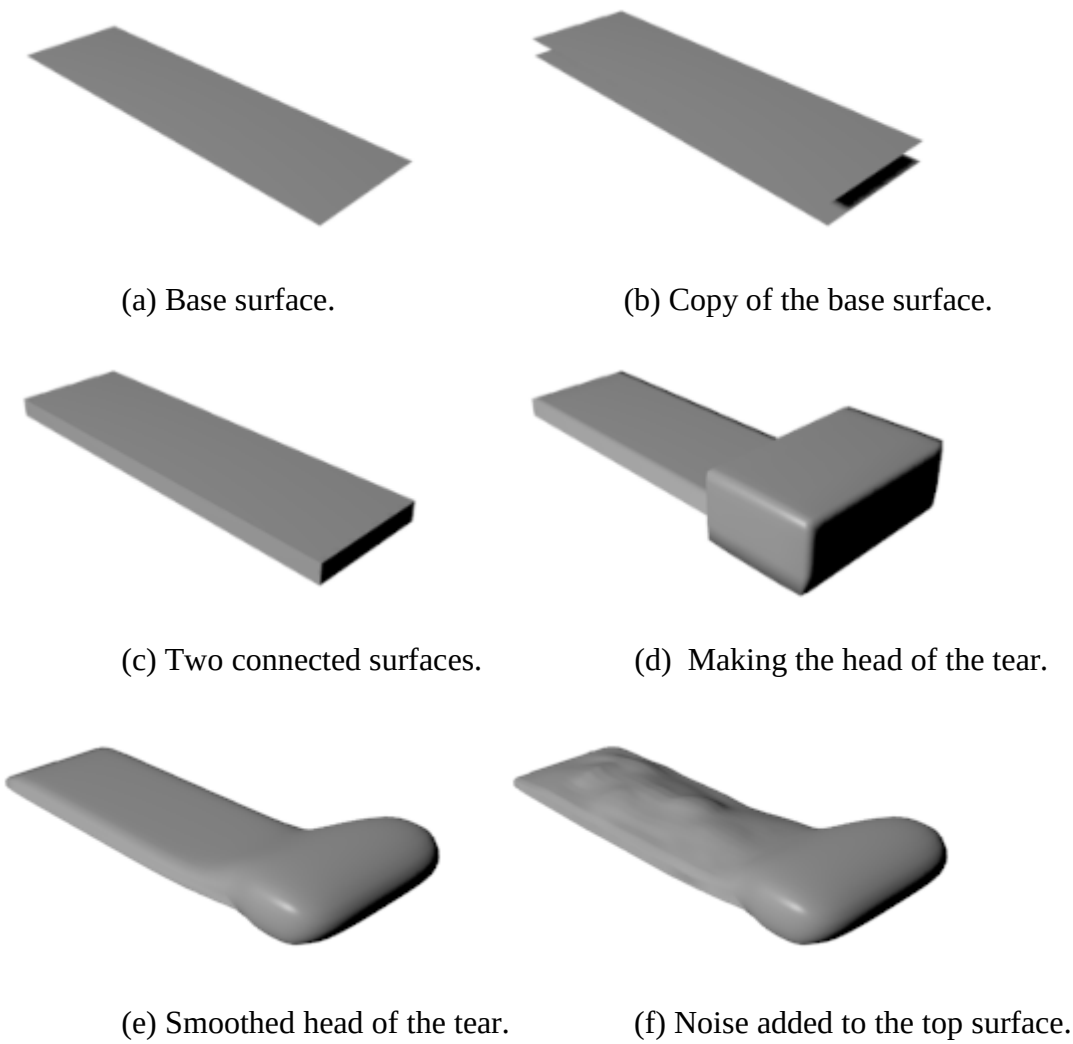
3.3.2 Shedding tears

The other type of tear is the shedding tear. This kind of tear flows along the surface of the skin on the face (Figure 3.6). The tear travels over the skin, but it is an individual object. The tear has to become seamlessly connected with the skin and will roll down as time passes. The procedure for generating shedding tears that we have created is the following:

Step one: We select the areas that the tear will pass through on the face, and then extract this area as an individual surface. Since the face mesh generally is not fine enough for detailed simulation and smoothness, we subdivide the extracted surface to refine meshes, and call it the base surface (Figure 3.2 (a)).

Step two: From the outline of the base surface, we construct the object- the tear will wrap around on the base surface. Figure 3.3 is a simple illustration. The red plane is the base surface. The black plane (top surface) is a surface parallel to the base surface (red). The remaining planes in Figure 3.3 are constructed surfaces that all connect to both parallel surfaces (red and black). Also see Figure 3.2 (b) and (c).

Step three: Figure 3.4 shows how we make the head of the tear when the tear is rolling down the face. The plane *abcd* represents a base surface that is the same as the red plane in Figure 3.3. The plane *mnpo* is the top surface and is the same as the black plane in Figure 3.3. Now, we extrude the planes *gfcp, hgpo* and *hedo*. The result is shown by the green region. The green part of the object is the head of the tear (also see Figure 3.2 (d)). Then, we smooth the head of the tear (green region), giving us the result as Figure 3.2(e).



(a) Base surface. (b) Copy of the base surface.

(c) Two connected surfaces. (d) Making the head of the tear.

(e) Smoothed head of the tear. (f) Noise added to the top surface.

Figure 3.2: Generating shedding tears.

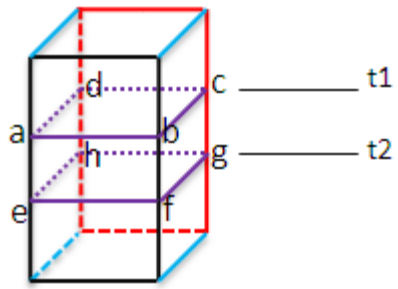


Figure 3.3: Constructed object (represents the tears).

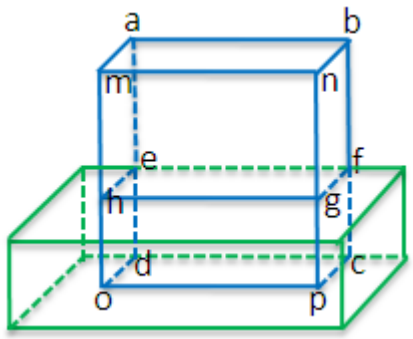


Figure 3.4: Making the head of the tear.

Step four: We randomly add noise to the tear surface to simulate the roughness of the skin. Generally, the geometric structure of the facial surface is smooth and the roughness of face skin is represented by texture. Tears have high specular reflection and transparency. With the smooth facial surface, the tear simulation will not properly achieve realism. Thus, we add random noise to the top surface (black in Figure 3.3) to reflect the facial skin roughness. In Figure 3.2(b) we see two parallel planes. It is only for demonstration. These two surfaces are seen as almost one surface in the simulation. They are very close, so we randomly choose vertices to rise from the top surface. The

vertices cannot go down from the top surface; otherwise the tear may intersect with the face surface. The density of noise we add depends on the skin's roughness. The rougher skin will need more bumps. The top surface with the noise is shown in Figure 3.2 (f).

When the noises are added to the top surface, the surface is deformed. The deformation is based on the method of moving least squares. Suppose that we have n points located at positions p_i, $P = \{p_i \in \mathbb{R}^d, d = 3\}$, $i \in \{1, 2, ..., n\}$. The surface function $f(x)$ is defined in arbitrary parameter domain Ω, which approximates the given scalar values f_i for a moving point $x \in \mathbb{R}^d$ in the MLS sense. f is taken from $\prod_r^d$, the space of polynomials of total degree r in d spatial dimensions. $F(x)$ is a weighted least squares (WLS) formulation for an arbitrary fixed point in $\mathbb{R}^d$. When this point moves over the entire parameter domain, a WLS fit is evaluated for each point individually. Then, the fitting function $f(x)$ is obtained from a set of local approximation functions $F(x)$.

$$f(x) = \arg\min_{F \in \prod_r^d} \sum_{i=1}^{n} w_i(\|x - p_i\|)\|F(p_i) - f_i\|,$$

$$F(x) = b^T(x)c(x) = \sum_{j \in [1,m]} b_j(x)c_j(x)$$

then, $f(x)$ can be expressed as $f(x) = \min \sum_i w_i(\|x - p_i\|)\left\|b^T(p_i)c(x) - f_i\right\|^2$

where, $b(x) = \left[b_1(x), b_2(x), \ldots, b_m(x)\right]^T$ is the polynomial basis vector and $c(x) = \left[c_1(x), c_2(x), \ldots, c_m(x)\right]^T$ is the vector of unknown coefficients, which we want to resolve. The number m of elements in $b(x)$ and $c(x)$ is given by $m = \frac{(d+r)!}{d!r!}$.

$w_i(\|x - p_i\|)$ is the weighting function by distance to x. $w(\theta) = \frac{1}{\theta^2 + \varepsilon^2}$. Setting the parameter ε to zero results in a singularity at $\theta = 0$, which forces the MLS fit function to interpolate the data.

Setting the partial derivatives of $f(x)$ to zero, we obtain $\frac{\partial f(x)}{\partial c_j(x)} = 2\sum_i w_i(\|x - p_i\|) b_j(p_i)\left[b^T(p_i)c(x) - f_i\right] = 0$ where $j = 1, ..., m,$ and we can get

$$\sum_i w_i(\|x - p_i\|) b(p_i) b^T(p_i) c(x) = \sum_i w_i(\|x - p_i\|) b(p_i) f_i$$

If the matrix $A = \sum_i w_i(\|x - p_i\|) b(p_i) b^T(p_i)$ is not singular, i.e., its determinant is not zero, then $c(x) = A^{-1}(\sum_i w_i(\|x - p_i\|) b(p_i) f_i)$.

As a result, $f(x) = b^T(x) A^{-1}(\sum_i w_i(\|x - p_i\|) b(p_i) f_i)$

Step five: For the animation of shedding tears, we assume Figure 3.3 is a path of tear flowing from the top of the object. When time starts (t_1), we select a certain portion of the vertices (see Figure 3.3, the vertices above the plane *abcd*) on the top side of the object. As time passes (t_2), the selected portion of the object increases in size (see Figure 3.3, the vertices above the plane *efgh*). For each time point, there is a corresponding set of vertices. We use the height of the tear path to indicate the covered portion of the vertices. Height is $h = s + c \times t,$ where s and c are constant, and t is time.

We used the above five-step procedure to make an example (Figure 3.6) of shedding tears that flow along the skin surface. The example shows four frames of a running cycle.

3.4 Summary

We have presented a novel approach for generating realistic, animated tears. We have introduced a formula for generating teardrops which continually change shape as the tear drips down the face. We also show the method and procedures for generating shedding tears. We extract the tear's path from the facial surface, and construct the three dimensional tear's object that is based on tear stains. We randomly add noise to the tear surface to simulate the roughness of the skin. We extrude and smooth the head of the moving tear to create real-time vivid rolling of the tears down the face. Our results indicate that these methods are robust and accurate. Our methods both broaden CG and increase the realism of facial expressions. It will evolve 3D facial expression and animations significantly. Our methods can also be applied to CG human body skins and other surfaces to simulate flowing water. Our method is not only for animation by a trained artist, but also for industry.

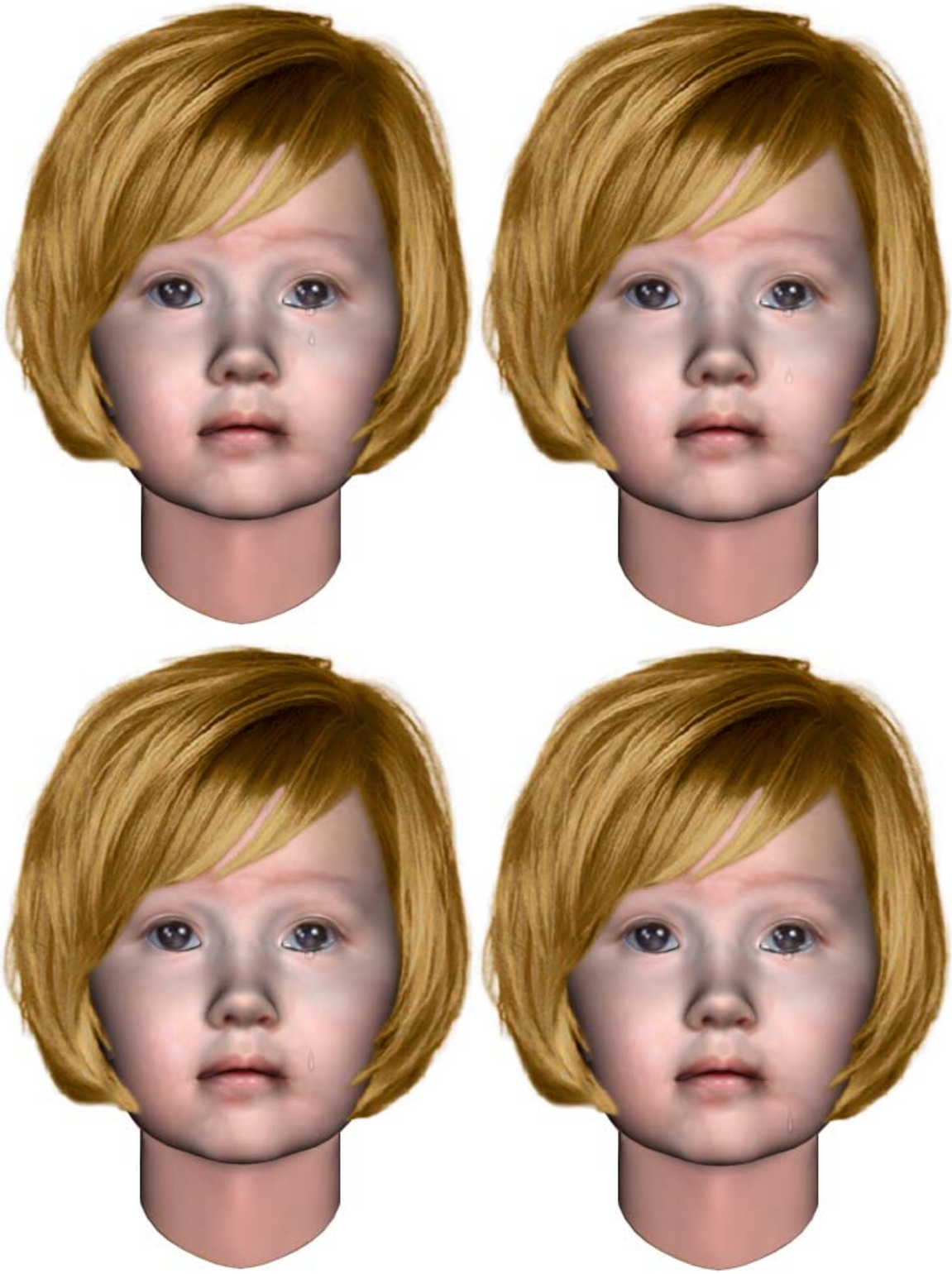


Figure 3.5: Four selected frames of a running cycle (dripping tears).

Figure 3.6: Four selected frames of a running cycle (shedding tears).
Copyright © Alice J. Lin 2011

Chapter 4 Automatically Placing Bones

4.1 Introduction

Recent years have seen an explosion in 3D computer generated animation. The entertainment industry has played a crucial role, pushing computer graphics technology to the limit. Studios producing computer animated films are increasing every year, while digital characters are becoming commonplace in traditional films. Most of the top grossing films in the United States have scenes with some form of computer animation. Animation is an important ingredient in computer games as well. The computer gaming industry is one of the fastest growing industries in recent times. This high demand has resulted in a need for improving the efficiency of the computer animation process.

One process is rigging. Rigging is the process of defining and implementing the possible motions of a virtual character and providing controls to execute them. This is a complex and time-consuming procedure requiring both artistic and technical savvy, especially when dealing with the human face, which is the most expressive part of the body and communicating both language and emotion. Human beings have developed an acute sensitivity to facial expressions and artifacts in facial movement are immediately noticeable. Additionally, although humans all share the same underlying facial structure, the range of face types is immense. Current facial animation techniques, for example, parameterized models, provide ready-made, configurable and animatable faces, but the numbers of characters, which can be represented, are limited by the number of conformal parameters. Morphing systems, on the other hand, can represent virtually any character,

but they depend on facial expression libraries, which must be built for each character. Additionally, these popular morphing algorithms do not simulate the motions in the face correctly.

Building a facial model rig is an essential element for animating CG characters in feature films and computer games. Due to their intuitive controls and art-directable nature, facial rigs are particularly popular among artists for creating realistic looking facial animations. The process of rigging is tedious and requires many hours of manual work for an artist. The common skinning approach requires manually define the location of the joints (bones) of the skeleton, such as left of the right eyebrow, top of the lip, etc. on the mesh. Then the artist needs to specify which parts of the surface are attached to which joint(s). More sophisticated models based on simulation of the facial tissues and muscles also need human intervention to attach tissues to the skeleton and to tune complex muscle configuration.

In this work, we intend to increase the productivity of professional artists and allow even inexperienced users to quickly generate animated models. We present a method for the automated placement of bones on human face models, allowing the artist to immediately concentrate on the animation phase, providing significant time savings for this arduous task. Our approach is to group the vertices on the surface of a model which is optimized to support the deformations of a human face. Our method works on a particular person's face and it allows the artist to control the shape of the skin deformations quickly and effectively.

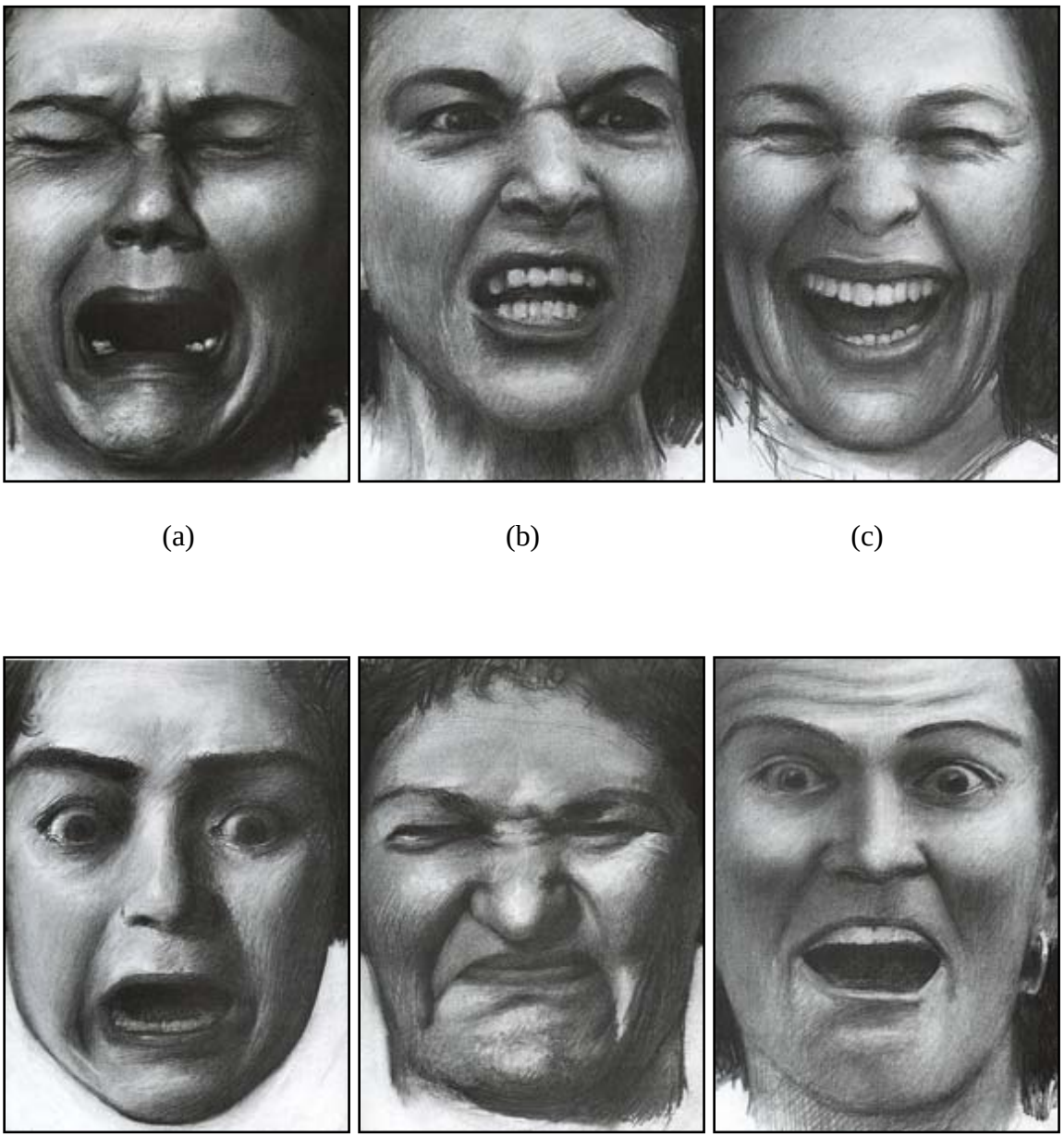


Figure 4.1: (a) Sadness (b) Anger (c) Joy (d) Fear (e) Disgust (f) Surprise

4.1.1 Facial Rigging

The animator's task is to manipulate the model's controls to bring the face to life. A rich, interactive environment that supports the animator in this activity is highly desired. Interactive animation tools consist primarily of motion control tools and motion preview tools. Facial rigging is the process of creating the animation controls for a facial model and the animator's interface to those controls. One early example is a system developed by Hanrahan and Sturman [76], which allowed the animator to establish functional relationships between interactive input devices and control parameters. Modern interactive animation systems provide a wide variety of ways for linking user interactions to animation actions.

One widely used approach is to embed within the face model an articulated joint hierarchy. Each joint corresponds to the articulated connection between two skeletal segments or bones. These joints are manipulated by varying their orientation angles. For the face, this joint structure might consist of several joints for the neck, connected to a joint for the jaw and a joint for the skull. The joint for the skull might in turn be connected to joints for the eyeballs. The joint for the jaw might in turn be connected to a joint for the tongue. Influences are established between the surface vertices or control points and one or more of these joints. As the joints are manipulated and their orientation angles adjusted, the associated surface points are rotated along with the joints. The influence between a specific surface point and a given joint is usually a weighting value. Some surface points close to the joint might be completely influenced by it, moving exactly as it moves. Other points might be only partially influenced by a joint, moving

only a percentage of its motion. Each surface point is likely to have a different weight or influence from a given joint. Some points on the surface may be influenced by several joints. Establishing the joint structure and determining the surface point weighting values for the joints is a part of the rigging process. Examples of this approach can be seen in Osipa [77].

Another widely supported approach is the use of blend shapes. Blend shapes are a form of surface shape interpolation. Here the surface is sculpted into two or more shapes. One of these is the base shape, while the others are called target shapes. The differences between the base and target shapes are represented as vector sets. Each vector set corresponds to the difference between the base shape and one of the target shapes. Each vector in a blend shape is the difference in location between a point in the base shape and its corresponding point on the target shape. When a blend shape is applied to the base shape, it takes on the target shape for that blend. If the blend shape is only partially applied, the base shape moves partially toward the target shape. This allows interpolation between the base and target shapes. This interpolation is controlled by a blending coefficient. If the coefficient is 0.0, the surface has the base shape. If the coefficient is 1.0, the surface takes on the target shape. For values between 0.0 and 1.0, the surface takes on a shape between the base and target shapes.

For a given blend shape, perhaps only a few of the surface points are affected. That is, for a given blend shape, some of the surface points may have zero length vectors. This means that the positions for these points are the same in both the base and target shapes.

Also, a given surface point may be affected by more than one blend shape. If more than one blend shape is applied to a surface, each surface point will be moved, based on the sum of the point vectors for each blend shape, weighted by their respective coefficients.

4.1.2 Facial Expressions

Research in facial expression has concluded that there are six universal categories of facial expressions that are recognized across cultures [67]. These categories are sadness, anger, joy, fear, disgust, and surprise, as shown in Figure 4.1. Within each of these categories there may be a wide range of expression intensity and variation in expression details.

Faigin [68] presented an excellent discussion, from an artist's point of view, of these expressions and their variations. Faigin described each expression category and its variations in terms of the appearance of three facial regions. The three expressive regions are the eyebrows, the eyes, and the mouth.

Sadness: In simple sadness, the inner portions of the eyebrows are bent upward. The skin and soft tissue below the eyebrow are piled up above the upper eyelid. The eyes are slightly closed because of downward pressure from the tissue above the eyelid and upward motion of the lower eyelid. In simple sadness, the mouth is relaxed. Sadness is illustrated in Figure 4.1(a). Sadness has many intensities and variations, including open-mouthed crying, closed-mouth crying, suppressed sadness, nearly crying, and looking

miserable. These variations may include completely lowered eyebrows, tightly shut eyes, a square-shaped open mouth, a bulge on the chin, and a very pronounced nasolabial fold.

Anger: In simple anger the inner corners of the eyebrows are pulled downward and together. The lower edge of the eyebrow is at the same level as the upper eyelid. The eye is wide open, but pressure from the lowered brow prevents the white of the eye from showing above the iris. The mouth is closed, with the upper lip slightly compressed or squared off. Anger is illustrated in Figure 4.1(b). Variations of anger include shouting rage, rage, and sternness. These variations may include tightly compressed lips with a chin bulge, or an open mouth with a sneering upper lip and a straight lower lip, showing both the upper and lower teeth.

Joy: In simple joy, the eyebrows are relaxed. The upper eyelid is lowered slightly and the lower eyelid is straight being pushed up by the upper cheek. The mouth is wide, with the corners pulled back toward the ears. If the mouth is closed, the lips are thin and pressed tight against the underlying bone. If the mouth is open, the upper lip is straight, showing the upper teeth; the lower lip is straight in the middle and angled near the corners. Joy is illustrated in Figure 4.1(c). Variations on joy include uproarious laughter, laughter, open-mouthed smiling, smiling, stifled smile, melancholy smile, eager smile, ingratiating smile, sly smile, debauched smile, closed-eye smile, false smile, and false laughter. False smiles and false laughter are indicated by diminished crow’s feet at the corners of the eyes and by only slight or absent folds under the lower eyelids.

Fear: Fear can range from worry to terror. In fear, the eyebrows are raised and pulled together. The inner portions of the eyebrows are bent upward. The eyes are alert. The mouth may be slightly dropped open and stretched sideways. Fear is illustrated in Figure 4.1(d). In worry, the lips are squeezed tightly together, and the lip margins disappear. There is a bulging below the lower lip and over the chin. In terror, the eyes and mouth are wide open. The upper lip is relaxed, while the lower lip is stretched wide and tight, exposing the lower teeth. The nasolabial fold becomes straight and shallow. Bracket shaped folds appear to the sides of the lower lip.

Disgust: Disgust ranges from disdain to physical repulsion. In disgust, the eyebrows are relaxed. The eyelids are relaxed or slightly closed. The upper lip is raised into a sneer, often asymmetrical. The lower lip is relaxed. The nasolabial fold is deepest alongside the nose. Disgust is illustrated in Figure 4.1(e). In disdain, the eyelids may be partly closed, with the eyes looking down. For physical repulsion, the eyebrows are lowered, especially at the inner corners. The eyes may be mostly shut in a squint. The upper lip is raised in an intense sneer, which may show the upper teeth. The lower lip is slightly pushed up.

Surprise: In surprise, the eyebrows are raised straight up, as high as possible. The upper eyelids are open as wide as possible, with the lower eyelids relaxed. The mouth is dropped open without muscle tension to form an oval shape. In surprise, horizontal folds are formed across the brow. Surprise is shown in Figure 4.1(f).

There are many expressions that are not directly related to emotion. Faigin described additional expressions associated with physical states such as pain and sleepiness. Faigin asserted that many of these expressions are also universally recognized. Faigin's list includes pain, exertion, drowsiness, yawning, sleeping, singing, shouting, passion, intensity, attention, perplexity, shock, and the facial shrug.

4.1.3 Rigging and Skinning

Rigging is the process of building controls for the movement of a character or an object. This stage cannot start until the character model is finished. The character rig consists of a hierarchical bone structure that can mimic a real, anatomical skeleton. Unlike real bones, however, CG bones are not necessarily rigid; thus, they can be used to simulate muscles as well. Special controls and constraints can be applied to each bone to ensure that they move realistically. Constraints can be applied to any part of the rig for the animator to use. The goal of the rigging phase is to provide intuitive controls that can be manipulated to achieve all the movements and poses desired by the animators.

Skinning defines how the model will deform in relation to the rig. A subset of the model's points is associated with each bone in the skeleton. In a rigid skin, each point is controlled by only one bone. In a smooth skin, two or more bones can influence a point; thus, the influence of each bone is weighted. Often, skinning alone does not achieve the desired deformation effect. Additional deformers can be applied on top of the skin to undo any undesirable creasing or pinching that result from skinning. Since skinning depends on the geometry of the model, it cannot begin until the model is completed.

4.1.4 Motion Capture

It is often easier to record movement of a real object and map it onto a synthetic object than it is for the animator to craft realistic-looking motion. Motion capture involves sensing, digitizing, and recording an object's motion [78]. In motion capture, a (human) subject's movement is recorded. It should be noted that locating and extracting a figure's motion directly from raw (unadulterated) video is extremely difficult and is the subject of current research. As a result, the figure whose motion is to be captured is typically instrumented in some way so that positions of key feature points can be easily detected and recorded. There are primarily two approaches to this instrumentation: electromagnetic sensors and optical markers.

Electromagnetic tracking, also simply called magnetic tracking, uses sensors placed at the joints that transmit their positions and orientations back to a central processor to record their movements. While theoretically accurate, magnetic tracking systems require an environment devoid of magnetic field distortions. To transmit their information, the sensors have to use either cables or wireless transmission to communicate with the central processor. The former requires that the subject be "tethered" with some kind of cabling harness. The latter requires that the subject also carry a power source such as a battery pack. The advantage of electromagnetic sensors is that the three-dimensional position and orientation of each sensor can be recorded and displayed in real time (with some latency). The drawbacks relate to the range and accuracy of the magnetic field and the restricted movement resulting from the instrumentation required.

Optical markers, on the other hand, have much larger range, and the performers only have to wear reflective markers on their clothes. The optical approach does not provide real-time feedback, however, and the data from optical systems are error prone and noisy. Optical markers systems use video technology to record images of the subject in motion. Because orientation information is not directly generated, more markers are required than with magnetic trackers. Some combination of joint and mid-segment markers is used. While industrial strength systems may use fast infrared cameras, basic optical motion control can be implemented with consumer-grade video technology.

Motion capture is a very powerful and useful tool. It will never replace the results produced by a skilled animator, but its role in animation will expand and increase as motion libraries are built and the techniques to modify, combine, and adapt motion capture data become more sophisticated.

4.2 Related Work

Rigging is the process of taking a static, inanimate computer model and transforming it into a character that an animator can edit frame-by-frame to create motion [79]. The result is a rig that can be manipulated by a set of controls like a virtual puppet [80] or by motion capture data. Creating a character rig is a very complex, time-consuming and labor intensive task. Still, there is no defined standard methodology for rigging a face. Studios continue to redefine the techniques, processes, technologies and production pipelines to efficiently create films and videogames. Today, facial animation is done manually by skilled artists, who carefully place and manipulate the animation controls to

create the desired motion. As models become more and more complex, it is increasingly difficult to define a consistent rig that can work well for many different characters. Thus, each facial rig has to be individually created by hand. This traditional method ensures high quality results, but is slow and costly. It takes an experienced digital artist for weeks to create a complete facial rig, depending on its complexity. But if any change must be applied to an already created rig, the rigging process has to be restarted. Facial rigging becomes a serious bottleneck in any CG production.

A rig is a set of controls that allows an artist to manipulate a character. The character rigging process is analogous to setting up the strings that control a puppet, which in the hands of an experienced digital artist "comes to life" [81]. After having been rigged, a character's shape can be controlled through a set of abstract parameters. For each key frame, instead of having to position each vertex of the surface mesh, the animator needs only to set the values of the control parameters.

In production environments, facial rigging is commonly done by geometric deformations, blendshapes, or combination of the two. Geometric deformations are driven by simulated muscle actions, which are loosely based upon the dynamics of facial tissue [7]. Blendshapes interpolate a large number of sculpted shapes. Skeleton-based rigs are most often employed for full-body animation due to their intuitive control for articulated motion. While skeletons are often used for facial animation of cartoon characters, this approach is less suited to produce detailed facial expressions that exhibit wrinkles and folds.

Many methods have been devised for the geometric deformation of surfaces, including free-form deformation [82], shape interpolation [83], and wire deformation [84]. Such methods form the building blocks for rigging in modern animation programs. The work of Kry et al. [85] used simulated input deformations to produce hand deformations.

The techniques generating realistic skinned animation from skeletal motion can be roughly categorized into three classes: skeleton subspace deformations, physics-based deformations, and data-driven deformations. Skeleton subspace deformations attach skin to bones and each vertex of the skin is transformed based on a weighted transformation of the neighboring bones. This technique is fast to compute and easy to implement.

Almost any system for mesh deformation can be adapted for skeleton-based deformation. Teichmann and Teller [86] proposed a spring-based method. Unfortunately, at present, this method is unsuitable for real-time animation of even moderate size meshes. Because of its simplicity and efficiency, despite its quality shortcomings, linear blend skinning, also known as skeleton subspace deformation remains the most popular method used in practice. Most real-time skinning work, e.g. [85, 87], has focused on improving linear blend skinning by inferring the character articulation from multiple example meshes.

Baran and Popovic [88] proposed skeleton embedded rigging for full-body animations of Pinocchio. Kahler et al. [89] fitted a complex anatomical model to partial

3D scan data. Orvalho et al. [90] introduced a general rigging method by transferring a generic facial rig to 3D input scans and hand-crafted models.

4.3 Automatic Placement of Bones on Face/Head

Depending on how realistic the result is meant to be and what kind of control the animator is provided, there are a wide range of approaches to model and animate the face. Results vary from cartoon faces to parameterized surface models to skull-muscle-skin simulations. Realistic facial animation remains one of the interesting challenges in computer animation.

Bone-based rigs are most often employed for full-body animations due to their intuitive control for articulated motion. Currently, it requires manually setting bones to specify its internal skeletal structure. Since both the body and limbs have articulations, it is not hard to attach bones to the surface; one only needs to set the bones to the skeleton's position. The human face has an underlying skeletal structure and one main skeletal articulatory component - the jaw. In addition to the jaw, the other rigid articulatory components are the eyes. The surface of the skull is covered with muscles, most of which connect areas of the skin to positions on the skull. These muscles tug on the skin to create movement, often recognizable as expressions. Features of the skin include eyelids, mouth, and eyebrows. Because the face only has a single mesh, it is very hard to manually place bones on the surface of the face to control facial expressions and animate it. Until now, there is no automation to set the bones for each individual face. We present a novel method to auto group the vertices on the face and place the bones in proper positions.

To automatically set the bones in the face/head, we divide vertices into groups that capture the "natural" structure of the vertices and place the bones in the centroid of the groups. We group vertices based on information found in the vertices that describe the face/head as well as relationships between each part of the face/head. The goal is that the vertices within a group have high similarity but are very dissimilar to vertices in other groups. Dissimilarities are assessed based on the attribute values describing the face/head and the distance measured. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the group. The controls of facial animation are based on human facial structures. We can decompose it into three components: the facial mask, the eyes, and the tongue. For the eyes, it is easy to place the bone in the center of each eyeball to control the eye movements. The tongue has simpler movement, so we will not discuss that here.

4.3.1 Placing the Head Bones

Beginning with the head model, we group the head vertices into three classes – the face part (facial mask), the middle part and the back of the head part, see Figure 4.2. The vertices within a group are related to one another and unrelated to the vertices in other groups. We set the head width as the x direction, height as the y direction and depth as the z direction. We group the head in its depth and specify the desired number – three groups as a termination condition.

We start with individual vertices as groups by placing each vertex into its own group, then successively merging the two closest groups into larger and larger groups until the

termination conditions are satisfied. If you are not sure how many groups you need, you can merge the two closest groups until a single, all-encompassing group remains. Then you can go back to see the operation of each merge to choose the group number. The result is shown in Figure 4.2. We will further group vertices in the red area – facial mask which excludes the neck vertices. Since the blue region (back of the head) is not associated with facial expressions, we will not need to group it in detail. We group the green (middle) part again with the same method and two groups as a condition. The head bone passes the mean (upper green part) in the y direction. The neck bone passes the mean (lower green part) and parallels the neck direction. We place two bones (see Figure 4.3) connected together through these two means to control the head animation.

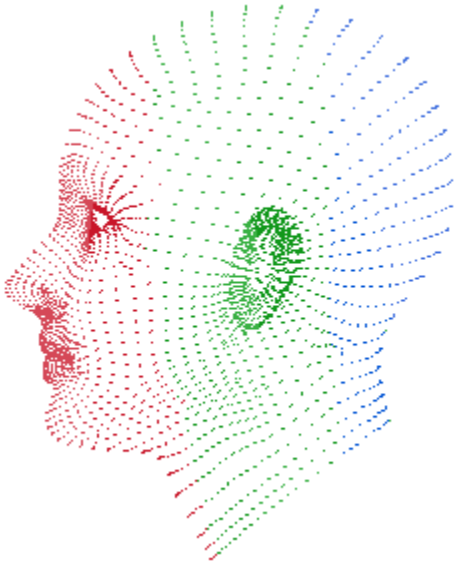


Figure 4.2: Grouping the head vertices.

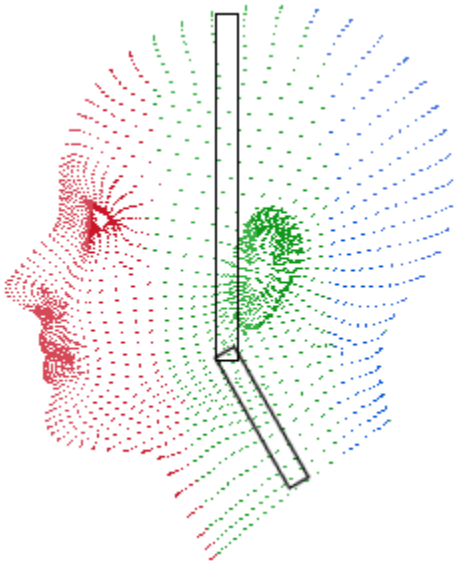


Figure 4.3: Setting head bones.

The distance between two groups uses the average distance between pairs of vertices. It tends to join groups with small variances. Thus, it is a compromise between the minimum and maximum distances and overcomes the outlier sensitivity problem. The proximity of two groups is the average pairwise proximity (average length of edges) among all pairs of vertices in different groups (Figure 4.4). The group proximity $d_{avg}(C_i, C_j)$ of groups C_i and C_j, which are of size n_i and n_j, respectively, where $|v - v'|$ is the distance between two vertices, v and v', is expressed by the following equation:

$$d_{avg}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{v \in C_i} \sum_{v' \in C_j} |v - v'|$$

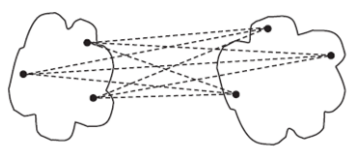


Figure 4.4: The average pairwise proximities of all pairs of vertices from different groups.

The following is a demonstration of this. Figure 4.5 illustrates how grouping works. (a) A set of two-dimensional points consists of five points. We specify that the desired number of groups is two as a termination condition. Table 4.1 shows the points and the Euclidean distance between the points. Start with the points as individual groups. (b) We average the pairwise proximities of all pairs of two points from different groups, and then merge the closest two groups. (c) Update the proximity to reflect the proximity between the new group and the original groups. We then calculate the distance between groups.

Distance between groups {3, 4} and {1} = (0.22 + 0.39)/(2∗1) = 0.31

Distance between groups {2, 5} and {1} = (0.26 + 0.34)/(2∗1) = 0.30

Distance between groups {3, 4} and {2, 5}

= (0.15 + 0.28 + 0.20 + 0.29)/(2∗2) = 0.23

Since the distance between groups {3, 4} and {2, 5} is smaller than the distance between groups {3, 4} and {1} and the distance between groups {2, 5} and {1}, groups {3, 4} and {2, 5} are merged (see Figure 4.5 (c)).

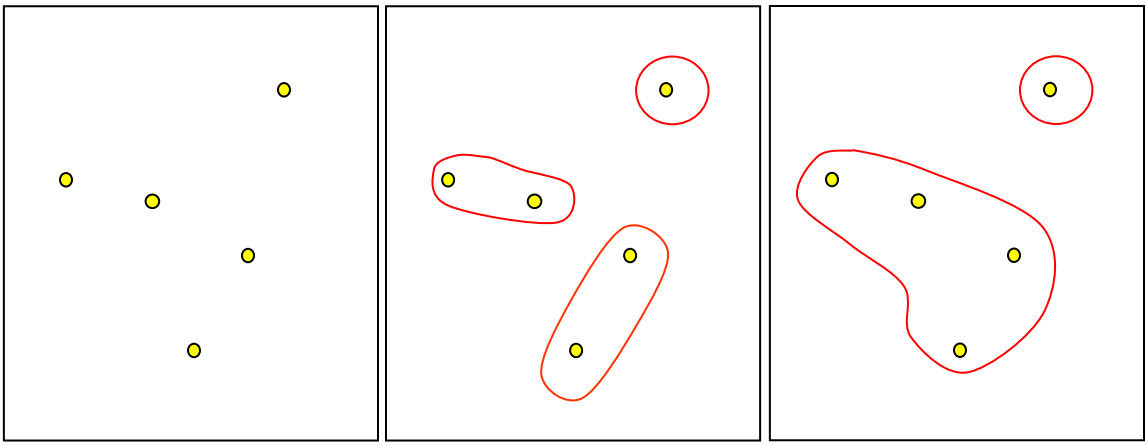
	P1	P2	P3	P4	P5
P1	0.00	0.26	0.22	0.39	0.34
P2	0.26	0.00	0.15	0.20	0.14
P3	0.22	0.15	0.00	0.15	0.28
P4	0.39	0.20	0.15	0.00	0.29
P5	0.34	0.14	0.28	0.29	0.00

Table 4.1: Euclidean distances between five points.

4.3.2 Multi-density Setting of the Facial Bones

The facial mask accounts for most of movement in the face. It is responsible for simulating facial skin (Figure 4.10), opening/closing the eyelid (Figure 4.13) and rotating the jaw (Figure 4.11, Figure 4.12). Therefore, it is the most complex component of the group vertices.

A group is a collection of vertices that are similar to one another within the same group but are dissimilar to the vertices in other groups. We take the input parameter and



(a) (b) (c)

Figure 4.5: The results of the grouping approach for the set of five points. (a) Initially, we place each point into a group of its own. (b) The groups are merged into three groups according to the criterion. (c) The group merging process repeats until the specified desired number of groups (two) as a termination condition is reached.

partition a set of vertices into groups so that the resulting intragroup similarity is high but the intergroup similarity is low. Group similarity is measured in regard to the *mean* value of the vertices in a group, which can be viewed as the group's centroid or center of gravity. For each group, we find a point that is the centroid of the group vertices. The bone is placed in the centroid to control the group's vertices. First, we estimate the number of bones we need in the face. Then, we attempt to find the number of groups, which are represented by their centroids. Different repeating group times will generate different group sizes. This means we can have multi-density bone distributions according to the requirements. Figures 4.6 and 4.7 demonstrate the results of our method. The black squares represent the position of the bones. Figures 4.6 and 4.7 show two densities of bone placement. We connect the facial bones to the head bone for children. All facial

bones follow as the head bone moves. Figure 4.8(a) shows such an example. Figure 4.8(b) shows that the head bone controls head rotation.

We first randomly select *k* vertices from the data set, each of which initially represents a group mean or center, where *k* is a specified parameter, namely, the number of groups desired. The initial centroids are chosen from dense regions so that they are well separated. For each of the remaining vertices, a vertex is assigned to the group to which it is the most similar, based on the distance between the vertex and the group mean. For each vertex, we calculate the distance to all centroids and find the minimum distance. This vertex is said to belong to the group that has the minimum distance from this vertex. Each vertex is assigned to the closest centroid, and each collection of vertices assigned to a centroid is a group. The centroid of each group is then updated based on the vertices assigned to the group. Since we are not sure about the location of the centroid, we need to adjust the centroid location based on current updated data. Then we assign all the data to this new centroid.

In the course of the iterations, we try to minimize the sum over all groups of the squared within-group residuals, which are the distances of the vertices to the respective group centroids. Convergence is reached when the objective function (i.e., the residual sum-of-squares) cannot be lowered any more. In other words, the groups obtained are such that they are geometrically as compact as possible around their respective centroids. This process iterates until the criterion function (square-error criterion) converges, i.e. no point is moving to another group anymore, and the centroids remain the same.

To assign a vertex to the closest centroid, we need a proximity measure that quantifies the notion of "closest" for the specific vertex under consideration. The goal of the grouping is typically expressed by an objective function that depends on the proximities of the vertices to the group centroids; e.g., minimize the squared distance of each vertex to its closest centroid. For our objective function, which measures the quality of a grouping, we use

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left| v - \mu_i \right|^2 ,$$

where E is the sum of the squared error for all vertices in the data set; v is the vertex in the space representing a given object; and μ_i is the mean of group C_i (both v and μ_i are multidimensional). In other words, for each vertex in each group, the distance from the vertex to its group center is squared, and the distances are summed. This criterion tries to make the resulting k groups as compact and as separate as possible.

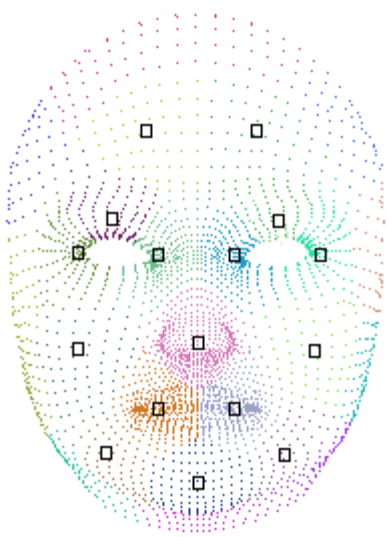


Figure 4.6: Sparse bone placements.

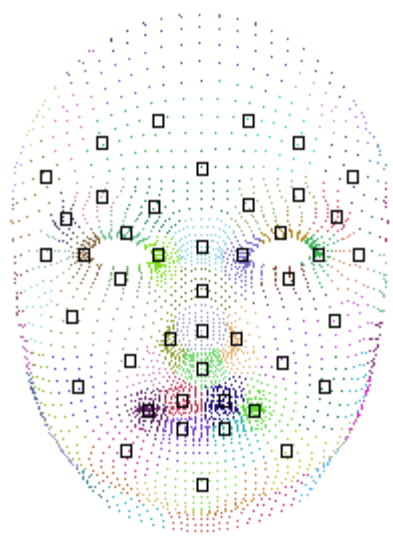


Figure 4.7: Dense bone placements.

We calculate the error of each vertex, i.e., its Euclidean distance to the closest centroid, and then compute the total sum of the squared errors. Given that two different sets of groups are produced by two different runs, we prefer the one with the smallest squared error because the centroids of this grouping are a better representation of the vertices in their group.

We seek to minimize $Error = \sum_{i=1}^{k} \sum_{\vec{v} \in C_i} \left\| \vec{v} - \vec{c}_i \right\|^2 = \sum_{i=1}^{k} \sum_{\vec{v} \in C_i} \sum_{j=1}^{d} (v_j - c_{ij})^2$

where $\vec{v}$ is a vector (point) and $\vec{c}_i$ is the "center" of the group, d is the dimension of $\vec{v}$ and $\vec{c}_i$, and v_j and c_{ij} are the components of $\vec{v}$ and $\vec{c}_i$.

We can solve for the pth group, $\vec{c}_p$, by solving for each component, $c_{pk}, 1 \le k \le d$ by differentiating the Error, setting it to 0, and solving as the following equations indicate.

$$\frac{\partial}{\partial c_{pk}} Error = \frac{\partial}{\partial c_{pk}} \sum_{i=1}^{k} \sum_{\vec{v} \in C_i} \sum_{j=1}^{d} (v_j - c_{ij})^2$$

$$= \sum_{i=1}^{k} \sum_{\vec{v} \in C_i} \sum_{j=1}^{d} \frac{\partial}{\partial c_{pk}} (v_j - c_{ij})^2 = \sum_{\vec{v} \in C_p} 2 * (v_k - c_{pk}) = 0$$

$$n_p c_{pk} = \sum_{\vec{v} \in C_p} v_k$$

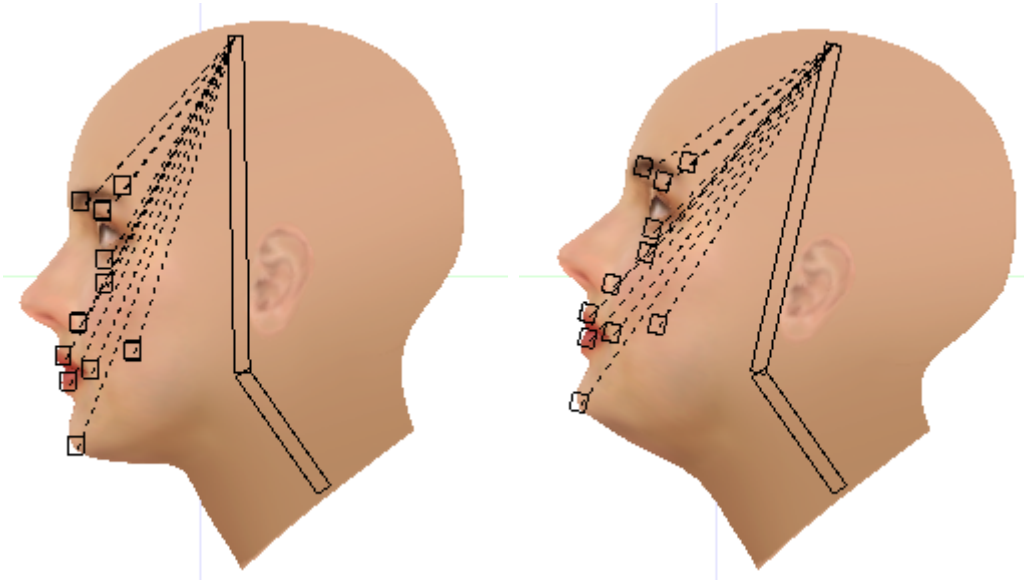
$$c_{pk} = \frac{1}{n_p} \sum_{\vec{v} \in C_p} v_k$$

It is shown that the centroid that minimizes the error of the group is the mean.

$\vec{c}_p = \frac{1}{n_p} \sum_{\vec{v} \in C_p} \vec{v}$. $\vec{c}_p$ is the mean of the vertices in the group.

We illustrate this with an example (Figure 4.9). Suppose that there is a set of vertices located in space as depicted in the rectangle shown in Figure 4.9 (a). We would like the vertices to be partitioned into two groups. We arbitrarily choose two vertices as the two

initial group centers, where group centers are indicated by red dots, as shown in (b). Each vertex is distributed to a group based on the group center to which it is closest. Such a distribution forms silhouettes encircled by curves, as shown in (c). Next, the group centers are updated in (d). That is, the mean value of each group is recalculated based on the current vertices in the group. Using the new group centers, the vertices are redistributed to the groups based on which group center is the closest. Such redistribution forms new silhouettes encircled by curves, as shown in (e). This process iterates, leading to (f). The process of iteratively reassigning objects to groups to improve the partitioning is referred to as iterative relocation. Eventually, no further redistribution of the objects in any group occurs, and the process terminates. The resulting groups are returned by the grouping process.



(a) (b)

Figure 4.8: (a) Relationship of head bone with facial bones. (b) The head bone controls the head rotation.

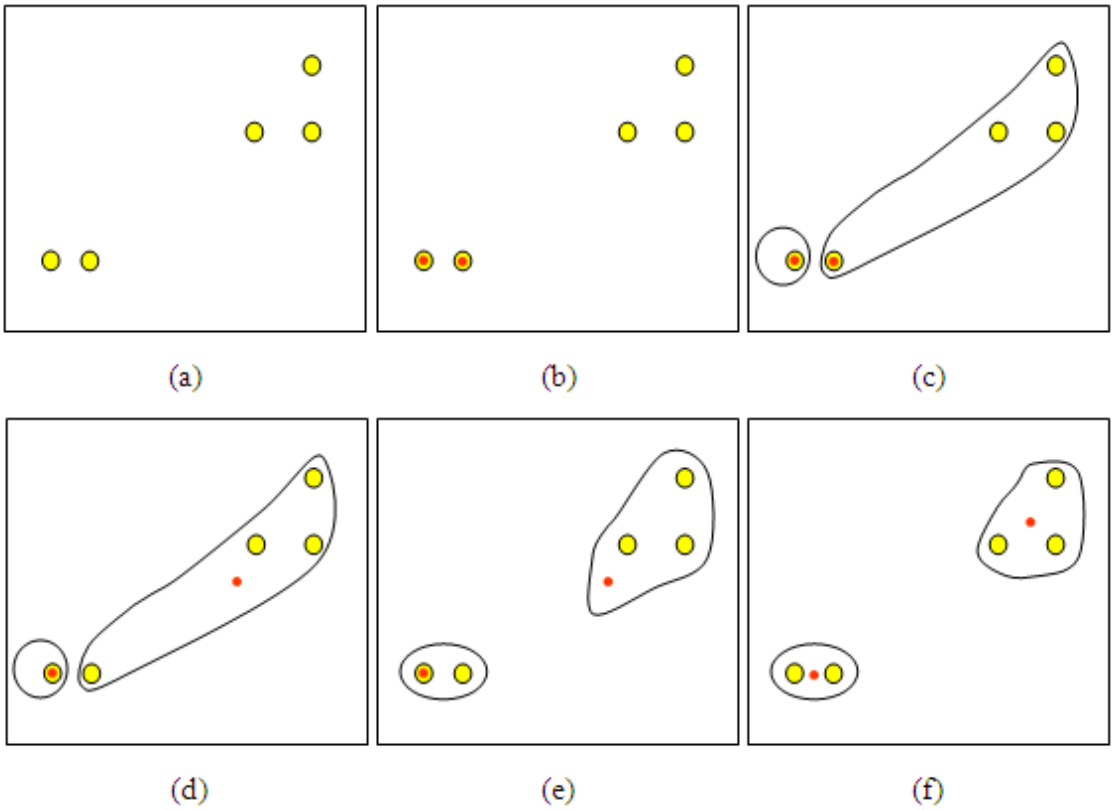


Figure 4.9: The operation of identifying the natural groups of vertices. (a) A set of vertices. (b) Two vertices as single-element groups, as shown in red. (c) Assign each of the remaining (three) vertices to the group with the nearest centroid. (d) After each assignment, recompute the centroid of the gaining group. (e) Take each vertex in sequence and compute its distance from the centroid of each of the groups. If a vertex is not currently in the group with the closest centroid, switch this vertex to that group. (f) Update the centroid of the group gaining the new vertex and the group losing the vertex. The convergence is then achieved.

4.4 Summary

We have presented a method for automatically placing bones on facial models to speed up the rigging process of a human face. We accomplish this by grouping the vertices that have natural structures of an expressive face, and then place the bones in the centroid of each group. The resulting facial deformations and animations are realistic and expressive.

Figure 4.10 is an example. It is a set of facial expressions which includes common expressions: happiness, anger, sadness, fear, disgust, and surprise. Figure 4.11 shows selected frames from an animation of chewing. Figure 4.12 shows selected frames of a running cycle of chewing. Figure 4.13 shows selected frames of a running cycle as the eyes close.

Our goal is to reduce the amount of time required to set the bones for generating realistic facial animations of different characters. Our method works on a particular person's face quickly and effectively, and provides a powerful means of generating facial expressions and animations. The time saved with our method is significant. Although we have focused entirely on the human face in this work, we believe the concepts presented are flexible enough to be used on non-human characters. Our method, in conjunction with existing techniques, allows users to go from a static mesh to an animated character quickly and effortlessly.

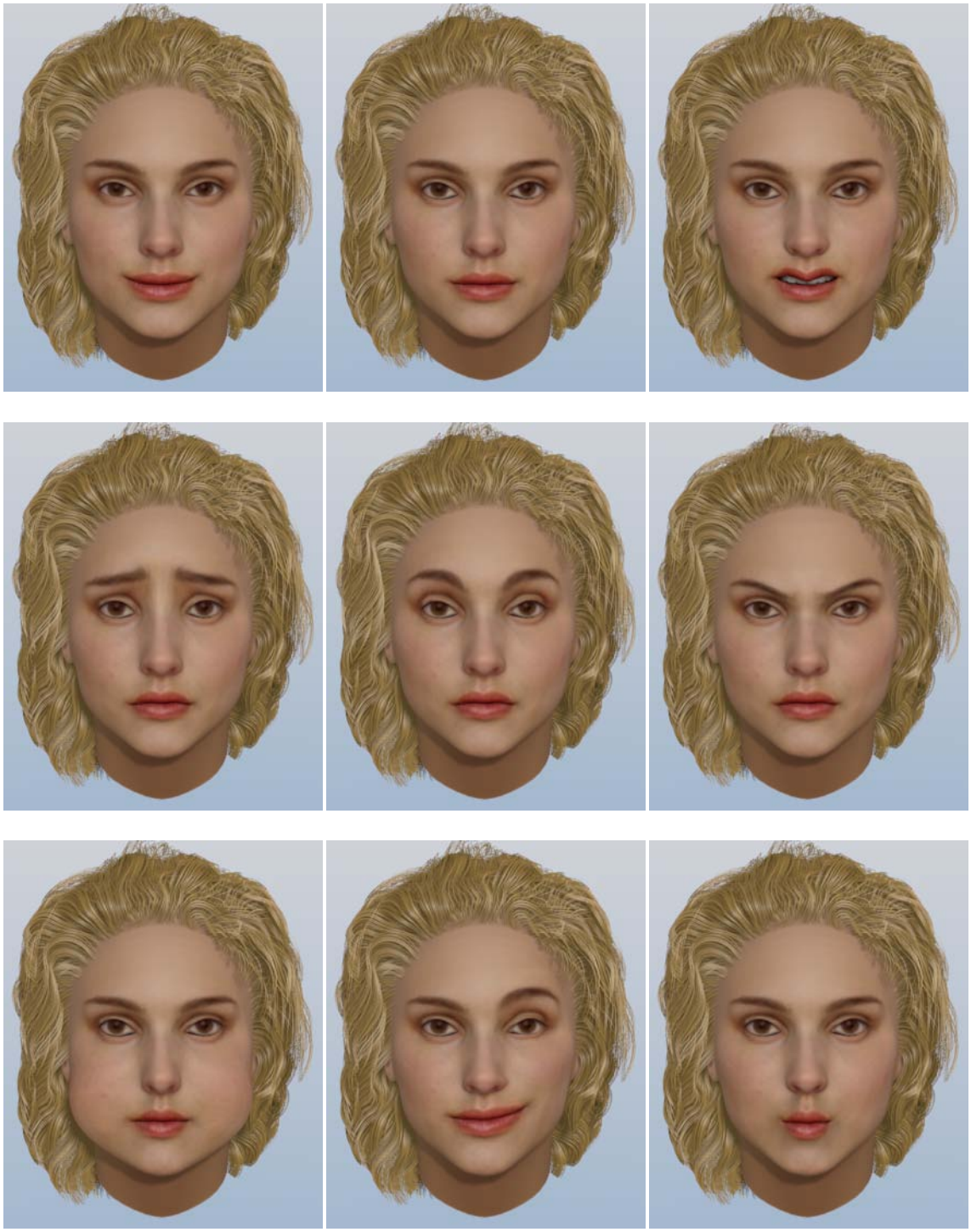


Figure 4.10: Facial expressions.

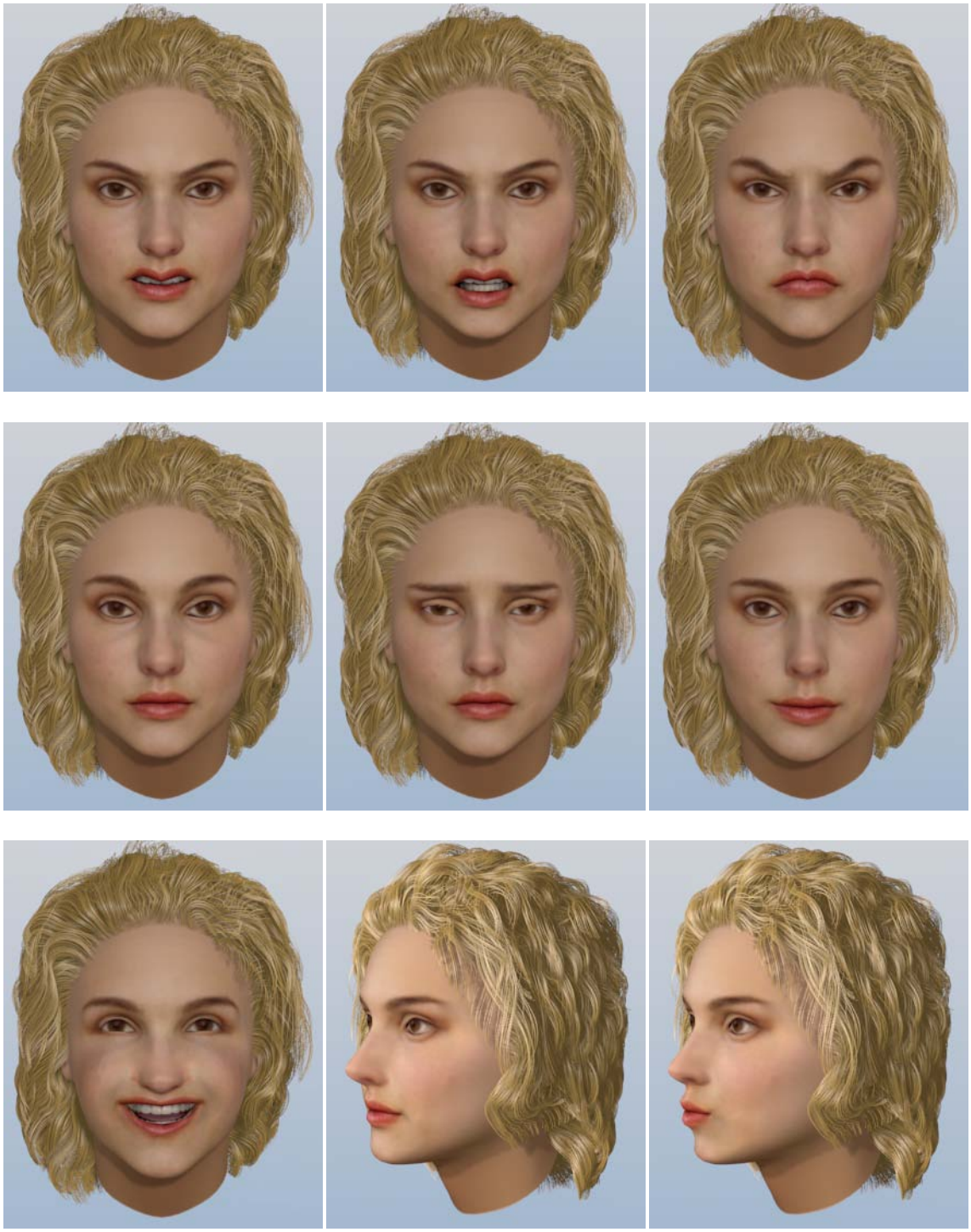


Figure 4.10, continued.

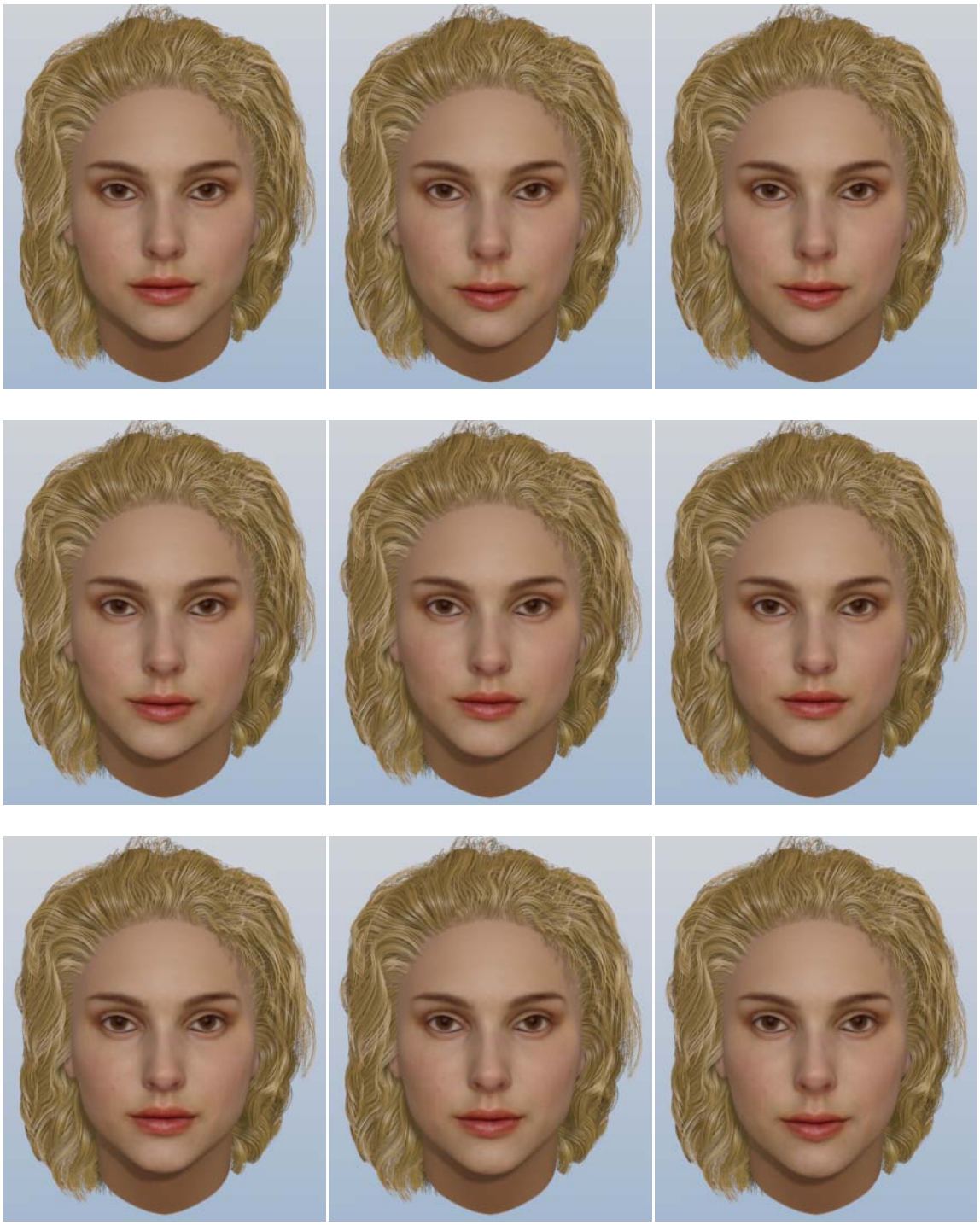


Figure 4.11: Selected frames from an animation of chewing.

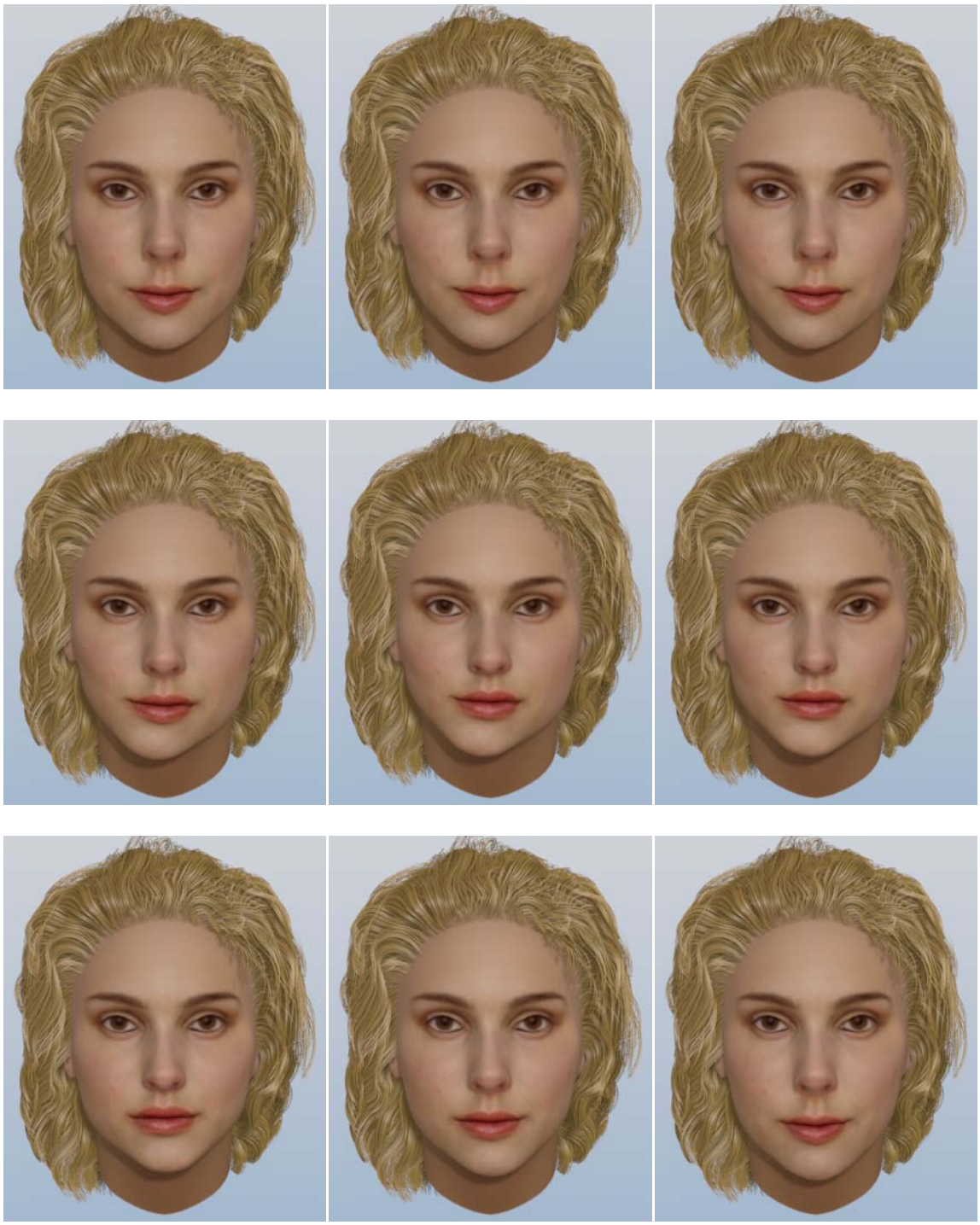


Figure 4.11, continued.

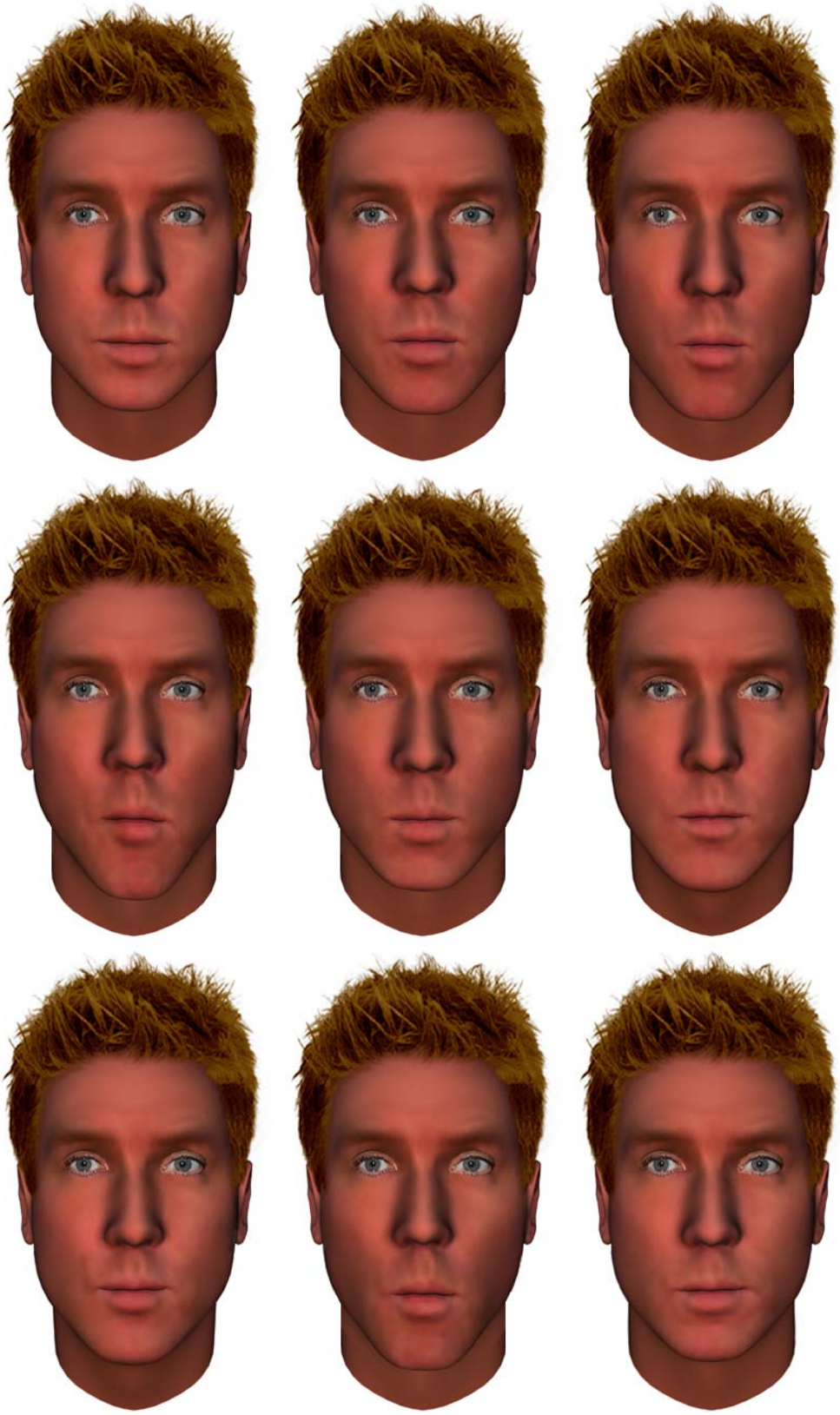


Figure 4.12: Selected frames of a running cycle of chewing.

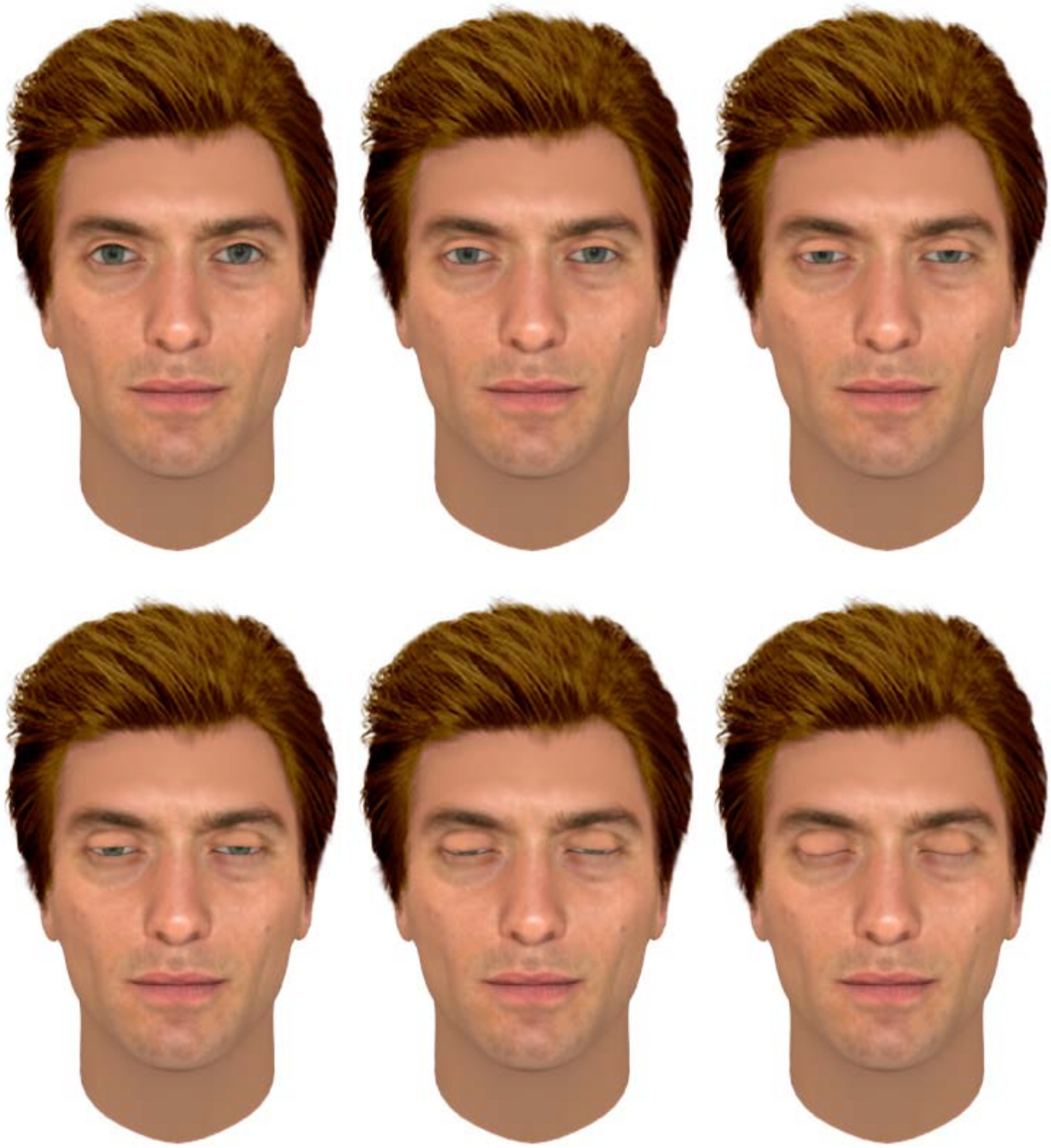


Figure 4.13: Selected frames of a running cycle as the eyes close.

Copyright © Alice J. Lin 2011

Chapter 5 Copying Facial Expressions

5.1 Introduction

The face is the primary part of the body that we use to recognize individuals. We can recognize a specific face from a vast universe of similar faces and are able to detect very subtle changes in facial expression. The ability to model human faces and then animate subtle nuances of facial expressions remains a significant challenge in computer graphics. Despite a heavy reliance on traditional computer graphics algorithms, facial modeling and animation are still being regarded as without broadly accepted solutions.

To generate animated facial expressions requires generating continuous and realistic transitions between different facial expressions. In general, morphing between arbitrary polygonal meshes is difficult, since it requires a set of correspondences between meshes with potentially different topologies that can produce a reasonable set of intermediate shapes.

Facial animations of 3D models derive from physical behaviors of the bone and muscle structures. Others focus on the surface of the face, using smooth surface deformation mechanisms to create dominant facial expressions. These approaches are only adequate for making individual models. If one wants to create a new model for an animation, method-specific tuning is inevitable (or otherwise the animation has to be produced from scratch).

A parametric approach associates the motion of a group of vertices to a specific parameter [27]. This manual association must be repeated for models with different mesh structures. Animation parameters do not simply transfer between models. If manual tuning or computational costs are high in creating animations for one model, creating similar animations for new models will take similar efforts. Vector-based muscle models place the heuristic muscles under the surface of the face [6, 91]. This process is repeated for each new model. A three-layer mass-spring-muscle system requires extensive computation [92]. The final computed parameters are, however, only useful for one model. Free-form deformation manipulates control points to create key facial expressions [49], but there is no automatic method for mapping the control points from one model to another.

In practice, animators often sculpt key-frame facial expressions for every three to five frames to achieve the highest-quality animations. When a new model is considered, those fitting or sculpting processes must be repeated even if the desired expression sequences are similar.

5.2 Background

5.2.1 Deforming Objects

Deforming an object shape and transforming one shape into another is a visually powerful animation technique. It adds the notion of malleability and density. Flexible body animation makes the objects in an animation seem much more expressive and alive.

There are physically based approaches that simulate the reaction of objects undergoing forces. However, many animators want more precise control over the shape of an object than that provided by simulations and/or do not want the computational expense of simulating physical processes. In such cases, the animator wants to deform the object directly and define key shapes. Shape definitions that share the same edge connectivity can be interpolated on a vertex-to-vertex basis in order to smoothly change from one shape to the other. A sequence of key shapes can be interpolated over time to produce flexible body animation. Multivariate interpolation can be used to blend among a number of different shapes. The various shapes are referred to as blend shapes or morph targets and are a commonly used technique in facial animation.

5.2.2 Shape Interpolation

Changing one 3D shape into another 3D shape is a useful effect, but techniques are still being developed for arbitrary shapes. Several solutions exist that have various limitations. The techniques fall into one of two categories: surface-based or volume-based. The surface-based techniques use the boundary representation of the objects and modify one or both of them so that the vertex-edge topologies of the two objects match. Once this is done, the vertices of the object can be interpolated on a vertex-by-vertex basis. Surface-based techniques usually have some restriction on the types of objects they can handle, especially objects with holes through them. The number of holes through an object is an important attribute of an object's structure, or topology. The volume-based techniques consider the volume contained within the objects and blend one volume into the other. These techniques have the advantage of being less sensitive to different object topologies.

However, volume-based techniques usually require volume representations of the objects and therefore tend to be more computationally intensive than surface-based approaches. In addition, the connectivity of the original shapes is typically not considered in volume-based approaches. Thus, some information, often important in animation, is lost [71].

For most approaches, the shape transformation problem can be discussed in terms of two subproblems: (1) the correspondence problem, or establishing the mapping from a vertex (or other geometric element) on one object to a vertex on the other object; and (2) the interpolation problem, or creating a sequence of intermediate objects that visually represent the transformation of one object into the other. The two problems are related because the elements that are interpolated are typically the elements between which correspondences are established. A notable characteristic of the various algorithms for shape interpolation is the use of topological information versus geometric information. Topological information considers the logical construction of the objects and, when used, tends to minimize the number of new vertices and edges generated in the process. Geometric information considers the spatial extent of the object and is useful for relating the position in space of one object to the position in space of the other object.

5.2.4 Performance Animation

The creation of facial animation control parameters is a tedious process at best. Often, many dozens of parameters have to be specified and coordinated to create a short sequence. For the face, it is important to carefully orchestrate these parameters, since actions have to be precisely timed to create believable expressions.

To alleviate the manual activity of creating animation parameters, it is possible to capture an actor's performance and translate her motions into a suite of control parameters. This has advantages for the director because real-time motion capture can be created at low resolution, allowing performances to be blocked in and allowing subsequent high-fidelity rendering, incorporating all the capture data, to be completed offline.

The precise capture of an actor's performance, and the playback of his performance as a computer-generated model, has become important. Capture sessions can now provide sufficient fidelity to accurately create an individual's motion nuances. Tom Hanks in The Polar Express (Sony Pictures Imageworks 2004) and the characters in Monster House (Sony Pictures Imageworks 2006) have been used in commercial production. To date, this requires specialized techniques, where the actor is marked and tracked in a controlled studio environment [60]. The interest in capturing performances from actors has led to a variety of different techniques, with varying degrees of success [13].

MoCap or motion capture refers to the recording of specific markers placed on the body or face during a performance. The motion data is then used to drive a computer-generated character. Systems often require reflectance markers to be carefully placed on the face, and for whole body systems, the performer wearing a body suit with markers strategically placed on the surface.

Blend shapes are key facial postures defined in a dataset, which can be used to define a linear space for facial expressions. As the name suggests, a new posture can be generated as a blend of two or more existing postures. Controlling the linear space can be specified through generalized functions, or determined automatically from capture sessions. Blend shaping is one of the most widespread techniques in facial animation and is controlled by weights - values on the basis vector - the geometry.

Re-targeting is that the recorded source material is adapted to a target character. This is sometimes referred to as cross-mapping and is extensively used in performance-based systems. Re-targeting allows an actor's performance to be mapped onto characters that need not be a direct resemblance in themselves.

Rotoscoping is a technique that has a long history in traditional animation. A live performance is recorded and through each frame, the actor is traced to create a reference from which the animation sequence is generated. In many respects, computer-based performance animation is a versatile variant of rotoscoping, especially when the captured data of a three-dimensional model can be observed from a variety of novel viewpoints.

Expression cloning is the ability to map vertex motion vectors from a source model to a new geometry having a different geometric proportions and topological layout. In many respects this is the same as re-targeting, with the variation being more versatile.

PCA, principal component analysis, is a mathematical technique capable of reducing multi-dimensional data sets to lower dimensions for analysis. Within the context of faces, PCA is capable of capturing all the geometric variations of a performance in a single dataset. Such analysis is useful when considering playback compression techniques on devices with low computational resources, such as game consoles and video-teleconferencing systems. PCA is particularly effective for face data where there is a high level of correlated geometric structure and texture map motion over long sequences.

Dense motion capture refers to tracking multiple markers at high density. To create high density markers, reflective elements in phosphorescence paint are applied to the face, like makeup. The markers can be seen under ultraviolet light and tracked. Whereas single large markers are placed at strategic locations on the face and tracked frame to frame in most MoCap systems, dense marker patches are uniquely identified and tracked over time.

The ultimate performance-based systems will be unobtrusive and markerless. Such systems will operate with unconstrained actors, who then can perform in sets with props, as well as with other actors, both indoors and outside. This is in contrast to current state-of-the-art systems, where performances are typically captured in controlled environments. For example, high-fidelity data capture sessions often require high luminosity, physical restraints for the head, or even clumsy headgear, hardly a conducive environment to generate the best performances from actors. Nevertheless, much progress has been achieved in the recent past and has extended the quality of facial animations. Indeed, the

levels of realism demanded by the film industry have come close to fooling the observer into thinking that the final virtual actor is real.

A majority of techniques developed using performance systems combine modeling, animation and control. As a result, adjusting the geometry of the face and modifying the captured performances can be awkward. Therefore, systems that separate geometry capture from animation control are likely to be more successful, because they will allow flexibility for modelers and animators.

Feature tracking and correspondence is identifying salient features, (such as the outermost point of the eyebrow), observing its location from frame-to-frame, then mapping that point to a corresponding vertex of the geometry model. Many of today's performance-driven systems explicit mapping of tracked feature points placed on the head and face. The simplest approach to tracking facial features is to track markers placed directly on the subject at salient points. This technique is appropriate for one-time performances in which the actor is placed in an artificial environment and recorded. The actions can then be processed in an offline process to extract the x, y location of the markers in the image plane. The resulting displacements of these points then can be used to control a face geometry and texture map. This technique was first demonstrated by Williams [9] and has been repeated in many other systems to date [9, 36, 60].

5.3 Related Work

Automatically transferring facial motions from an existing (source) model to a new (target) model can significantly save painstaking work and model-specific animation tuning for the new facial model. Transferring facial motions between two 3D face meshes can be performed through geometric deformations. Noh and Neumann [93] proposed a technique to transfer vertex displacements from a source model to target models. The basic idea is to construct vertex motion mappings between models through the radial basis functions morphing. But each motion vector's direction and magnitude have to be adjusted before transferring. The direction of a source motion vector must be altered to maintain its angle with the local surface when applied to the target model. The magnitude of a motion vector must be scaled by the local size variations. Sumner and Popovic [94] proposed a general framework that transfers geometric deformations between two triangle meshes, which can be directly applied to retarget facial motions from one source face mesh to a target face mesh.

A number of approaches were proposed to transfer source facial motions to blend shape face models [35, 52, 55, 95, 96] due to the popularized use of blend shape methods in industry practice. Choe and Ko [35] transferred tracked facial motions to target blend shape face models composed of hand-generated muscle actuation bases, by iteratively adjusting muscle actuation bases and analyzed weights through an optimization procedure. The work of Pyun et al. [95, 96] showed transfer of facial animations using example-based approaches. Essentially, these approaches require animators to sculpt proper blend-shape face models based on a set of key facial poses, delicately chosen from

source facial animation sequences. Hence, it is difficult to apply these techniques to pre-designed blend-shape models without considerable effort. Sifakis et al. [55] first created an anatomically accurate face model composed of facial musculature, passive tissue, and underlying skeleton structure. They then used nonlinear finite element methods to determine accurate muscle actuations from the motions of sparse facial markers. Anatomically accurate 3D face models are needed for this approach, which is another challenging task itself in computer animation. Deng et al. [52] proposed an automatic technique to directly map 3D facial motion capture data to pre-designed blend shape face models. In their approach, Radial Basis Function networks are trained to map a new motion capture frame to its corresponding blend shape weights, based on chosen training pairs between mocap frames and blend shape weights. Bilinear models and multilinear models were proposed to transform facial motions [97-99]. Chuang and Bregler [98, 99] learned a facial expression mapping/transformation function from training video footage using the bilinear models [100]. They then used this mapping function to transform input video of neutral talking to expressive talking. Vlasic et al. [97] proposed a framework to transfer facial motion in video to other 2D or 3D faces by learning statistical multilinear models from scanned 3D face meshes. In their work, the learned multilinear models are controlled via intuitive attribute parameters.

5.4 Generating Identical Topologies

We first construct a neutral face model, and then perform the deformation on the neutral model to get a variety of facial expressions. Figure 5.1 shows eight facial expressions. Once the animated facial expressions are created, we install them as the template models.

Figure 5.2(a) is an example of a template model. There is a separate facial model (target model) and its mesh structure is different from the template model. Figure 5.2(b) is an example of a target model. We want the target model to have the same animated facial expressions as the template model (e.g. smiles). Our goal is to generate a mesh (see Figure 5.2(c)) that has the same topology as Figure 5.2(a) for the target model (Figure 5.2(b)). Our novel approach is to transform the mesh representation of the template model into a mesh representation of the target model so that topology of the target model's mesh structure is the same as the template model's mesh structure. Once the target and the template models have the same mesh structure, we transfer facial expressions from the template model to the target model by using its motion data.

In general, pre-defined models have extra dangling mesh pieces. For instance, Figure 5.3(a) is the front view of the surface of mouth and Figure 5.3(b) is the back view of the surface. The preliminary process is to remove these dangling pieces. The results are shown in Figure 5.3(c) and Figure 5.3(d). Essentially all human faces have the same basic structure and are similar in shape. For models with human facial structures, we choose the same starting facial expression (e. g. neutral) for both the template model and the target model, and then search for corresponding facial feature points in both models. The eye corners, upper and lower eyelids, mouth corners, uppermost and lowermost parts of the lips, nose tip and edges, etc. are marked. We resize the template model to try to establish the same size facial features as the target model, and then superpose the template model on the target model according to correspondence between features.

Assume that both the template and the target models have θ specified corresponding facial feature points, $i = 1, ..., \theta$. The template model has n vertices. j is not a marked facial feature point in the template model. $j = 1, ..., n - \theta$.

We fix the target model facial feature point F_i as a vector, and we move the template model facial feature point P_i as a vector toward F_i. The displacement vector for the facial feature point is $D_i = F_i - P_i$. The displacement vector for a non-facial feature point is: $D_j = \sum_{i=1}^{\theta} W(j,i) D_i$.

$W(j,i)$ is a weight function. $W(j,i) = \frac{dis(j,i)}{\sum_{i=1}^{\theta} dis(j,i)}$, where $dis(j,i)$ is the distance between point j and point i.

When specific facial feature points are matched between the two models, a morphing is performed on the mesh of the template model. The template model is deformed to certain extent, depending on the differences between these two models. Once that is done, we map each vertex in the template model to the surface of the target model.

Assume that the template model has n vertices and that the target model has m vertices. The algorithm for generating the same topology works even when $m < n$, however, we can get better results if we subdivide the target model to make $m > n$. $i = 1, ..., n$. $j = 1, ..., m$. V_i is the vector of the ith vertex on the template model. V_j is the

vector of the jth vertex on the target model. For each V_i when the distance d between V_i and V_j is minimum, $d = \min\left|V_i - V_j\right|$, V_i maps to V_j. The transformation from the template model to the target model is defined by a replacement vector W_i for each vertex v_i. $W_i = V_j$ on the generated mesh.

Each mapped vertex w_i in the generated mesh is subject to the following constraint. The vertex v_i in the template model has spatial relationships with its neighboring vertices. For example, in Figure 5.4(a), there are nine vertices in the template model. Suppose $v_i = v_9$, then v_9 has spatial relationships with eight vertices $v_1, v_2, ...v_8$. These nine vertices are mapped to the target model (see Figure 5.4(b)). The corresponding vertex of v_9 is w_9. w_9 has to keep the same spatial relationships with $w_1, w_2, ..., w_8$, including the nine vertices' order. Figure 5.4(b) shows a correct relationship on the generated mesh. The situation in Figure 5.4(c) is something we want to avoid. However, if w_7 and w_8 overlap, then there is no problem. The result is that the *n* vertices of the template model are all mapped onto the surface of the target model. The mesh now represents the target model (see Figure 5.2(c)).

Once the topologies of the two facial meshes are identical, there is already a natural correspondence between their vertices. A satisfactory 3D morphing sequence can be obtained using simple linear interpolation between geometric coordinates of corresponding vertices in the two facial meshes. To copy an expression from the template model to the target model, we need motion data for the template model animation from

one expression (e.g. the neutral face) to another expression (e.g. the smiling face). The motion data contains each vertex's positional change between these two expressions. We proportionally apply the motion data to the newly generated mesh. The newly generated mesh thus gets an expression (e.g. smiling). For the animation, we transform one (neutral face) expression into another facial expression (e.g. smiling). We can simply interpolate the two shapes since they share the same vertex-edge topology. Facial expressions are transformed by interpolating the positions of vertices on a vertex-by-vertex basis. The correspondence between the two shapes is established by the vertex-edge connectivity structure shared by the two models.

The above approach faithfully copies facial expressions and motions between models. With this method, anyone can create many different models with the same animated expression, even if these models have different topologies. For each model, anyone can create a variety of expressions. Figure 5.5 shows examples of copying facial expressions between three models.

5.5 Summary

We have presented a method to make two facial models to have the same mesh structure. Thus, 3D morphing sequence can be obtained using linear interpolation between coordinates of corresponding vertices in the two facial meshes. Our technique produces natural looking expressions and animations for arbitrary mesh structures. This method is not developed with ad hoc techniques, so it is easily extendible and will not rapidly become brittle. Generating the same expression model often is a tedious and complex

process requiring substantial artistic skills. Our method is not only for animation by a trained artist, but also for ordinary engineers.

Since the motion capture facial animation method can produce good results and speed up the animation process, we performed tests to map motion capture data onto the template model, and then transfer them to the target model. This is an interesting direction for future research and a means to extend the application of our work.

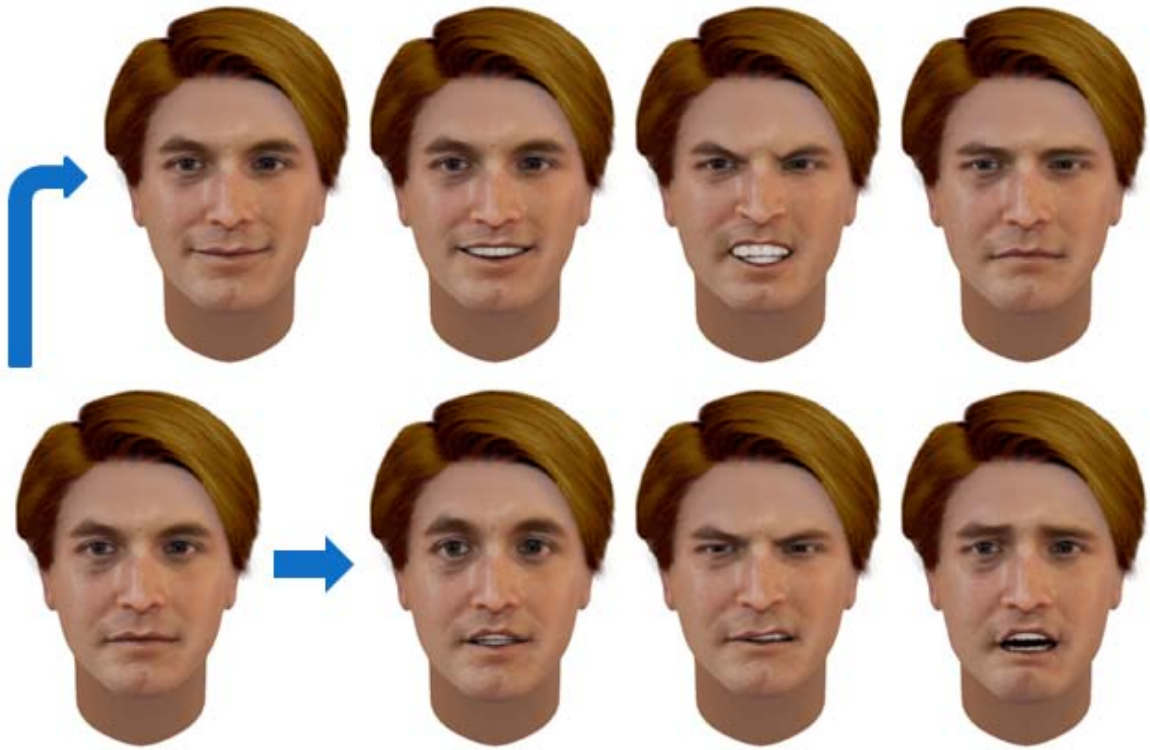
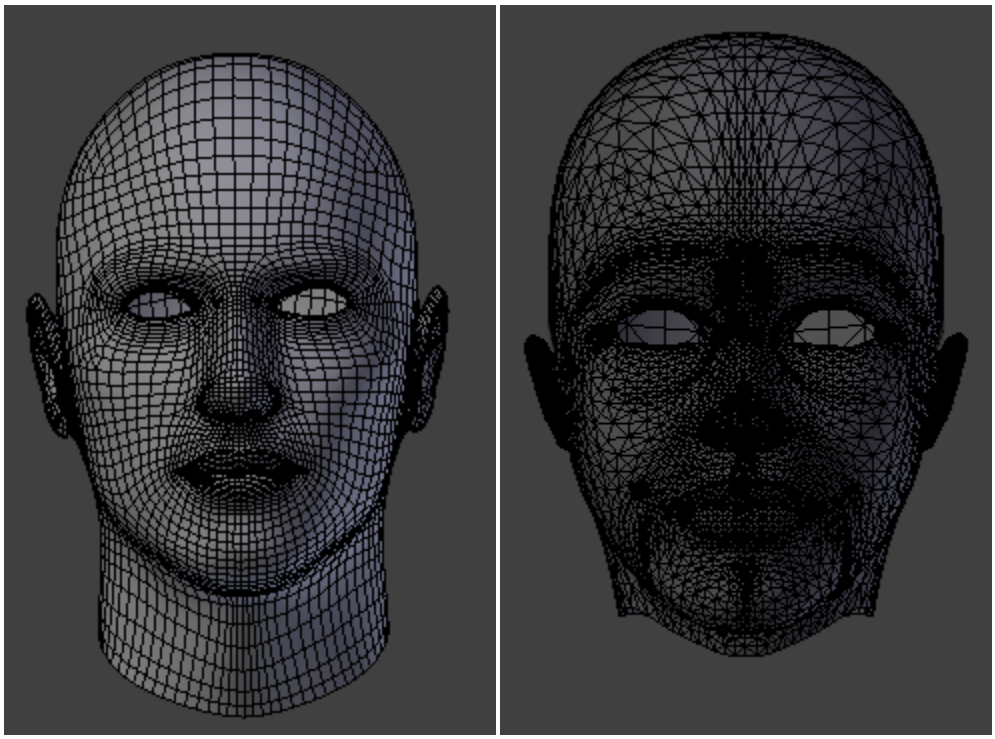
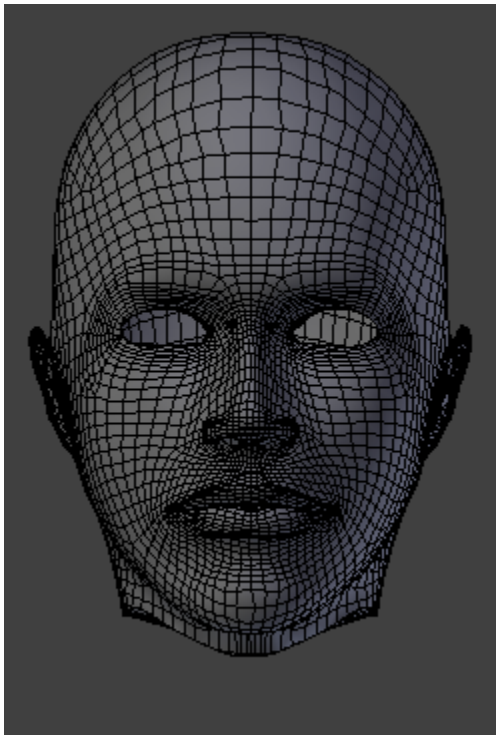


Figure 5.1: Eight facial expressions.



(a) (b)



(c)

Figure 5.2: (a) Template model. (b) Target model. (c) New mesh representation created for the target model.

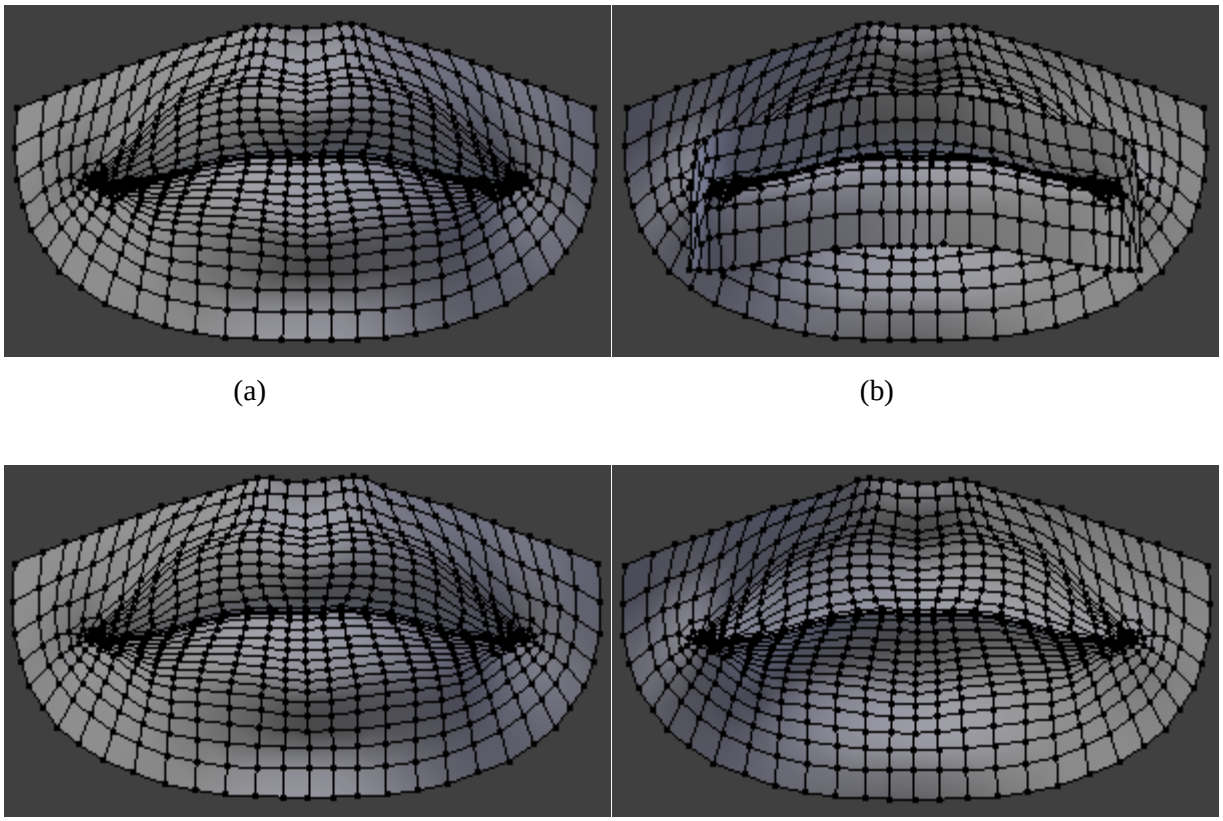


Figure 5.3: (a) Original front view. (b) Original back view. (c) Front view of the processed mouth. (d) Back view of the processed mouth.

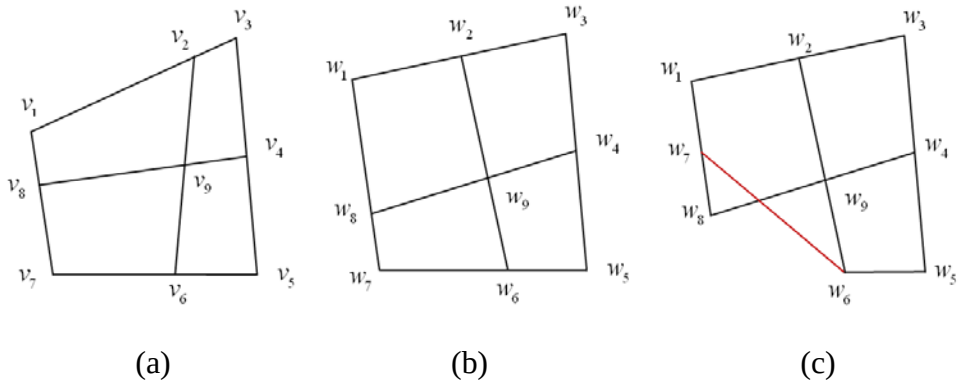


Figure 5.4: Spatial relationship between neighboring vertices. (a) Vertex relationships on the template model. (b) Correct relationship on the generated mesh. (c) Incorrect relationship on the generated mesh.

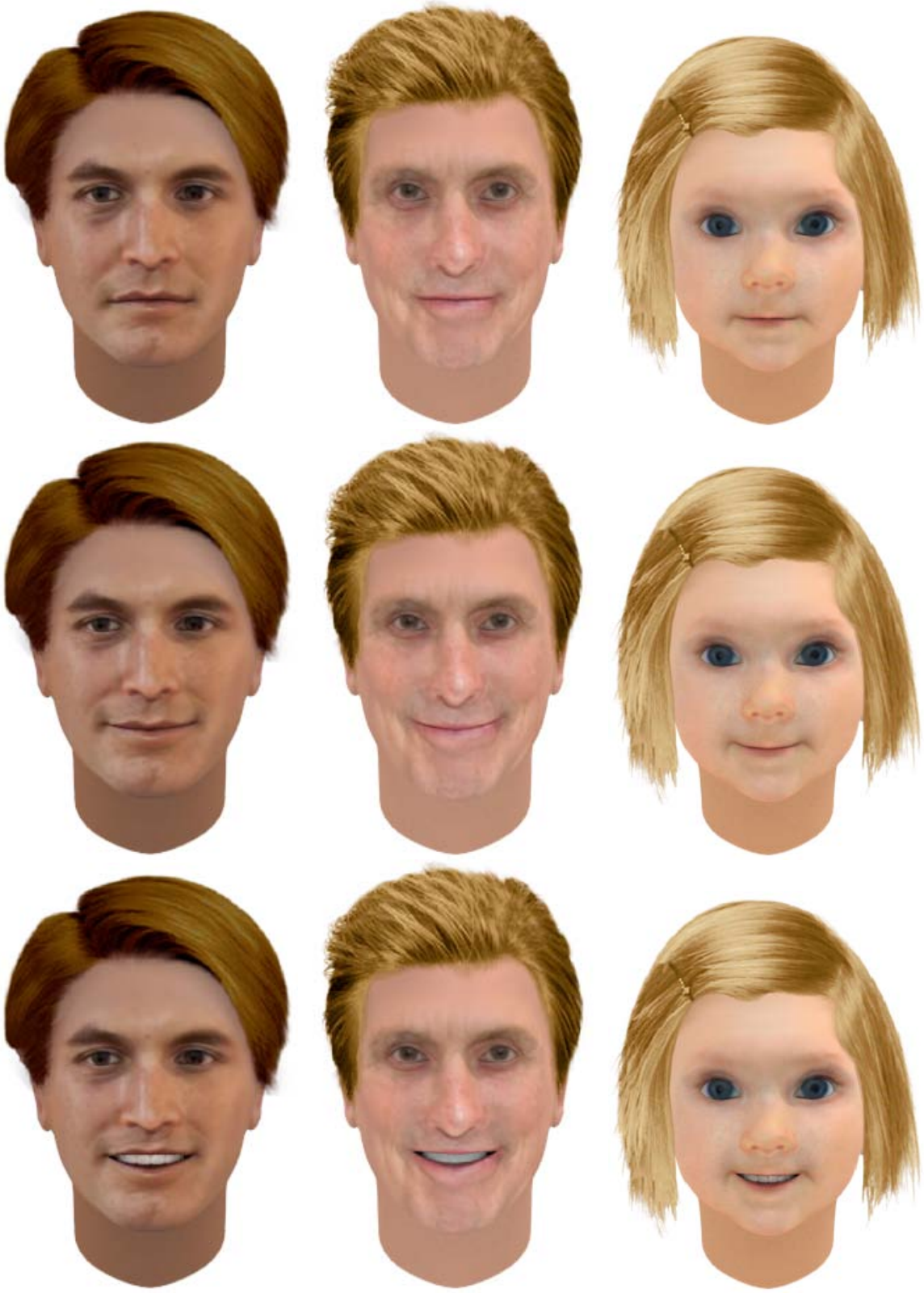


Figure 5.5: Examples of facial expressions and copied facial expressions.

Figure 5.5, continued.

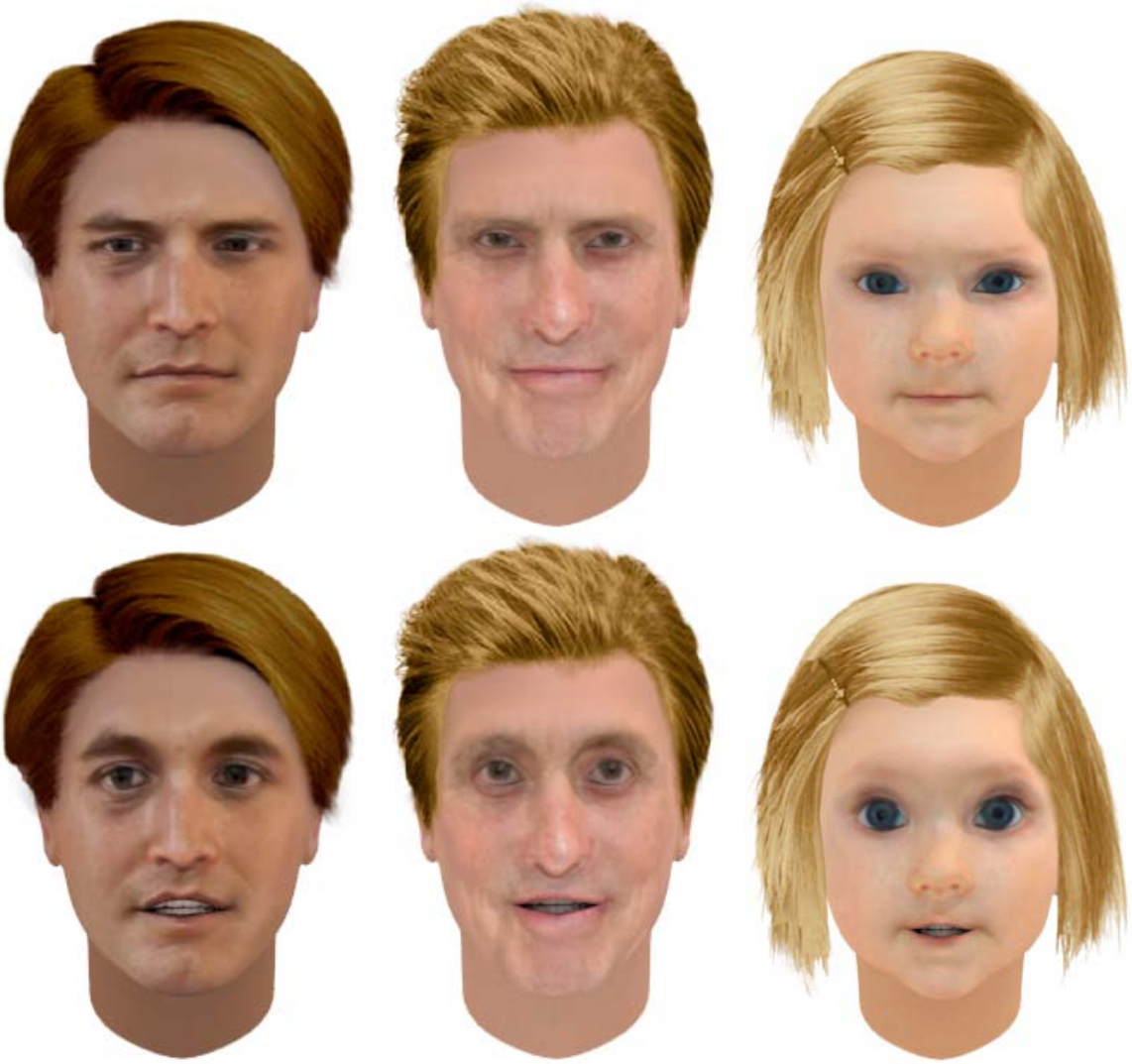


Figure 5.5, continued.

Copyright © Alice J. Lin 2011

Chapter 6 Conclusion and Future Work

In this dissertation, we have designed and developed methods for modeling and animating facial expressions. In Chapter 2, we presented a system for generating detailed, dynamic facial expressions. Despite the complexity of human facial anatomy and people's inherent sensitivity to facial appearances, we have created a real-time system that generates realistic subtle facial expressions and adapts to individual faces. Our method has produced realistic results with fast performance, and the process for generating facial animation requires less human intervention or tedious tuning. The tool is intuitive and easy to use, allowing users to directly manipulate it and see immediate results. The system also allows the expressive features to be applied to any existing animation or model. The location and style can be specified, thus allowing the dynamic detailed facial expressions to be modeled as desired. It also can be applied to a variety of CG human body skins and animal faces.

In Chapter 3, we developed a novel approach for generating realistic, animated tears. We formulated teardrops which continually change shape as the tear drips down the face. We also designed the method and procedures for generating shedding tears. Our results indicate that these methods are robust and accurate. Our methods both broaden CG and increase the realism of facial expressions. It will evolve 3D facial expression and animations significantly. Our methods can also be applied to CG human body skins and other surfaces to simulate flowing water. Our method is not only for animation by a trained artist, but also for industry.

Chapter 4 presented a method for automatically placing bones on facial models to speed up the rigging process of a human face. The resulting facial deformations and animation are realistic and expressive. Our method works on a particular person's face quickly and effectively, and provides a powerful means of generating facial expressions and animations. The time saved with our method is significant. Although we have focused entirely on the human face in this work, we believe the concepts presented here are flexible enough to be used on non-human characters. Our method, in conjunction with existing techniques, allows users to go from a static mesh to an animated character quickly and effortlessly.

Chapter 5 presented a method which transformed two different facial topologies to make them identical. Thus, 3D morphing sequences can be obtained using simple linear interpolation between the geometric coordinates of corresponding vertices in each of the two facial meshes. With this method, anyone can create many different models with the same animated expression, even if these models have different topologies. For each model, anyone can create a variety of expressions. Our technique produces natural looking animations for arbitrary mesh structures. This method is not developed with ad hoc techniques, so it is easily extendible and will not rapidly become brittle.

In the future, we will extend our research to mouth animations. The inside parts of the mouth play important roles in facial animations, especially in speech simulation [101-104]. In most facial animations, the teeth, gums and tongue are often omitted or oversimplified. When modeled, the mouth is usually represented as a simple

parallelepiped. Although only a small portion inside the mouth is visible during normal speech, its simulation is important for realistic, synthesized mouth animations. Eye movements are complex. Generally they are saccadic (or discontinuous and jumping from one point to another) in nature, followed by fixations or focus at a particular area [105]. The direction of the gaze can be an evocative element in a portrait. Downcast eyes, upraised eyes, eyes looking sideways, even out-of-focus eyes are all suggestive of the state of the mind [68]. Gaze and eye positions have important meanings in emotion too. Staring at somebody during anger, looking down during sadness, raising the eyes during surprise or fear, are all powerful reflections of emotions. Eye motion during complex expressions needs to be modeled.

Another facet that may prove of interest is the facial skin color changes. It is also associated with the perception of certain emotions. For example, the skin color pales during anger and reddens during shame. Both add to the emotional effect. When included, this can increase realism and amplify extreme emotions or effort. The dynamic texture of facial models should thus be further explored. The simulation of hair may also be a fruitful enhancement to the facial expression and animation. Hairs of the eyebrows accentuate the movement of the eyebrows. In the case that the facial model has a moustache or beard, those would be affected by facial animation.

Since the motion capture facial animation method can produce good results and speed up the animation process, we performed tests to map motion capture data onto the

template (source) model, and then transferred it to the target model. This is an interesting direction for future research and a means to extend the application of our work.

Copyright © Alice J. Lin 2011

Bibliography

[1] H. Chernoff, “The Use of Faces to Represent Points in N-Dimensional Space Graphically,” *Technical Report Project NR-042-993, Office of Naval Research*, 1971.

[2] M. L. Gillenson, “The Interactive Generation of Facial Images on a CRT Using a Heuristic Strategy,” *Ph.D. thesis, Ohio State University, Computer Graphics Research Group, Columbus, Ohio*, 1974.

[3] F. I. Parke, “A Parametric Model for Human Faces,” Ph.D. thesis, University of Utah, Salt Lake City, 1974.

[4] S. M. Platt, “A System for Computer Simulation of the Human Face,” Master’s thesis, The Moore School, University of Pennsylvania, 1980.

[5] P. Bergeron, and P. Lachapelle, “Controlling Facial Expressions and Body Movements,” *ACM SIGGRAPH 1985 Tutorial Notes*, pp. 61-79, 1985.

[6] K. Waters, “A Muscle Model for Animating Three-Dimensional Facial Expressions,” *Computer Graphics (SIGGRAPH Proceedings)* vol. 21, no. 4, pp. 17-24, 1987.

[7] N. Magnenat-Thalmann, E. Primeau, and D.Thalmann, “Abstract Muscle Action Procedures for Human Face Animation,” *Visual Computer*, vol. 3, no. 5, pp. 290-297, 1988.

[8] F. I. Parke, “State of the Art in Facial Animation,” *SIGGRAPH Course Notes 26.* , 1990.

[9] L. Williams, “Performace Driven Facial Animation,” *Computer Graphics* vol. 24, no. 4, pp. 235-242, 1990.

[10] V. Blanz, and T. Vetter, “A Morphable Model for the Synthesis of 3D Faces,” *In Proceedings of SIGGRAPH 99*, pp. 187-194, 1999.

[11] Y. Lee, D. Terzopoulos, and K. Waters, “Constructing Physics-Based Facial Models of Individuals,” *In Graphics Interface ’93*, pp. 1-8, 1993.

[12] Z. Deng, and U. Neumann, "Data-Driven 3D Facial Animation," Springer Verlag, 2008.

[13] F. I. Parke, and K. Waters, "Computer Facial Animation," A K Peters, Ltd, 2008.

[14] P. J. Ekman, and W. V. Friesen, "The Facial Action Coding System: A Technique for the Measurement of Facial Movement," Consulting Psychology Press, 1978.

[15] P. Ekman, and H. Oster, “Facial Expressions of Emotion,” *Annual Review of Psychology,* vol. 30, pp. 527-554, 1979.

[16] P. Ekman, "Basic Emotions," *Handbook of Cognition and Emotion*, Wiley & Sons, 1999.

[17] K. L. Schmidt, and J. F. Cohn, “Human Facial Expressions as Adaptations: Evolutionary Questions in Facial Expression Research,” *American Journal of Physical Anthropology,* vol. Suppl 33, no. S33, pp. 3-24, 2001.

[18] J. F. Cohn, and K. L. Schmidt, “The Timing of Facial Motion in Posed and Spontaneous Smiles,” *International Journal of Wavelets, Multiresolution and Information Processing,* vol. 2, pp. 1-12, 2004.

[19] T. Valentine, S. Darling, and M. Donnelly, “Why Are Average Faces Attractive? The Effect of View and Averageness on the Attractiveness of Female Faces,” *Psychonomic Bulletin & Review,* vol. 11, pp. 482-487, 2004.

[20] M. Oka, K. Tsutsui, A. Ohba *et al.*, “Real-Time Manipulation of Texture-Mapped Surfaces,” *SIGGRAPH conference on Computer graphics and interactive techniques* vol. 21, pp. 181-188, 1987.

[21] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, “Calculus of Non-rigid Surfaces for Geometry and Texture Manipulation,” *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 5, pp. 902-913, 2007.

[22] F. Pighin, J. Hecker, D. Lischinski *et al.*, “Synthesizing Realistic Facial Expressions from Photographs,” *ACM SIGGRAPH Computer Graphics,* vol. 32, pp. 75-84, 1998.

[23] T. Beier, and S. Neely, “Feature-based Image Metamorphosis,” *ACM SIGGRAPH Computer Graphics,* vol. 26, pp. 35-42, 1992.

[24] F. I. Parke, “Computer Generated Animation of Faces,” *Proceedings of the ACM Annual Conference*, pp. 451-457, 1972.

[25] K. Arai, T. Kurihara, and K. Anjyo, “Bilinear Interpolation for Facial Expression and Metamorphosis in Real-Time Animation,” *The Visual Computer,* vol. 12, pp. 105-116, 1996.

[26] M. M. Cohen, and D. W. Massaro, “Modeling Coarticulation in Synthetic Visual Speech,” *Models and Techniques in Computer Animation*, pp. 139-156 1993.

[27] F. I. Parke, “Parameterized Models for Facial Animation,” *IEEE Computer Graphics and Applications,* vol. 2, no. 9, pp. 61-68, 1982.

[28] F. I. Parke, “Parameterized Models for Facial Animation Revisited,” *ACM SIGGRAPH Facial Animation Tutorial Notes*, pp. 53-56, 1989.

[29] S. Basu, N. Oliver, and A. Pentland, "3D Modeling and Tracking of Human Lip Motions," *Proceedings of International Conference on Computer Vision*, pp. 337-343, 1998.

[30] R. M. Koch, M. H. Gross, F. R. Carls *et al.*, "Simulating Facial Surgery Using Finite Element Models," *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 421-428, 1996.

[31] E. Sifakis, A. Selle, A. Robinson-Mosher *et al.*, "Simulating Speech with a Physics-based Facial Muscle Model," *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 261-270, 2006.

[32] S. Pieper, J. Rosen, and D. Zeltzer, "Interactive Graphics for Plastic Surgery: A Task Level Analysis and Implementation.," *Computer Graphics, Special Issue: ACM SIGGRAPH, 1992 Symposium on Interactive 3D Graphics*, pp. 127-134, 1992.

[33] S. Platt, and N. Badler, "Animating Facial Expression," *ACM SIGGRAPH Computer Graphics*, vol. 15, pp. 245-252, 1981.

[34] D. Terzopoulos, and K. Waters, "Physically-Based Facial Modeling, Analysis, and Animation," *The Journal of Visualization and Computer Animation*, vol. 1, pp. 73-80, 1990.

[35] B. Choe, H. Lee, and H. S. Ko, "Performance Driven Muscle Based Facial Animation," *Journal of Visualization and Computer Animation*, vol. 12, no. 2, pp. 67-79, 2001.

[36] B. Guenter, C. Grimm, D. Wood *et al.*, "Making Faces," *ACM SIGGRAPH 98, Computer Graphics*, vol. 32, pp. 55-66, 1998.

[37] F. Pighin, R. Szeliski, and D. H. Salesin, "Resynthesizing Facial Animation Through 3D Model-Based Tracking," *Proceedings of International Conference on Computer Vision*, pp. 143-150, 1999.

[38] J. M. Buenaposada, and E. Munoz, "Performance Driven Facial Animation Using Illumination Independent Appearance-Based Tracking," *Proceedings of the 18th International Conference on Pattern Recognition*, vol. 1, pp. 303 - 306, 2006.

[39] H. Fuchs, Z. M. Kedem, and S. P. Uselton, "Optimal Surface Reconstruction From Planar Contours," *ACM SIGGRAPH Computer Graphics*, vol. 11, no. 2, pp. 236-236, 1977.

[40] I. Braude, "Smooth 3D Surface Reconstruction from Contours of Biological Data With MPU Implicits," *Master's thesis. Drexel University*, 2005.

[41] D. Terzopoulos, and M. Vasilescu, “Sampling and Reconstruction with Adaptive Mashes,” *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, pp. 70-75, 1991.

[42] M. Vasilescu, and D. Terzopoulos, “Adaptive Meshes and Shells: Irregular Triangulation, Discontinuities, and Hierarchical Subdivision,” *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference* pp. 829-832, 1992.

[43] C. S. Choi, H. Harashima, and T. Takebe, "Highly Accurate Estimation of Head Motion and Facial Action Information on Knowledge-Based Image Coding," *Technical Report PRU90-68*, Institute of Electronics, Information and Communication Engineers of Japan, 1990.

[44] J. Y. Noh, and U. Neumann, “A Survey of Facial Modelling And Animation Techniques,” *Technical Report, University of Southern California*, pp. 99-705, 1998.

[45] S. Krinidis, I. Buciu, and I. Pitas, “Facial Expression Analysis and Synthesis: A Survey,” *In Proceedings of the 10th international Conference on Human-Computer Interaction* vol. 4, pp. 1432-1436, June 22-27, 2003.

[46] F. I. Parke, “Techniques for Facial Animation,” *New Trends in Animation and Visualization*, pp. 229-241, 1991.

[47] M. K. o. Rolf, H. G. s. Markus, R. C. Friedrich *et al.*, “Simulating Facial Surgery Using Finite Element Models,” *In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, pp. 421-428, 1996.

[48] M. K. o. Rolf, H. G. s. Markus, and A. Bosshard, “Emotion Editing Using Finite Elements,” *Computer Graphics Forum*, vol. 17, no. 3, pp. 295-302, 1998.

[49] P. Kalra, A. Mangili, N. Magnenat-Thalmann *et al.*, “Simulation of Facial Muscle Actions Based on Rational Free Form Deformations,” *Proc. Eurographics ’92, Cambridge, U.K., Computer Graphics Forum*, vol. 2, no. 3, pp. 59-69, 1992.

[50] K. Khler, J. Haber, and H.-P. Seidel, “Geometry-based Muscle Modeling for Facial Animation,” *Proceedings Graphics Interface*, pp. 37-46, 2001.

[51] S. Platt, and N. Badler, “Animating Facial Expression,” *Computer Graphics*, vol. 15, no. 3, pp. 245-252, 1981.

[52] Z. Deng, P. Y. Chiang, P. Fox *et al.*, “Animating Blendshape Faces by Cross-Mapping Motion Capture Data,” *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 43-48, 2006.

[53] Q. Zhang, Z. Liu, B. Guo *et al.*, “Geometry-driven Photorealistic Facial Expression Synthesis,” *IEEE Transactions on Visualisation and Computer Graphics*, vol. 12, no. 1, pp. 48-60, 2006.

[54] G. Borshukov, D. Pipone, O. Larsen *et al.*, "Universal Capture - Imagebased Facial Animation for "The Matrix Reloaded"," *In ACM SIGGRAPH 03 Sketches & Applications*, 2003.

[55] E. Sifakis, I. Neverov, and R. Fedkiw, "Automatic Determination of Facial Muscle Activations from Sparse Motion Capture Marker Data," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 417- 425, 2005.

[56] B. Bickel, M. Botsch, R. Angst *et al.*, "Multi-Scale Capture of Facial Geometry and Motion," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 33, 2007.

[57] J. F. Blinn, "Simulation of Wrinkled Surfaces," *Proceedings of SIGGRAPH'78*, pp. 286-292, 1978.

[58] C. Oat, "Animated Wrinkle Maps," *In Advanced Real-Time Rendering in 3D Graphics and Games Course - SIGGRAPH*, pp. 33-37, 2007.

[59] Y. Zhang, T. Sim, and C. L. Tan, "From Range Data to Animated Anatomy-Based Faces: A Model Adaptation Method," *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*, 2005.

[60] F. Pighin, and J. P. Lewis, "Performance-driven Facial Animation," *SIGGRAPH Course Notes*, 2006.

[61] Q. Zhang, Z. Liu, B. Guo *et al.*, "Geometry-Driven Photorealistic Facial Expression Synthesis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 1, 2006.

[62] J. Davis, M. Agrawala, E. Chuang *et al.*, "A Sketching Interface for Articulated Figure Animation," *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 320-328, 2003.

[63] B. Gooch, E. Reinhard, and A. Gooch, "Human Facial Illustrations: Creation and Psychophysical Evaluation.," *ACM Transactions on Graphics* vol. 23, no. 1, pp. 27-44, 2004.

[64] E. Chang, and O. Jenkins., "Sketching Articulation and Pose for Facial Animation," *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006.

[65] O. A. Karpenko, and J. F. Hughes, "Smoothsketch: 3D Free-Form Shapes from Complex Sketches," *ACM Transactions on Graphics* vol. 25, no. 3, pp. 589-598, 2006.

[66] M. Lau, J. Chai, Y.-Q. Xu *et al.*, "Face Poser: Interactive Modeling of 3D Facial Expressions Using Model Priors," *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*,, pp. 161-170, 2007.

[67] P. Ekman, "The Argument and Evidence About Universals in Facial Expressions of Emotion," *In Handbook of Social Psychophysiology*, Wiley, 1989.

[68] G. Faigin, "The Artist's Complete Guide to Facial Expressions," Watson-Guptill, 1990.

[69] R. R. Provine, K. A. Krosnowski, and N. W. Brocato, "Tearing: Breakthrough in Human Emotional Signaling," *Evolutionary Psychology* vol. 7, no. 1, 2009.

[70] R. R. Provine, "Emotional Tears and NGF: A Biographical Appreciation and Research Beginning," *Archives Italiennes de Biologie*, vol. 149, no. 2, pp. 269-274, 2011.

[71] R. Parent, "Computer Animation: Algorithms and Techniques," Elsevier Science & Technology, 2008.

[72] N. Chentanez, and M. Müller, "Real-time Simulation of Large Bodies of Water with Small Scale Details," *SCA '10 Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 197-206, 2010.

[73] T. Pfaff, N. Thuerey, J. Cohen *et al.*, "Scalable Fluid Simulation Using Anisotropic Turbulence Particles," *Proceedings of ACM SIGGRAPH Asia 2010*, vol. 29, no. 6, 2010.

[74] N. Thürey, F. Sadlo, S. Schirm *et al.*, "Real-time Simulations of Bubbles and Foam Within a Shallow Water Framework," *SCA '07 Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 191-198, 2007.

[75] C. Yuksel, D. H. House, and J. Keyser, "Wave Particles," *ACM SIGGRAPH'07* vol. 26, no. 3, pp. 99, 2007.

[76] P. Hanrahan, and D. Sturman, "Interactive Animation of Parametric Models," *The Visual Computer*, vol. 1, no. 4, pp. 260-266, 1985.

[77] J. Osipa, "Stop Staring: Facial Modeling and Animation Done Right," Sybex, 2003.

[78] A. Menache, "Understanding Motion Capture for Computer Animation and Video Games," Morgan Kaufmann, 2000.

[79] R. Falk, D. Minter, C. Vernon *et al.*, "Art-directed Technology: Anatomy of a Shrek 2 Sequence," *ACM SIGGRAPH'04 Course Notes*, 2004.

[80] J. Schleifer, "Character Setup from Rig Mechanics to Skin Deformations: A Practical Approach," *ACM SIGGRAPH' 02 Course Notes*, 2002.

[81] K. Richie, O. Alexander, and K. Biri, "The Art of Rigging," *CG Toolkit*, vol. 2, 2005.

[82] R. Maccracken, and K. I. Joy, "Free-Form Deformations with Lattices of Arbitrary Topology," *Proceedings of SIGGRAPH 96*, pp. 181-188, 1996.

[83] B. Allen, B. Curless, and Z. Popovic, "Articulated Body Deformation from Range Scan Data," *Proceedings of SIGGRAPH 2002*, vol. 21, no. 3, pp. 612- 619, 2002.

[84] K. Singh, and E. Fiume, "Wires: A Geometric Deformation Technique," *In Proceedings of ACM SIGGRAPH 98*, pp. 405-414, 1998.

[85] P. G. Kry, D. L. James, and D. K. Pai, "Eigenskin: Real Time Large Deformation Character Skinning in Hardware," *In Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation*, pp. 153-159, 2002.

[86] M. Teichmann, and S. Teller, "Assisted Articulation of Closed Polygonal Models," *In Computer Animation and Simulation '98*, pp. 87-102, 1998.

[87] R. Wang, K. Pulli, and J. Popovic, "Real-Time Enveloping with Rotational Regression," *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[88] I. Baran, and J. Popovic, "Automatic Rigging and Animation of 3d Characters," *ACM Trans. Graph*, vol. 26, no. 3, pp. 72, 2007.

[89] K. Kahler, J. Haber, and H.-P. Seidel, "Geometry-based Muscle Modeling for Facial Animation," *In Proc. Graphics Interface*, 2001.

[90] V. C. T. Orvalho, E. Zacur, and A. Susin, "Transferring the Rig and Animations from a Character to Different Face Models," *Computer Graphics Forum*, vol. 27, no. 8, pp. 1997-2012, 2008.

[91] K. Waters, and F. J, "A Coordinated Muscle Model for Speech Animation," *Graphics Interface*, pp. 163-170, 1995.

[92] Y. C. Lee, D. Terzopoulos, and K. Waters, "Realistic Face Modeling For Animation," *SIGGRAPH Proceedings*, pp. 55-62, 1995.

[93] J. Y. Noh, and U. Neumann, "Expression Cloning," *Proceedings of ACM SIGGRAPH'01*, pp. 277-288, 2001.

[94] R. W. Sumner, and J. Popovic, "Deformation Transfer for Triangle Meshes," *Proceedings of ACM SIGGRAPH 2004*, vol. 23, no. 3, pp. 399-405, 2004.

[95] J. Ma, R. Cole, B. Pellom *et al.*, "Accurate Automatic Visible Speech Synthesis of Arbitrary 3d Model Based On Concatenation of Diviseme Motion Capture Data," *Computer Animation and Virtual Worlds*, vol. 15, pp. 1-17, 2004.

[96] H. Pyun, Y. Kim, W. Chae *et al.*, "An Example-Based Approach for Facial Expression Cloning," *In Proceedings of Symposium on Computer Animation*, pp. 167-176, 2003.

[97] D. Vlasic, M. Brand, H. Pfister *et al.*, "Face Transfer with Multilinear Models," *Proceedings of ACM SIGGRAPH 2005,* vol. 24, no. 3, pp. 426-433, 2005.

[98] E. S. Chuang, H. Deshpande, and C. Bregler, " Facial Expression Space Learning," *In Proceedings Of Pacific Graphics*, pp. 68-76, 2002.

[99] E. Chuang, and C. Bregler, "Moodswings: Expressive Speech Animation," *ACM Transactions On Graph,* vol. 24, no. 2, 2005.

[100] J. B. Tenenbaum, and W. T. Freeman, "Separating Style and Content with Bilinear Models," *Neural Computation,* vol. 12, no. 6, pp. 1247-1283, 2000.

[101] T. Ezzat, G. Geiger, and T. Poggio, "Trainable Videorealistic Speech Animation," *ACM Transactions on Graphics,* vol. 21, no. 3, pp. 388-398, 2002.

[102] S. Lee, S. Yildirim, A. Kazemzadeh *et al.*, "An Articulatory Study of Emotional Speech Production," *Proceedings of InterSpeech*, pp. 497-500, 2005.

[103] Z. Deng, U. Neumann, J. Lewis *et al.*, "Expressive Facial Animation Synthesis by Learning Speech Coarticulation and Expression Spaces," *IEEE Transactions on Visualization and Computer Graphics,* vol. 12, no. 6, pp. 1523-1534, 2006.

[104] C. Busso, Z. Deng, M. Grimm *et al.*, "Rigid Head Motion in Expressive Speech Animation: Analysis and Synthesis," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, no. 3, pp. 1075-1086, 2007.

[105] S. P. Lee, J. B. Badler, and N. I. Badler, "Eyes Alive " *In Proceedings of SIGGRAPH*, pp. 637-644 2002.

Vita

Alice J. Lin

She received her Master's Degree in Computer Science from the University of Kentucky, Lexington, KY. Her research areas are 3D computer animation and simulation, 3D computer modeling and algorithms, computer aided geometric design, computer graphics, image processing, and visualization.