

A SKETCH-BASED INTERFACE FOR PARAMETRIC
CHARACTER MODELING

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Nasheet Zaman

January 2006

© 2006 Nasheet Zaman

ALL RIGHTS RESERVED

ABSTRACT

We present an innovative modeling technique that allows artists to freely *sketch* changes to a realistically proportioned face mesh. Our concept builds a sketch-based interface over a parameterized, topologically optimized, rigged, textured, ready-to-animate prototype face model. Parametric modeling and sketch-based mesh editing are two previously disjoint areas of research we have merged to develop a unique system. The first component is the parametric setup. The prototype is marked with selectable anthropometric landmarks, which can be moved to locally reshape the geometry. As the user changes the positions of landmarks, facial measurements are recorded internally. Anatomical limits are calculated based on demographic data and figure drawing conventions. These limits may be visualized interactively to guide realistic modeling, or ignored by the artist for more stylized creations. The second component is the sketch-based interface, which streamlines the process of moving landmarks and reshaping the mesh. Using a tablet and stylus, the user first sketches a selection curve through some landmarks, and then sketches a second curve describing the desired shape of the region defined by the selected landmarks. The corresponding region of the mesh is automatically morphed in 3D space to conform to the new curve. During this process, the user can continue to use the parametric visualization mechanisms to maintain or exaggerate proportions. Thus, the artist can enjoy the freedom and simplicity of drawing to quickly design and generate a vast range of complex and original characters, using just one simple, intuitive, and familiar tool.

Biographical Sketch

The author was born on April 26, 1982 in Columbus, Ohio. As a first-generation Bangladeshi-American, Nasheet enjoyed the benefits and quirks of growing up with two languages and cultures. While her parents hoped to see her become a doctor or engineer, she enjoyed drawing and aspired to be a 2D animator. Despite being shy, she also loved the thrill of being on stage and acted in several plays, reinforcing her interest in the film industry. Meanwhile, her math teachers urged her to enter a technical field, and her English teachers encouraged her to cultivate her passion for creative writing. After graduating from Hilliard Davidson High School in 2000, she pursued double major in Computer Science and English at Cornell University with a special interest in 3D animation, which kept everyone happy.

At Cornell, Nasheet took electives in perception, art, photography, animation, and rendering to explore various aspects of computer graphics. She kept her love of the performing arts alive by singing with the *à cappella* group Tarana.

The summer after graduation in 2004, Nasheet traveled to Culver City, CA for an internship with Sony Imageworks. She returned to Cornell the following fall to begin her work at the Program of Computer Graphics, which culminated in 2006 with this thesis.

To all the sketchy characters out there.

And also, to my parents.

Acknowledgements

First and foremost, I would like to extend a heartfelt thanks to my advisor and mentor, Donald P. Greenberg for giving me this opportunity. It was an honor and pleasure to be guided by someone who not only broke the ground in computer graphics, but also continues to catalyze the growth of interdisciplinary work with his genuine excitement over new ideas. I look forward to future meetings with him and hope he will continue to inspire and motivate students as he has inspired me for many years to come.

Thanks to my minor advisor, Stan Taft, for his valuable advice on my research from the artist's perspective, and for helping me polish my painting skills. Thanks to Steve Marschner and Kavita Bala for providing me with a solid foundation in programming for graphics. It was nice to be surrounded by many great minds at the PCG, including Jim Ferwerda, Ken Torrance, Bruce Walter, and the PhDs, Milos, Ganesh, Piti, Jon, and Adam. Fabio Pellacini made sure the animation class ran smoothly and kept things interesting with his cynical sense of humor.

With my fellow Masters students I have shared the joys of TAing, summer softball, brainstorming, wrestling with LaTeX codes, competing to see who could stay awake the longest, and rescuing each other when keys were misplaced. I'll kill two birds with one stone by thanking the Jeffs, who have made my time

in the office more entertaining than watching paint dry: Jeff Wang, thanks for your fun one-liners, helpful and friendly attitude, and excellent taste in music; Jeff Budsberg, thanks for your enthusiasm, artistic advice, and drawing things on the whiteboard to explain things. Mark, thanks for being a good officemate and helping me lug home supplies from Lowes that were taller than me. The previous year's graduates, Jacky, Vikash, and Will, imparted valuable wisdom about how to prevent a neverending thesis. Best wishes to the new students, Brendan, David, and Todd.

A warm thanks to the friendly PCG staff. Linda patiently helped me sift through the archives at crucial times and kept track of important paperwork. Peggy welcomed us with a smile every morning and remembered everyone's birthday. Hurf came to the rescue and kept his sense of humor whenever Murphy's Law attacked technology in the lab. Martin helped when Murphy broke things outside Hurf's turf. Mary came in at 5 am every morning to keep the office sparkling for us, without getting too scared by the occasional grad student sleeping under the desk.

There are many friends who have made Ithaca my home for the past six years; I will leave Cornell with an abundance of great memories. The "Berts", Polly and Ramona, who helped me survive late nights in the CS Undergraduate Lab, Okies dinners, crazy taxi drivers, and many Cornell adventures, continue to be my nearest and dearest friends even after moving to the opposite coast. The frequent "Fellowship" reunions were great fun. Sally, Alex, and Andru shared my love of anything to do with graphics. Jer and Neal made sure I had a life during my first year as a grad; Saurav and Nick took over during the second year. Vysali has been a great friend for the past six years (the first at Cornell), and wonderful

roommate for the past two. My Collegetown housemates, Tianhui, Yimmui, and Naiting shared with me their culture and their delicious cooking. A special thanks to Karan for helping me move on short notice, and Maya for taking me in during my last week in Ithaca and showing me some great times in Downtown. I love you all.

My friends from Columbus have been there for me through all the ups and downs of my life and I cannot thank them enough for helping me get through the toughest obstacles in the last few months. Sharmeela, thanks for always being an optimist and "zapping" sense into me whenever I doubted myself. Shammi, thanks for being my favorite "partner in crime" for the past eleven years, and being near the phone whenever I needed to talk.

Naila, as our dad would say, "Don't be so...thanking your sister." Well, I "will be so." Thanks for making me laugh, keeping me young, and missing school to attend my thesis defense.

Finally, I extend a very special thanks to my parents, Rizia Zaman and Mohammad Harunuzzaman. They have worked hard over the years to support me in all my academic endeavors. This work would not have been possible without their love, support, prayers, and smart-genes.

This research was made possible by software donations from Autodesk, Inc.; funded by the the Program of Computer Graphics and the Department of Architecture at Cornell University, and the National Science Foundation Grant CCF-0205438.

Table of Contents

1	Introduction	1
2	Background	5
2.1	Human Uniqueness: Perception and Recognition	5
2.2	Art and the Human Form	11
2.2.1	Learning to Draw	11
2.2.2	Standard Proportions in Art	20
2.3	Anthropometry	22
2.4	3D Computer Modeling	28
2.4.1	Polygons	29
2.4.2	NURBS	31
2.4.3	Subdivision Surfaces	32
2.4.4	Scanned Models	32
2.5	The Animation Pipeline	33
2.5.1	Conceptualization: Art and Storyboarding	34
2.5.2	Modeling	35
2.5.3	Articulation: Rigging and Skinning	36
2.5.4	Animation	37
2.5.5	Shading and Lighting	37
2.5.6	Special Effects	38
2.5.7	Rendering	38
2.5.8	Compositing and Post-Production	39
2.6	Summary	39
3	Goals	41
3.1	Stating the Problem	41
3.2	Characteristics of an Ideal CG Interface	44
4	Previous Work	46
4.1	The Anatomy of a Prototype Digital Human	47
4.1.1	Deformable Face Topology	47
4.1.2	Applications of Animatable Prototypes	47
4.1.3	Fitting a Prototype to a Scan	49
4.1.4	Rigging prototypes	51

4.2	Maintaining Human Proportions in 3D	53
4.2.1	Parameterized Facial Models	54
4.2.2	Anthropometric Facial Modeling	57
4.2.3	Morphable Parameterized Scans	61
4.2.4	Parameterizing the Body	64
4.3	Sketch-Based 3D Modeling Interfaces	69
4.3.1	Freeform Design from Sketches	71
4.3.2	Sketch-Based Mesh Editing	73
5	The Prototype Model	77
5.1	Geometry	78
5.1.1	Face and Head	78
5.1.2	Body	80
5.2	Rigging, Deformation, and Parameterization	80
5.2.1	The Animation Controls	84
5.2.2	Parametric Modeling Setup	87
5.3	Texture	99
5.4	Summary	104
6	Implementation	105
6.1	Maya Architecture	105
6.1.1	The Model-Sketch Plugin	107
6.2	Landmarks and Parametric Visualization	108
6.2.1	Parameters	110
6.2.2	Landmarks	113
6.2.3	Parametric Model	118
6.2.4	Translating Anthropometric Data	118
6.3	Sketch-Based Modeling	125
6.3.1	Moving Landmarks	126
6.3.2	Curve Fitting	128
6.3.3	Landmark Selection Constraints	129
6.4	Summary	133
7	Changing Faces	134
7.1	Visualizing Contours	134
7.2	Procedure	136
7.3	Facial Features	139
7.3.1	Eyes	139
7.3.2	Nose	145
7.3.3	Lips	145
7.3.4	Face	145
7.3.5	Combining Parameters	149
7.4	Body Variation	151
7.5	Caricatures	151

7.6	Results	153
7.7	Performance	153
7.8	Limitations	163
8	Conclusions and Future Work	165
	References	173

List of Tables

3.1	Process of Animation in Different Media	43
6.1	Nodes of the Model Sketch system.	108
6.2	Commands of the Model Sketch system.	109
6.3	Commands of the Model Sketch system.	109
6.4	The Parameter File	122
6.5	The Parameters File: Joints	123
6.6	The Parameters File: Landmarks	124
6.7	The Parameters File: Measurements	125
7.1	Eye Spacing Limits	141
7.2	Eye Height Limits	142
7.3	Eye Protrusion Limits	143
7.4	Model-Sketched Eye Shapes	144
7.5	Model-Sketched Nose Profile Shapes	146
7.6	Model-Sketched Nostrils and Tip Shapes	147
7.7	Model-Sketched Lip Shapes	148
7.8	Model-Sketched Face Shapes	150

List of Figures

1.1	Cubic Tragedy	1
2.1	Newborns' preference for face-like structures	6
2.2	Diversity of human facial structure	7
2.3	Basic facial proportions	8
2.4	Caricatures	9
2.5	The Clinton-Gore Illusion	10
2.6	Learning to draw	13
2.7	Drawing error: A front-view eye in the profile view	16
2.8	Correction of the chopped off skull error	17
2.9	Portrait by Van Gogh	18
2.10	Using primitive shapes to draw the face	19
2.11	The Vitruvian Man	21
2.12	Artistic facial proportions	23
2.13	Farkas Anthropometric Landmarks Diagram	24
2.14	Anthropometric data chart from Farkas	25
2.15	Anthropometric Measurements Diagram	26
2.16	Farkas Distance measurements	27
2.17	Farkas Angle measurements	27
2.18	Types of 3D Surfaces	30
2.19	Diagram of the Animation Pipeline	33
2.20	Concept Sketches and Color Keys from Finding Nemo	34
4.1	A face mesh painted onto a volunteer's face for Parke's photogram- metric model	48
4.2	Parke's topologically optimized facial model	48
4.3	Cornell's Cyberware Scanner	50
4.4	Simulating facial muscles in Bibliowicz's system	52
4.5	Automated facial rigging by Bibliowicz	53
4.6	Parke's Parametric Face Model	56
4.7	Prototype face mesh used by DeCarlo	58
4.8	Diagram of DeCarlo's proportions	58
4.9	Faces generated by DeCarlo's system	60
4.10	Problems with reproducing the curvature between landmarks	61
4.11	Diagram of Blanz and Vetter's face morphing pipeline	62

4.12	Faces generated by Blanz’s system	63
4.13	Matching a 3D model to a photograph	63
4.14	Allen’s surface fitting algorithm	65
4.15	How Allen’s system handled areas that scan poorly	66
4.16	Texture transfer using Allen’s system	67
4.17	Transferring rigs between characters using Allen’s system	68
4.18	Diagram of Allen’s system	68
4.19	WACOM Intuos tablet	69
4.20	A WACOM Cintiq tablet in use	70
4.21	Gestures recognized by SKETCH	71
4.22	Teddy demo	72
4.23	Fitting a mesh to a curve exactly using Nealen’s system	74
4.24	Fitting a mesh to a curve approximately using Nealen’s system	75
4.25	Preserving local detail with Nealen’s system	75
4.26	Drawing creases and ravines in Nealen and Sorkine’s system	76
5.1	Our prototype facial mesh	79
5.2	Prototype model’s eyes, teeth, and mouth	81
5.3	Our prototype body mesh	82
5.4	Replacing poorly scanned areas with our prototype	83
5.5	Our Prototype Skeleton	85
5.6	The Gesture Sketch Harness	89
5.7	The Gesture Sketch Analogy	90
5.8	3D Views of the Gesture Sketch Harness	92
5.9	Effect of the wire deformer	93
5.10	Automated rigging using the GSH	94
5.11	Effects of the Sculpt Deformer	95
5.12	Landmarks on the prototype model	97
5.13	Weighting Landmark cluster deformations	98
5.14	Landmark based deformation	98
5.15	Frankie’s Unwrapped UV Map	100
5.16	Texture Variations for Frankie	101
5.17	Eye Colors for Frankie	102
5.18	Our Texturing UI	103
6.1	Attribute Editor view of an Anthropometric Parameter Node	112
6.2	Shortest Distance Measurement Visualization	114
6.3	Axial Distance Measurement Visualization	115
6.4	Angle of Incline Measurement Visualization	116
6.5	Landmark Node Display and Limits	117
6.6	Hypergraph view of Parameter Node	119
6.7	Hypergraph view of Landmark Node	120
6.8	Hypergraph view of Parametric Model Node	121
6.9	Curve Fitting Algorithm Illustration	127

6.10	Problem caused by skipping landmarks in the selection curve . . .	131
7.1	GSH Facial Features	135
7.2	GSH visualization of newly created characters	136
7.3	Model-Sketch Example: Nose	137
7.4	Model-Sketch Example: Ear	138
7.5	Variations to Frankie’s Body Shape	152
7.6	Creating 3D Caricatures using Anthropometric Limit Visualization	153
7.7	Example caricatures created using anthropometric guides	154
7.8	New Faces	155
7.9	New Faces	156
7.10	New Faces	157
7.11	New Faces	158
7.12	New Faces	159
7.13	New Faces	160
7.14	New Faces	161
7.15	All Faces	162

Chapter 1

Introduction



Figure 1.1: How can we prevent CG modeling tragedies? [WC05]

At SIGGRAPH 2005, an audience of professionals, researchers, artists, programmers, and students in computer graphics howled with laughter during *Cubic Tragedy*, a clever computer animated short produced by Chun-Wang and Ming-Yuan Chuan of the National Taiwan University of Science and technology [WC05].

The short depicted a young woman, crudely modeled from polygons, playing with a "makeup kit" of polygonal modeling tools that are found in most standard 3D software packages. Hilarity ensues when she applies various strange-looking tools to her face and they work in unpredictable, unintuitive ways, mangling her face to the point where the makeup kit's "Undo" button could not save her. Despite the fact that she had cutting-edge technology in her hands, she ended up looking like a 1973 Picasso painting.

Knowing laughs came from every artist who has ever been frightened off by CG software, from every programmer who could see the underlying flaws in the implementation of the "makeup kit." More than anything, the film brought to light what many people in the field already know from experience: Modeling tools are powerful, but unpredictable. Many sophisticated tools are available for use, however their functionality is not always clear, and due to the sheer number of tools, the most useful tools are often buried deep within the interface. Consequently, creating a complex 3D model such as a human character requires not only artistic talent and a basic knowledge of anatomy, but also the ability to navigate an equally complex user interface.

The ever-growing popularity of CG films, special effects, and video games has increased the demand for high-quality digital characters. In recent years, 3D character animation has reached impressive levels of realism. Digital stunt doubles in live-action movies are sometimes indistinguishable from the real actors. Not surprisingly, the labor-intensive process of creating these characters requires multidisciplinary skills. Animation and gaming companies face a common dilemma when hiring character modelers: Do they hire creative, exceptionally talented artists who are unfamiliar with CG software, or do they hire computer whizzes who can

navigate the software with confidence, but have not had the opportunity to polish their artistic skills? In using 3D software, a user with exceptional creativity may not be able to fully realize his or her vision before a deadline if hindered by a frustrating lack of control over the medium.

In practice all artwork, 2D and 3D, generally begins with a sketch. The primary advantage of drawing is that all the work is done with one intuitive tool: a pencil. What makes a pencil intuitive? It is the fact that we know exactly what to expect when we drag a pencil across paper. A line will appear on the paper, starting where we put the pencil down, ending where we lift it. We can feel and hear the friction of the lead on paper as we draw.

That may seem obvious, but what happens when we use CG software to create a model? The user sees a 3D virtual object on a 2D display, and must use 2D mouse movements that translate into 3D changes to the model. The mouse is physically far away from the virtual object being manipulated, the object cannot be touched and thus there are no haptic or auditory cues to guide the user. It is not always obvious how the mouse movements will be interpreted. The mouse forces the user to use "clumsy" arm movements, which are *gross motor skills*, in lieu of *fine motor skills* to create details. Thus, 3D modeling suffers from an inherent lack of the hands-on intuitiveness we take for granted when using a pencil.

However, we cannot demean the sophisticated visualization capabilities of 3D software. After all, drawing would be easier if one could rotate the objects, zoom in and out on the paper, copy parts of the drawing, and make dramatic edits without losing hours of work. A 3D viewer allows the artist to visualize structures clearly from various angles, rather than relying on a single "flat" interpretation. In 2D, the process of conveying depth and making an object "look" 3D requires much

practice, talent, and keen observation of light and shadow. In CG, an artist can enjoy automatic rendering calculations to generate these complex effects.

Some recent CG software advances allow the user to generate a unique digital human in a matter of minutes. Parametric modeling interfaces such as Poser™ [EFr] provide a simple, intuitive controls for reshaping a pre-existing character model. By changing basic parameters of the height, and weight, and facial features, typically via sliders, a variety of models can be created. The main complexity of such systems is the sheer multitude of controls that a user must sift through before finding the desired attributes. Also, one could argue that sliders are restrictive and detract from the artistry of modeling.

We propose a new approach to reducing the complexity of standard CG modeling interfaces by bringing users back to the basics: paper and pencil. Using a WACOM Cintiq tablet and stylus [Cin], we simulate the sense of freedom one enjoys when making quick sketches in a sketchbook, while exploiting desirable features of a 3D viewer and parametric modeling controls.

Our approach is based on the observation that all humans, across different cultures, are remarkably similar both in anatomical makeup and their ability to move. The goal of our system is to help artists visualize the anatomical proportions of a human character without being too rigid, and giving them desired levels of artistic control over the form. We free the user from having to worry about the topology of the model by providing a rigged, animatable human prototype character as a starting point. The prototype, which is a detailed subdivision surface mesh created using standard 3D techniques, is equipped with measurement mechanisms to interactively visualize proportional relationships and can be reshaped using our fast, intuitive sketch-based interface.

Chapter 2

Background

2.1 Human Uniqueness: Perception and Recognition

The human eye is innately programmed to recognize faces. According to a 1991 experiment by Morton and Johnson, newborns are more intrigued by face-like stimuli than other types of visual stimuli [JM91]. Using the three plates shown in Figure 2.1, the experiment measured the amount of time an infant would gaze at each plate. Invariably, infants fixated on the plate with a face-like configuration. From the moment we are born, we know that a face is something with eyes, a nose, and a mouth. We know that eyes must be next to each other and above the nose, which is above the mouth.

After newborns develop a mental "blueprint" for a face, and even before they learn to speak, they learn to recognize individual faces. The surprising truth is that despite the colorful diversity of the human race, which is represented beautifully in Figure 2.2 [CRE], the proportions of all faces are remarkably similar. The eyes are approximately at the halfway point of the face. The nose is at the halfway point between the eyes and the bottom of the nose. The pupils line up with the corners

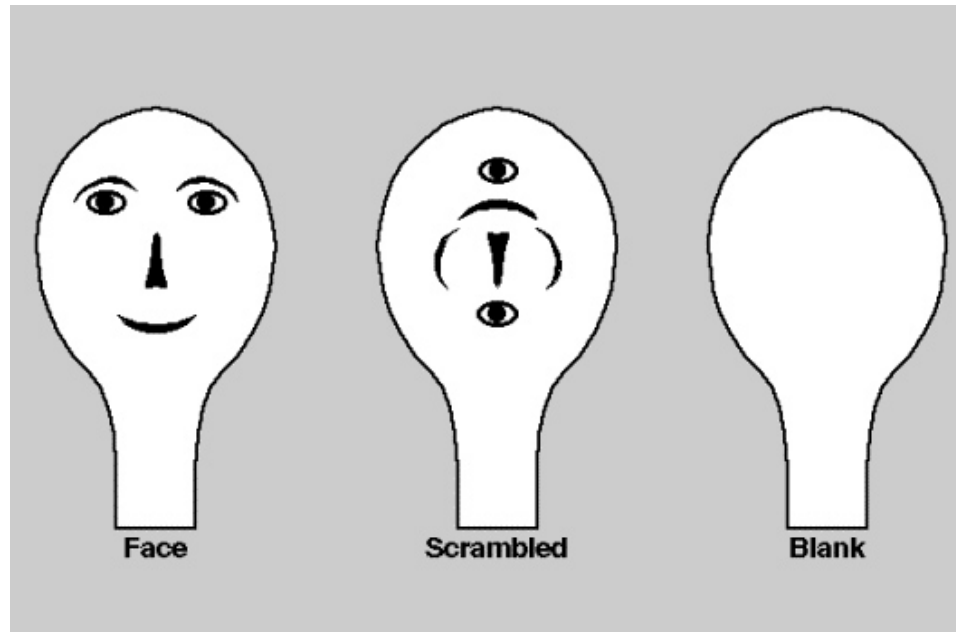


Figure 2.1: When Morton and Johnson showed these plates to a number of newborns, they observed that babies tend to focus on the left plate. They concluded that humans have an innate preference for face-like structures. [JM91]

of the mouth (Figure 2.3). A millimeter of difference in the size of the nose or the arch of an eyebrow is enough to tell us whether a person we glance at on the street is a close relative or a complete stranger. Thus, when we learn to recognize individual faces, we have achieved something quite amazing.

Recognizing individual faces requires an acute awareness of subtle variations. Effectively, the subconscious makes note of all *distinctive* features whenever a new face is seen. Are the eyes far apart or close-set? What is the shape of the face? Is the nose larger or smaller than average? Numerically quantified, the differences between two faces may be minute, but since the human eye is fine-tuned to notice the subtlest facial details, our perception exaggerates the differences. This is why caricatures are often more recognizable than the actual person drawn; the artwork does exactly what the brain does: exaggerates distinctive features. In other words,

**7 million
people,
300 languages,
14 faiths,
1 London.**

Simon, Southgate Paul, Camden Town Anjana, Whitechapel Simon, Hoxton Corina, Brixton Adam, Bethnal Green

Teepika, Southwark James, Brixton Corin, Maida Vale Yoko, Elephant

Henry, Park Royal Frank, Leyton Kasia, Borough

Stephen, Neasden Anthony, Soho Phovita, Manor Park Khaled, Lancaster Gate

Ahmed, Tufnell Park Neil, Chiswick Nadia, Fulham David, Stamford Hill Yvette, Shepherd's Bush Miah, Somers Town

Julia, Peckham Rye Barwal, Kings Cross Junior, Richmond Magdalena, Notting Hill Nhung, Chesham Road Eva, Mile End

Ignacio, Stamford Hill Mai, Tooting Bec

London's cultural, racial and religious diversity has always been its strength. Mayor Ken Livingstone and the CRE call on all Londoners to use that strength. Terror respects no one – let's not forget, more than 600 Muslims were killed on September 11th. And Islamophobia and racist attacks create exactly the climate of fear terrorists want to spread. Let's show that they have no place in London.

COMMISSION FOR RACIAL EQUALITY GREATER LONDON AUTHORITY

Figure 2.2: The vast variety of human faces is illustrated in this poster by the Commission for Racial Equality in London [CRE]

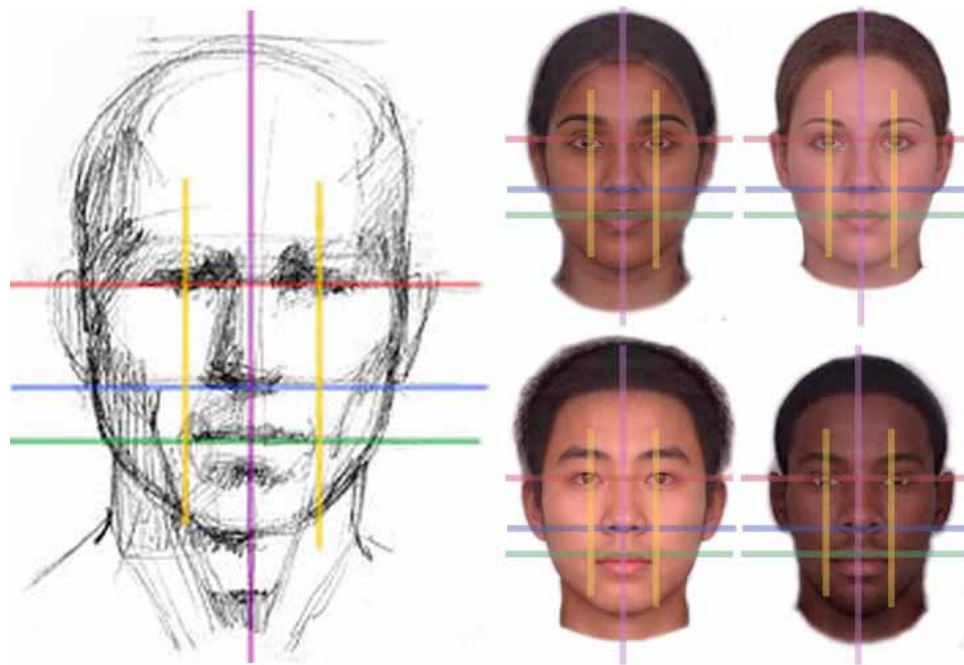


Figure 2.3: The drawing on the left is from a portrait drawing tutorial by J.R. Dunster [Dun03]. The faces on the right are blended prototype faces representing Canadian college students of different ethnicities. These prototypes are from research on facial preferences conducted by [Fac]. Superimposing the proportions diagram on each face illustrates their similarity.

our perception magnifies the deviation from the *average* face.



Figure 2.4: Making his big eyes bigger, pointy eyebrows pointier, and pronounced wrinkles deeper makes the caricatured Mr. Bean more recognizable in our perception. The photograph: [Beab]. The caricature: [Beaa]

But what does the "average" face look like? Every face seen by an individual, from birth, becomes part of his or her definition of an average, or **prototype** face. The most important structure that influences our definition of "average" is our own face. Because we see it every day, naturally, we evaluate other faces in relation to it.

A person's definition of a prototype face is also directly dependent on the types of people he or she has seen while growing up. A person raised in a community with a single primary race will be able to easily distinguish between people of that race, but will have difficulty telling the difference between two people of another race. To a person who grew up in China, for example, all Caucasian faces appear to have more prominent noses and larger eyes than the prototype Chinese face even if the measured difference between two individual Caucasian faces is significant. This is because the two Caucasian faces differ from the Chinese prototype in the

same direction, so the person only sees the fact that the faces deviate in that direction, without paying much attention to how much they deviate. Many facial recognition systems work the same way, and for this reason, many have protested that they improperly identify members of a minority race.

Facial structure is the most important factor in recognition. After a certain age, bone structure does not change. However, height, weight, coloring (of the skin and eyes), resting facial expression, age, and hair properties are also significant. A humorous example of how hair alone can transform our perception of a face is shown in Figure 2.5. The picture appears to show two familiar politicians.



Figure 2.5: Who are these men? [Ren]

A closer look reveals that the faces of the two men are exactly the same. For an artist to create a successful character, secondary characteristics are significant.

Our perception of other body parts is much weaker than facial perception. We generally cannot recognize a person by looking at their hands or feet. We tend to look at the body holistically, in terms of height, weight, and body types: pear, apple, triangle, or hourglass. One reason we do not rely largely on body types to recognize individuals is that unlike the face, the body's appearance is influenced primarily by the way that muscle, fat, and skin fit over the bones rather than the bones themselves, and as we learn during the holidays, these characteristics can change.

So far, we have only considered the static visual characteristics that allow us to recognize a person. Style of motion and voice are dynamic characteristics, i.e., characteristics that vary over time, that have also been proven significant factors. To create a truly successful character in a 3D animation, dynamic properties must also be taken into account.

From facial proportions to style of movement, we can see that every characteristic described in this section can be quantified as a **parameter** that is used in conjunction with other parameters to define a unique individual.

2.2 Art and the Human Form

2.2.1 Learning to Draw

Drawing is the most natural and native form of visual communication, and can be considered a universal language. Elaborate cave paintings appeared centuries before the first written word. The earliest letters were pictograms, or pictures that

represent words or syllables; eventually they were simplified to form the alphabets used today. Children learn how to draw before they learn how to write. The ability to draw is essential to communication; "a picture is worth a thousand words."

Picking up a pencil for the first time opens a world of possibilities, giving a child a tangible sense of control. There is a certain fascination in seeing a mark appear on paper in a way that follows the exact motion of the hand. The doodles of three year old children are free and expressive, but reveal a desire to exert even more control over the form. Slowly, the doodles become shapes, and the shapes become recognizable forms. (Figure 2.6)

At a certain point, children acquire the hand-eye coordination necessary to produce art. What then separates the artist from the doodler is the ability to observe and reproduce what he or she sees in a way that mimics the way a viewer would perceive the actual object.

However, many people never reach this stage. Once a child's fine motor skills become sufficient for handwriting, any further art training becomes optional.

Renowned art instructor Betty Edwards sought to correct many of the common problems faced by novice artists. She witnessed many intelligent, successful adults, even those who worked in highly detail-oriented jobs, becoming frustrated with their juvenile drawing skills. According to Edwards, the problem was not in the way they moved the pencil, but in the way they had been trained to process visual information. Elementary neurology describes the problem.

The theory is based on the concept of left and right-brained thinking. Edwards describes the left-brained mode, or L-mode, of perception using the following terms [Edw79]:

- *Verbal* - Uses words and names to define objects.



Figure 2.6: These portraits were done by (clockwise from the top left) a two-year-old, a four-year-old, an eight-year-old, and a sixteen-year-old. Notice the circular forms in the two-year-old's drawing, the symbolic representations of features in the eight-year-old's drawing, and the level of detail of the sixteen-year-old's drawing. [Edw79]

- *Symbolic* - Uses simplified representations of complex stimuli, discarding the unimportant.
- *Linear* - Sees things in discrete pieces.

In contrast, the right-brained mode, or R-mode processes the same information in the following ways:

- *Nonverbal* - Senses relationships and ideas without words.
- *Actual, Real* - Perceives things as they are, without breaking them down into symbolic components.
- *Holistic* - Sees the *Big Picture* rather than the parts.

L-mode thinking is extremely efficient, as it gathers only the minimal amount of information needed to remember a given stimuli. However, when it comes to drawing, this efficiency works against the artist. To draw an object realistically, one must be able to see the whole, paying equal attention to the *unimportant* parts. Some common manifestations of left-brained efficiency include:

- **Symbolic drawing**

What is this object?



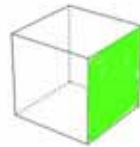
You guessed correctly: It is an eye. But look at it next to a realistically drawn eye.



They are not very similar. How do we immediately recognize the football-and-circle shape as an eye? It is a result of L-mode symbolism. When we see an actual eye, we first process the outlines and basic shapes. All the shading, highlights and subtle variations of the silhouette are secondary. Thus, when we draw, we try to be as concise as possible, communicating the minimal amount of information needed to tell a viewer that we have drawn an eye.

- **Faulty perspective and simplified views**

What is the shape of the highlighted face of the cube in the picture below?



If you said it is a square, you are only partially right. In 3D, it is indeed a square. However, to draw it in 2D, one must recognize it as this shape:



Often, knowledge about an object interferes with one's ability to draw it. The articulate left brain says, *this is a cube*, and draws the logical conclusion that *all the faces are squares*. This knowledge and verbalization causes the brain to mentally rotate more complex objects to the most descriptive views. It is not uncommon to see portraits where facial features are rotated to the "clearest" view, regardless of the face's global orientation. Shown in Figure 2.7 are two portraits illustrating this problem.

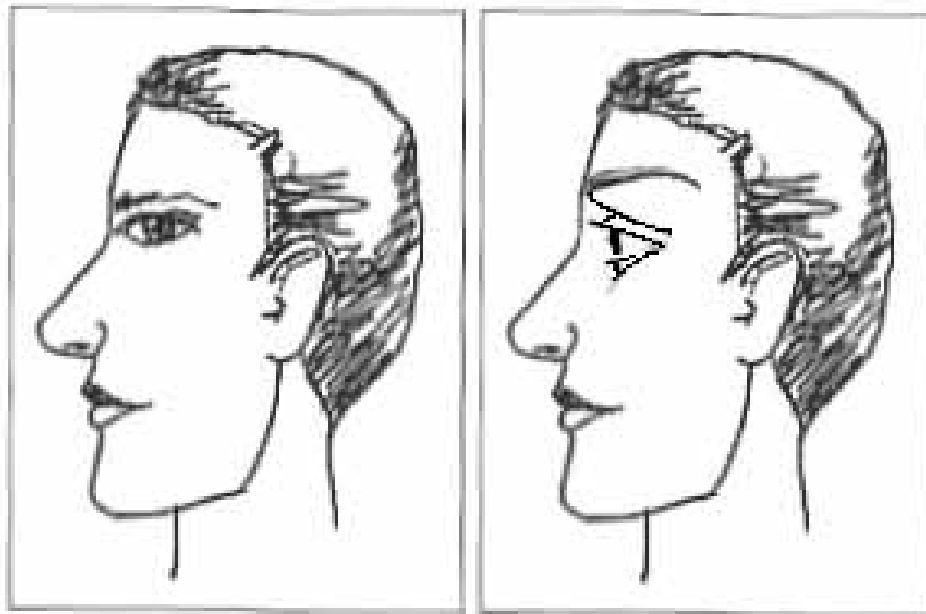


Figure 2.7: In the drawing on the left, the eye looks as it would in the front view even though the face is drawn in profile. In 3D, it would look as if the person has fish eyes! The image on the right corrects the problem. Modified from [Dun03]

- **Distorted proportions**

Perceptually, the face is the most important part of the body. In essence, a larger part of the brain is devoted to recognizing the face than other parts. Thus, as we saw in Figure 2.6, it makes sense that children generally draw heads larger than the body. Heads tend to be smaller in adult's drawings;

however, they are often still too large in proportion. Also, facial features, particularly the eyes, tend to be drawn larger than they actually are. Once again, this is a consequence of L-Mode thought discarding "unimportant" information in favor of the most compelling details.

Proportion-blindness happens to the best of us; for example, the famous portraits below show the infamous *chopped off skull error* (Figure 2.8 and Figure 2.9). The part of the head covered with hair is uniform and "boring" compared to the fascinating eyes and other features. However, if a drawing fails to include the skull at its full size, our right brain will know that something is wrong with it without being able to describe it in words.

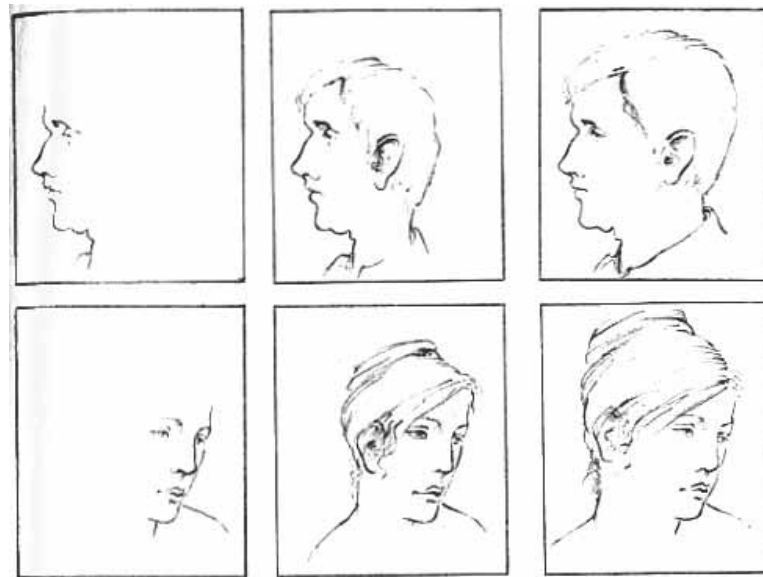


Figure 2.8: These pictures shows that the amount of space the top of the skull occupies is significant, even if we do not perceive it. The drawings in the middle column show erroneous interpretations of the skull's size; the drawings on the right correct the problem. Note that the facial features were not changed in any way. [Edw79]

Using Edwards' theory, J.R. Dunster offers some advice that uses the L-mode to simulate R-mode visualization [Dun03].

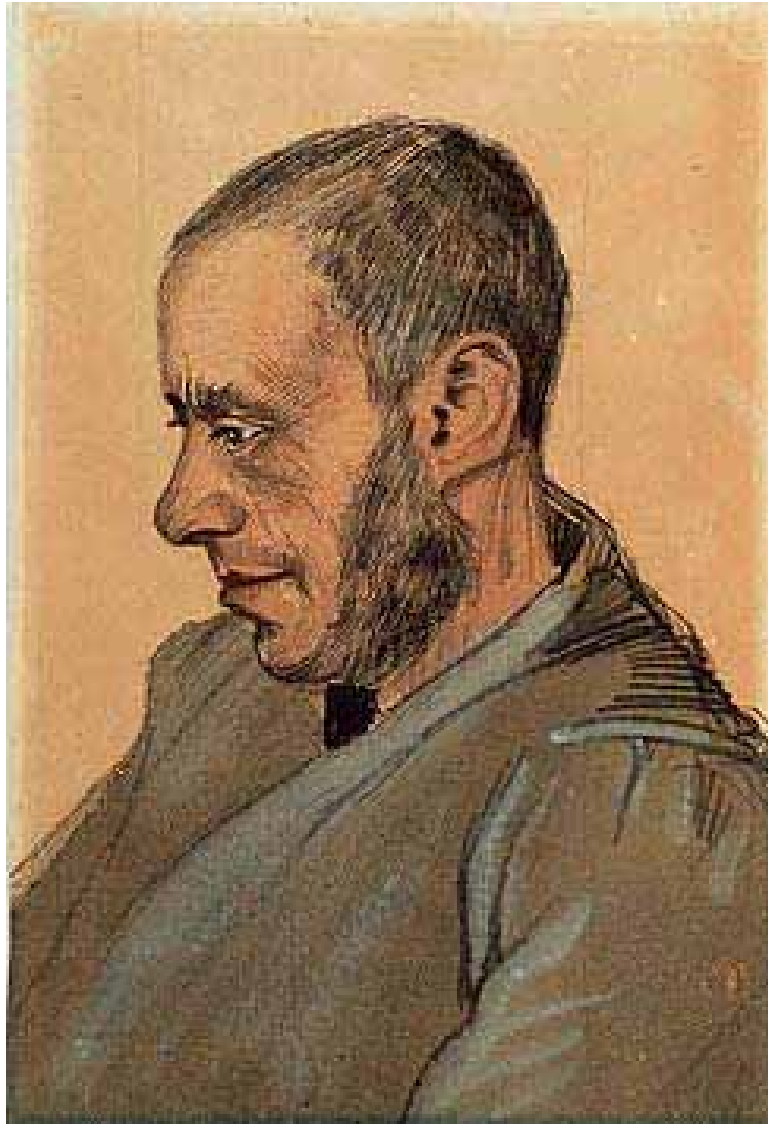


Figure 2.9: Even Van Gogh has committed the "chopped off skull error."
[Gog]

- Use primitive shapes

It is easier to view the proportions of an object and the relationships between the parts if one focuses on the general shapes and tries to ignore the details. This observation is integral to the process used by artists in **figure drawing**. Afterward, the details can be drawn around the primitives, as seen in Figure 2.10.

- Memorize basic proportions

For the common problematic proportions, one must simply force oneself to accept that the real proportions are different from what they *see*. Memorizing proportions exploits the L-mode's numerical, logical aptitude.

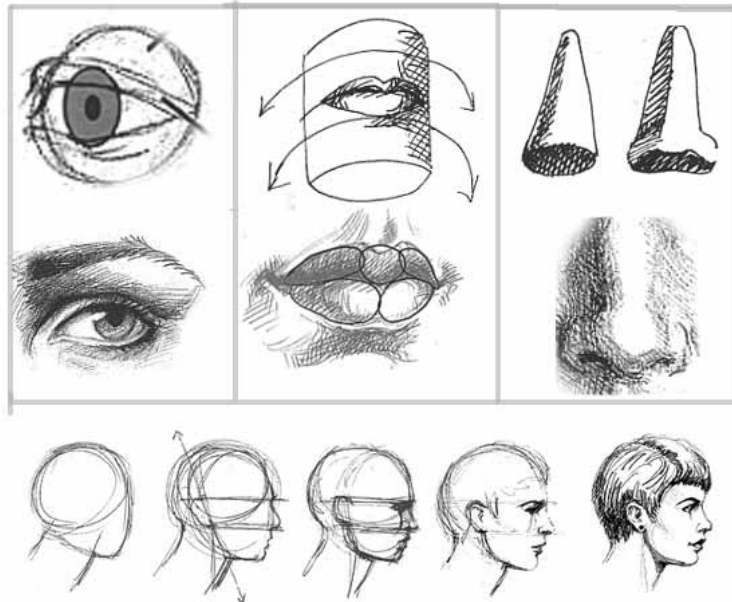


Figure 2.10: Using primitives to visualize the correct 3D contours for the facial features and head. [Dun03]

We will discuss how these concepts and tips influence 3D art in Section 2.4. In Chapter 6, we will show how we implemented some of these concepts in the design of our interface.

2.2.2 Standard Proportions in Art

Life drawing studies are an essential form of practice for art students. Students must learn basic anatomy in addition to developing keen observation skills. By watching a model pose in several positions in short spans of time and trying to reproduce the poses, artists gain a sense of how the body moves and deforms, how the muscles flex under the skin, and how the bones contribute to the underlying shape. When an artist develops this sense of anatomy, and knows generic proportion measurements, he or she will be able to draw *gesture sketches* in a few seconds and then finish the drawing without the model present. But where does one begin in learning these "magic numbers" that make a drawing look right?

Leonardo Da Vinci's famous Vitruvian Man (Figure 2.11) illustrates standard proportions of the human body. Da Vinci measured the entire body by using **cubits** as the units of measurement. One **cubit** is the length of an arm from the elbow to the thumb.

DaVinci derived the following measurements:

- 4 fingers = 1 palm
- 4 palms = 1 foot
- 6 palms = 1 cubit
- 4 cubits = man's height

These rough proportions or variations have been applied by artists for generations. The head is another commonly used unit of measure; the body is between seven and eight heads tall for an adult. Recall the facial proportions in Figure 2.3, and those shown in Figure 2.12.

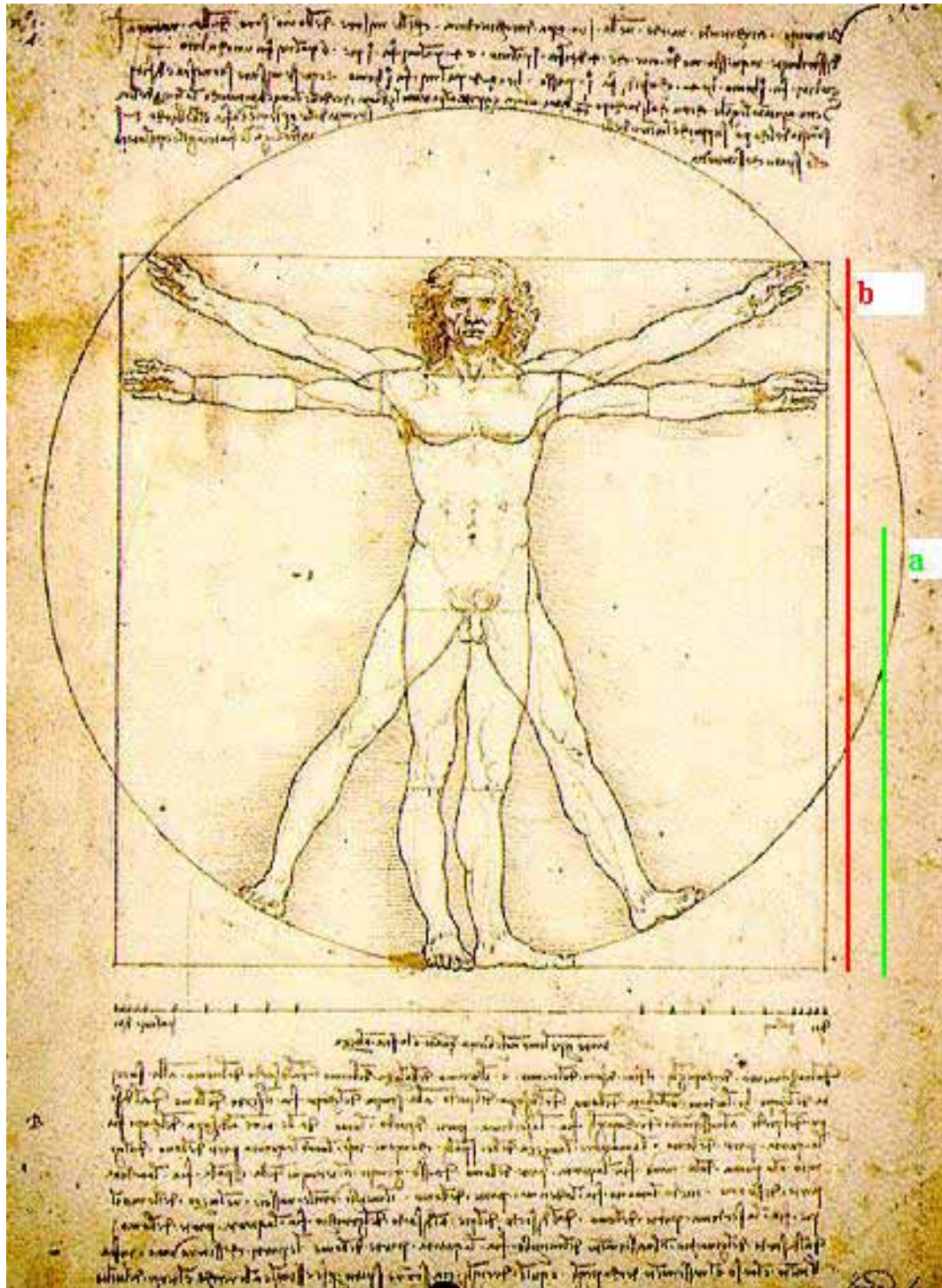


Figure 2.11: Da Vinci's definition of proportions.[Vin]

- Eyes are located at the halfway point of the skull.
- A face is five eyes wide.
- The space between the two eyes is the size of one eye.
- The corners of the mouth line up with the vertical lines drawn from the pupils.
- The nostrils are as wide as the space between the eyes.
- The bottom of the nose is halfway between the eyes and the lowest point of the chin.
- The lips are halfway between the bottom of the nose and the chin.

Rough proportions are a good starting point. The artist can then use his or her keen observation and interpretation skills to derive the rest of the form. In some applications, however, the rough proportions are not enough, and it is necessary to have exact measurements. In Section 2.3, we will discuss the scientific approach to measuring the body.

2.3 Anthropometry

Anthropometry is the science of human body measurement, based on important **landmarks** of the body. Anthropometric landmarks are a subset of anatomical landmark that is limited to those points that have a visible effect on the form, for example, the corners of the eyes. The locations of internal organs such as the heart and lungs are not considered anthropometric.

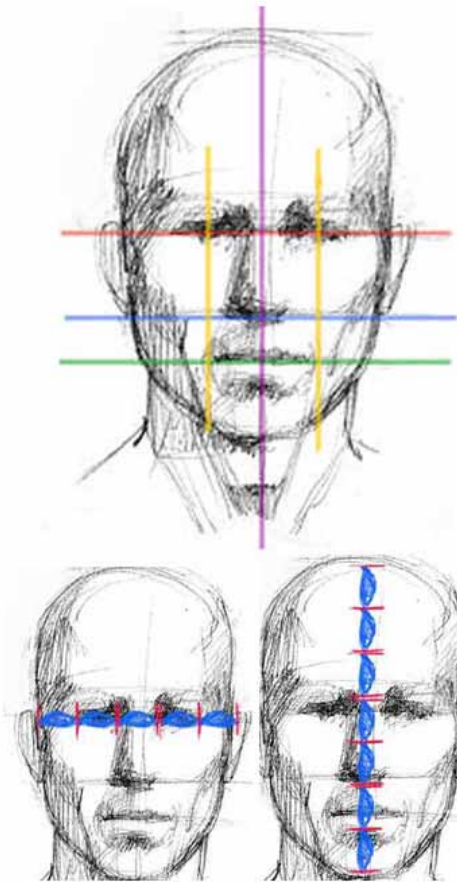


Figure 2.12: Artistic facial proportions by Dunster. [Dun03]

Each anthropometric landmark is identified by an abbreviation for its anatomical term. For example, in Figure 2.13, "or" stands for *orbitale*, which is the lowest point of the eye socket, and "t" stands for *tragion*, which is the base of the ear.

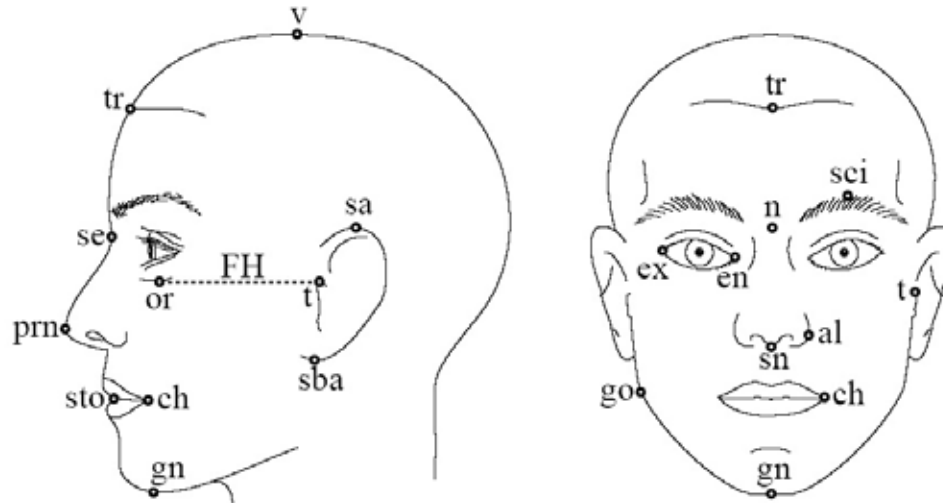


Figure 2.13: Anthropometric landmarks of the face.[Far94]

The foundation for our parametric modeling system is a well-known, comprehensive resource on traditional anthropometry; *Anthropometry of the Head And Face: Second Edition*, edited by Leslie G. Farkas [Far94]. This textbook, primarily catered to the field of plastic surgery, provides facial anthropometry diagrams (Figure 2.14), detailed definitions of each facial landmark, instructions for taking measurements of human subjects with photographs of actual subjects being measured, and a reference chart of 128 facial measurements, along with the mean and standard deviation, for over 100 Caucasian males and females ages 0-25. The book also provides some data for Asian and African-American subjects.

In order to compare anthropometric data for various subjects, all subjects were measured in the same standard pose. For facial measurement, the Frankfurt horizontal is used as a guide (abbreviated as FH in Figure 2.13). The FH is a line from

FACE

TABLE A-11-26. Depth in the maxillary region (sn-1, 1-jah) and left (mgn)

Age (years)	Male, right side			Female, right side		
	N	Mean	SD	N	Mean	SD
1	18	95.3	2.0	20	91.3	2.6
2	30	96.4	3.1	32	95.1	3.0
3	30	98.7	3.7	30	97.0	2.2
4	30	105.0	4.0	30	100.3	2.8
5	30	106.1	3.2	30	103.1	3.0
6	50	107.5	4.3	50	104.3	3.2
7	50	109.3	4.0	50	106.2	3.1
8	51	111.2	3.7	51	107.2	3.7
9	51	112.4	4.5	50	109.5	4.2
10	50	114.2	4.6	49	111.4	3.2
11	50	116.4	4.3	51	113.1	4.0
12	52	118.4	4.8	53	115.1	4.2
13	50	118.9	5.0	49	116.3	4.6
14	49	121.4	5.6	51	118.3	3.7
15	50	124.5	5.0	51	117.9	4.8
16	50	125.8	4.5	51	118.2	4.4
17	49	127.6	4.1	51	117.5	4.1
18	52	125.3	5.0	51	117.9	4.3
19-25	109	133.0	4.6	200	120.5	3.8

Age (years)	Male, left side			Female, left side		
	N	Mean	SD	N	Mean	SD
1	18	95.2	1.9	20	91.4	2.8
2	30	96.7	3.2	32	95.2	3.1
3	30	98.6	3.2	30	97.1	2.3
4	30	104.6	4.0	30	100.4	2.9
5	30	108.0	3.5	30	102.9	3.2
6	50	106.6	4.5	50	104.5	3.4
7	50	109.1	3.6	50	106.5	3.2
8	51	111.0	3.8	51	107.5	4.0
9	51	111.9	4.2	50	110.0	4.2
10	50	113.2	5.0	49	110.6	3.2
11	50	116.0	4.3	51	112.8	4.7
12	52	118.2	4.8	53	114.9	4.0
13	50	118.2	5.0	49	116.0	4.5
14	49	121.4	6.0	51	118.3	3.6
15	50	124.7	5.0	51	117.9	3.8
16	50	125.1	4.6	51	117.1	4.5
17	49	126.7	4.3	51	117.6	4.0
18	52	125.8	4.9	51	118.3	5.2
19-25	109	131.8	4.3	200	119.3	4.0




Figure 2.14: A typical chart found in the anthropometric inventory. Note that measurements are recorded for both the left and right sides. The mean and standard deviation are recorded. [Far94]

the *orbitale* (or) to the *tragus* (t), which is the highest point on the flap of cartilage at the base of the ear. The subject must position his or her head such that the FH is parallel to the ground in order to guarantee a standardized measurement.

Once the landmarks are located on a subject's face, Farkas uses five types of measurements to characterize the relationships between the landmarks (Figure 2.15). Figures 2.16 and 2.17 show how these measurements are taken from actual subjects:

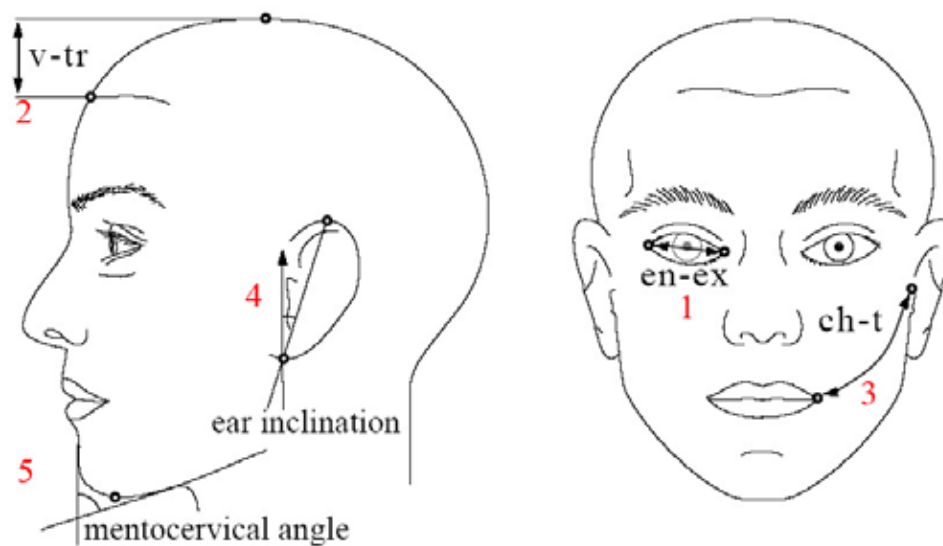


Figure 2.15: Types of Anthropometric Measurements. [Far94]

1. **Shortest Distance** between two landmarks.
2. **Axial Distance** The distance between two landmarks along one of the three canonical axes. The head must be in FH position for these measurements, which means the plane defined by the bottom of the eye sockets and the middle of the ear must be parallel to the ground.
3. **Tangential Distance** The distance between two landmarks measured along the surface of the face.

4. **Angle of Incline** The angle formed by one of the canonical axes and the ray between two landmarks. Again, the head must be in FH position.
5. **Angle Between** The angle between the tangents of two landmarks.

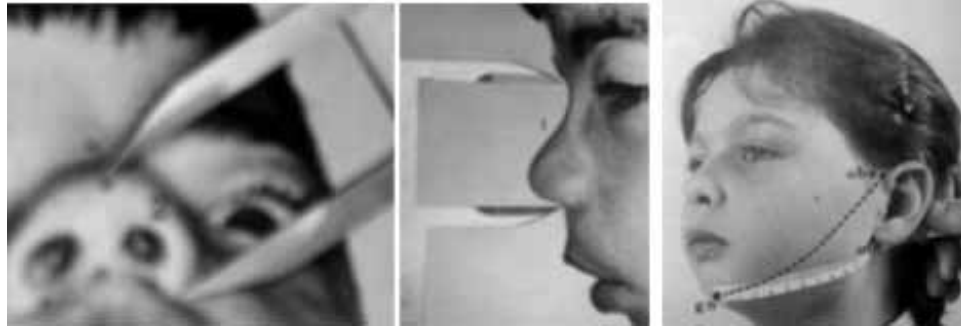


Figure 2.16: From Left: Shortest Distance, Axial Distance, and Tangential Distance. [Far94]

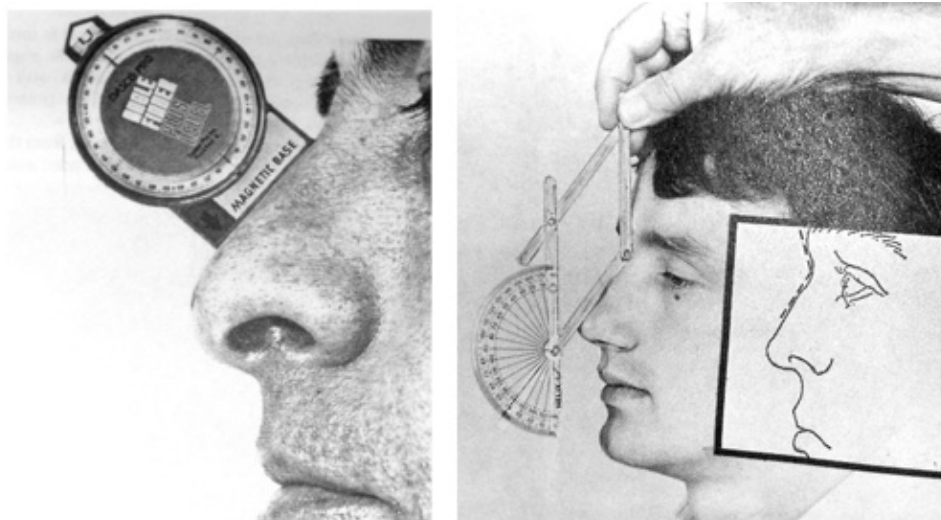


Figure 2.17: From Left: Angle of Incline and Angle Between. [Far94]

Note that most measurements are dependent on two landmarks. Some tangential measurements depend on three landmarks, but they can be split into two measurements. In our parametric modeling system, described in Chapter 6, we require that each anthropometric parameter be defined by exactly two landmarks.

Anthropometric data for the body is used in ergonomics and apparel design. NASA uses the measurements to ensure the safety and comfort of their astronauts, and has published an anthropometric database online [NAS]. Unlike facial anthropometry, body anthropometry relies on large-scale measurements, such as the height of the leg, rather than small-scale measurements, such as the bumps on the knee. Also, body measurements are taken in various poses. To design a comfortable chair, a company will be interested in the length of the thigh in the horizontal direction when the subject is sitting, whereas a clothing company designing pants would need a vertical measure of the height of the leg.

While these measurements were all taken manually, technological advances in the field have allowed the creation of a larger dataset based on 3D scans. The Civilian American and European Anthropometry Resource (CAESAR) dataset is a large source of 3D full-body scans of males and females ages 18-65 of various races and body types [CAE]. Landmarks are manually located on the scans and the anthropometric data is calculated accordingly.

2.4 3D Computer Modeling

This section and Section 2.5 serve as primers in computer animation and the animated film production pipeline, respectively, for those who are not familiar with the process. Knowledge of the traditional process of 3D computer modeling, and how modeling fits into the "big picture" of the animation pipeline, is essential to understanding the significance of the prototype parametric sketch-based modeling system we have developed. Readers with experience in the field of computer graphics might find these sections too elementary and may prefer to skip ahead to Section 2.6.

We have discussed the challenges of representing a 3D object in 2D. Since the late sixties, the idea of 3D computer modeling blossomed, and in the last decade its popularity has skyrocketed.

A 3D object, although defined geometrically in 3D, is only virtually 3D. At any given time it is simply a flat image on a computer screen. The 3D mesh is stored within a computer as a list of discrete points or vertices with coordinates along horizontal, vertical, and depth axes. A **camera** is defined as another coordinate in space with a ray whose direction can be changed by the user. Given the camera ray and the positions of points on the mesh, the system interactively computes the projection of each point onto the camera plane. By defining the projection type and the camera position, we can automatically create orthogonal or perspective images from any new point.

While we tend to think of 3D object as solid, computer animation requires only the surface geometry and commonly represents this information using one of three different surface types: Polygons, NURBS, or Subdivision surfaces (Figure 2.18). For each surface type, we will describe the workflow for manipulating it using a typical UI such as Maya™[Aut].

2.4.1 Polygons

Polygonal models were the first implemented surface type. Computationally, polygons are lightweight and flexible. Meshes are composed of a number of fused polygons, usually triangles or quadrilaterals.

Typically, to model using polygons, the user starts by creating a simple, predefined primitive polygon, such as a cube, sphere, cylinder, or cone. By applying a series of polygonal modeling operations, the primitive can be molded into a com-

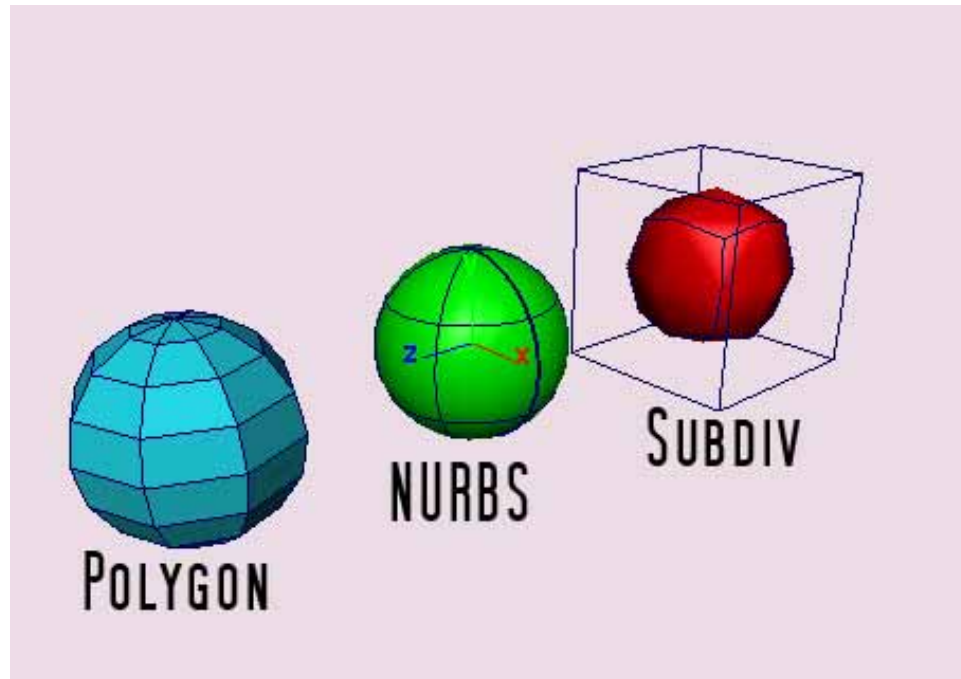


Figure 2.18: The three major surface types in 3D graphics.

plex object. Polygons are modified by editing their components: vertices, edges, and faces. The components can be selected dragged into any configuration, scaled, or rotated. Also, the user may split polygonal faces, create additional vertices to work with, extrude columns from the existing faces, fuse vertices, and delete unnecessary vertices.

Consequently, it is easy to add local details without affecting the rest of the model. As with traditional 3D sculpture, artists are generally advised to work from general to specific; perfecting the global shape before fleshing out details. When the model's complexity increases, the task of moving vertices while maintaining smoothness becomes arduous.

Polygons are ideal for creating structures with hard edges, such as buildings, but not for smooth, organic models. Meshes can, however, be smoothed by tessellating the existing faces and averaging the positions of resulting vertices. Unfortunately,

as the number of polygons increases, models become harder to work with and can create performance issues for the software.

2.4.2 NURBS

The desire for smoother models led to the development of NURBS modeling, which uses curves rather than discrete points as primitives. To create a NURBS surface, the user can start by drawing a number of curves and allowing the system to interpolate or "stretch" surfaces between them. Each point on the surface is interpolated from the original curves.

The user can further edit the surface by creating *isoparametric* curves that follow the existing contours of the surface. Each of the isoparametric curves has a number of control vertices that can be moved to change the shape of the curve locally, which in turn changes the surface. Consequently, when the point is moved, the underlying surface can accommodate smooth transitions.

While this method can produce attractive, organic models, it is not optimal for models that need hard edges. Adding local detail is difficult. The major disadvantage of a NURBS surface is that it is constrained to maintain a rectangular topology. It is impossible to create branching architectures, such as arms or legs, with a single NURBS patch. Multiple patches can be aligned to achieve this; however the artist must take special care to ensure that the seams are properly attached and will not become visible when the character moves. For more information on NURBS modeling, an interested reader may refer to *The NURBS Book*. [PT97].

2.4.3 Subdivision Surfaces

Subdivision surface modeling is an extension of polygonal modeling, where a crude polygonal cage is used to manipulate a procedurally smoothed version of itself that lies inside. It enjoys the advantages of both polygonal and NURBs modeling: Like polygons, subdivs accept local details, and like NURBS, subdiv models are smooth. The workflow is identical to that of polygons. Today, for their intuitiveness and flexibility, subdivision surfaces are preferred by many studios.

2.4.4 Scanned Models

The previous sections have presented ways to build models from scratch. Today, technology has also made it possible to acquire models from the real world. 3D range scanners work by shining a laser on a static object, recording the laser's hit points, and then triangulating the positions of surface points in virtual space. The points themselves can be used as rendering primitives, but more commonly the points are connected and converted to polygons.

Scanned models are extremely useful when a subject must be modeled exactly; applications include digital stand-ins for actors in live-action films and medical simulations. The disadvantage is that the scanners cannot properly capture concavities or complex, overlapping geometries such as the ear. Also, the layout of points may not be suitable for animation, and the models are of extremely high resolution. In many cases, an artist must manually clean up the scanned mesh to use it for animation.

Recently, the field of computer graphics has seen the development of intelligent systems that can fit an animatable model to a scanned mesh, a topic we will discuss further in Chapter 4. The concepts used by these systems have influenced

the implementation of our project. While the scan-fit algorithms fit 3D to 3D, our system uses similar computations to fit parts of a 3D model to user-drawn 2D curves.

2.5 The Animation Pipeline

We have discussed the technical process of creating a 3D model, but where does this fit into the ultimate goal: to create an appealing, dynamic character in an animated film? The production of an animated feature film can be divided into several consecutive, interdependent tasks. In this section, we briefly discuss each step in the standard animation pipeline used by many film studios, and how each stage influences the previous and subsequent stages.

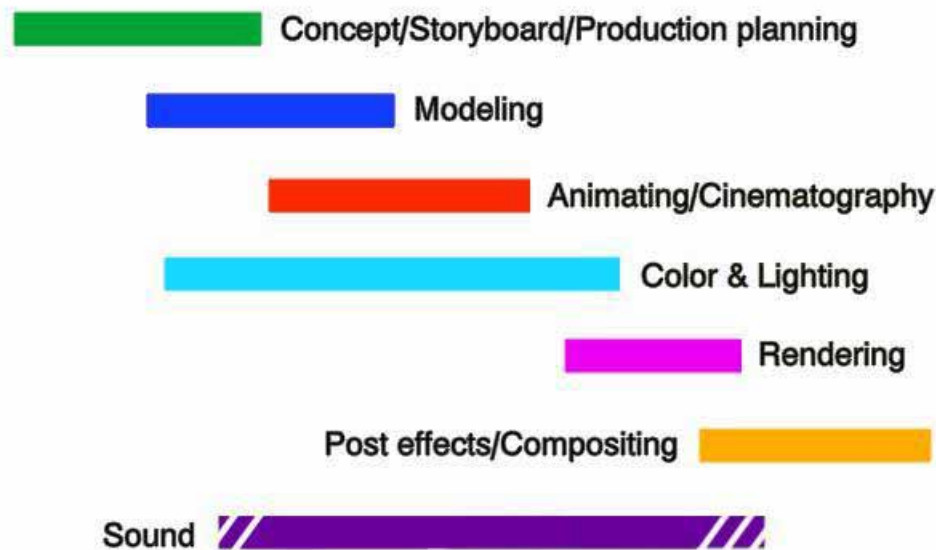


Figure 2.19: The standard Animation Pipeline. Some stages overlap significantly. [PSU]

2.5.1 Conceptualization: Art and Storyboarding

The first stage of the pipeline requires talented artists and great storytellers. The characters, sets, color schemes, and overall look and feel of the film are determined at this stage. Artists typically draw numerous concept sketches of the characters and eventually choose designs which will most vividly convey their personalities. When the final design is selected, a more detailed drawing is made as a blueprint for the modelers, who will translate the sketch into a 3D mesh.

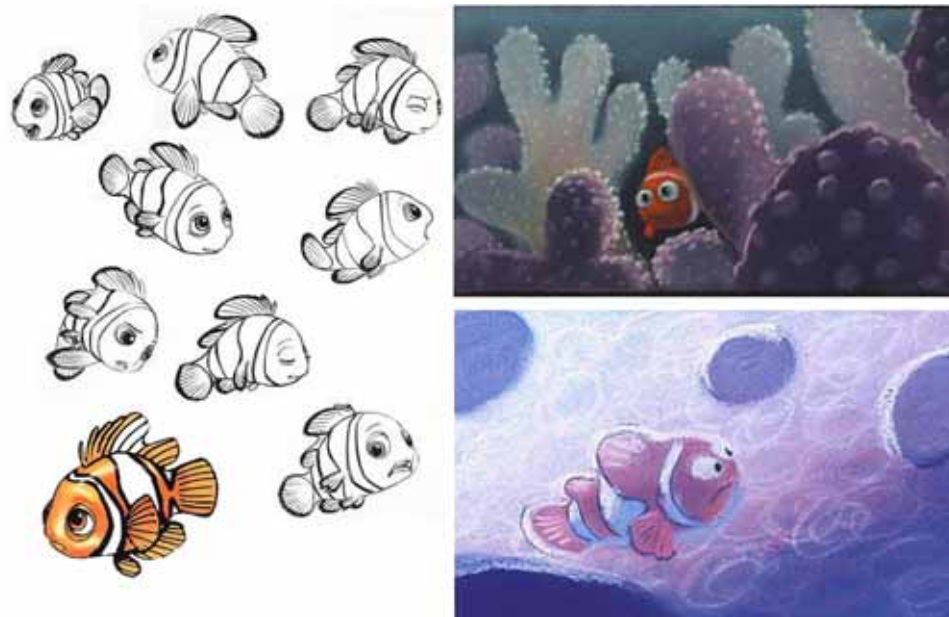


Figure 2.20: Early concept art and color keys for *Finding Nemo* by Pixar artists. [Pix]

The storyboard is essentially a comic book style representation of the story. The individual drawings are used to demonstrate the camera angles for each cut. Often, the storyboard drawings will be put together in an animatic, which is a video slideshow of the scenes. The animatic is used to determine timing, and is often fit to a rough soundtrack with dialogue between characters.

Conceptualization is the decision making stage, and must take into considera-

tion all the following stages. Will the characters look right when modeled in 3D? Will the animators be able to achieve the movement detailed in the storyboard? Will the shading and lighting department be able to achieve the intended look and feel? What special effects are necessary for storytelling, and will the studio's computing power support them? Will the camera angles hold the viewer's interest? Once all the details are finalized, the artwork is sent to the modeling, shading and lighting, and compositing departments.

2.5.2 Modeling

The details of modeling were discussed in the previous section; now we can see how it fits into the animation pipeline. The modeler works from the concept artist's sketches and reproduces them as closely as possible in 3D. It is the modeler's job to make sure the character's geometry will hold up when viewed from different angles.

But it is not enough to simply re-create the shape. The way the vertices and edges are laid out, or the **topology**, is extremely important. More detail is needed in areas that deform, such as the knees and elbows, than static areas, such as the back of the skull. The edges should follow the shape of the required deformation. For example, the edges around the mouth should form radial patterns around the lips that extend to the corners of the nose so that when the character smiles, the nasolabial crease¹ is visible. A mesh may look appealing even if it has bad topology, but the undesirable effects will become visible in the subsequent articulation and animation phases. Modelers must make sure that the model both conveys the character and has a clean, animatable topology.

¹The crease from the nose to the corner of the lips.

2.5.3 Articulation: Rigging and Skinning

Articulation, or rigging, is the process of building controls for a character or object's movement. This stage cannot start until the character model is finished and "frozen." 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. For example, the angle between the thigh bone and calf bone can be limited to prevent interpenetration. Constraints can be applied to any part of the rig as "handles" for the animator to use. The handles are simple objects that are visible to the animators but do not show up in the final rendered scenes. 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; 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.

Articulation can begin once the modelers have produced a rough model of the character and the primary proportions are frozen. However, since skinning depends on the geometry of the model, it cannot begin until the model is completed.

2.5.4 Animation

The animator is responsible for bringing the character to life. Many animators have an acting or directing background. Using the controls developed by the articulators, they refer to the storyboard and re-create the poses of the character at different points in time, along with camera angles. These poses are called **keyframes**. The animator determines how far apart keyframes must be, in space and time, to achieve realistically timed motion.

Unlike 2D animators, 3D animators do not have to pose the model for every single frame. After the animator sets extreme poses, the in-between poses are interpolated by the software.

Animation can start once articulation is finished. The models need not be shaded; however, the geometries cannot be changed at this point. The concurrent work in shading and lighting depends on the topologies of models, thus it is imperative to maintain consistency across departments.

2.5.5 Shading and Lighting

The shading and lighting department's job is to finalize the look and feel of a film. Each model is "painted" with a texture or a shader, which describes the way light reacts with the surface. The easiest way to texture a model is to apply a material uniformly. However, characters frequently require complex textures that vary in specific ways over the surface. This is achieved by assigning color and reflectance properties to each vertex coordinate in the model. To do this, the model must be virtually unwrapped and flattened into a parametric UV map (the U and V denote the two axes of the texture coordinate space). The flat texture image can be imported into photo editing software to paint variations over the

surface. Several texture maps can be superimposed on one model; attributes such as shininess and transparency can also be painted in the same manner.

While some basic uniform textures can be made before the models are finished, the model must be finalized before any texture mapping.

Because the appearance of a texture depends on the illumination sources within the scene, the shading and lighting phases often occur concurrently. The lighting department is responsible for creating the mood of a particular scene or set. Light is essential for bringing out the 3D structure of a CG model, drawing attention to particular objects in a scene, and guiding the viewer's eye across a digital terrain. When CG is merged with live action, realism is achieved by matching the CG lighting to the live scenes. Lighting usually waits until the animation is finished, since the placement of lights depends on the positions of characters.

2.5.6 Special Effects

The special effects department works in conjunction with shading and lighting to animate natural phenomena such as wind, fire, and water. In some studios, hair and cloth are a part of the special effects department, but sometimes special departments are created for these difficult tasks. Special effects require a conglomeraton of modeling, animation, and lighting; mixed expertise and creative problem-solving are necessary to produce breathtaking effects.

2.5.7 Rendering

Once all the animation, shading, lighting and special effects tasks are completed, scenes are **rendered**. Rendering computes all the necessary lighting calculations for and outputs still frames which are sewn together to produce the final animation.

Typically, the renderer consists of a network of computer systems that divide up the task of rendering an entire scene; each system usually outputs one frame at a time. Today, the rendering stage is considered the largest bottleneck in the pipeline; studios must plan to finish the other stages by a set deadline in order to allow sufficient rendering time before the film's release date.

2.5.8 Compositing and Post-Production

Often, rendering is done in pieces to improve efficiency. For example, in a scene where characters are moving but the background or portions of the background do not change for long periods of time, it is not necessary to re-render the entire scene for every frame. As an alternative, the background can be rendered by itself without any of the characters, and then the characters can be rendered in separate frames against a plain background for the entire duration. **Compositors** superimpose the characters onto the background and ensure that there are no visual incongruities.

Special effects and computationally expensive shaders can also be rendered on a separate layers which the compositors compile. In films that merge animation with live action, compositors may manually alter the color of a rendered image so that animated objects fit into the illumination of the live scene. Once all the compositing and post-production (which includes adding a soundtrack) is finished, the film is released.

2.6 Summary

Technology has profoundly changed the art of animation. It has invented new outlets for creativity that give an impressive degree of control over complex structures.

Imagine trying to produce a photo-realistic animation in 2D! With 3D software, the dream of photorealism in animation is now being realized with startling imagery.

However, computers also place a virtual barrier between the vision and the tools used to realize it. In terms of Betty Edwards' theory, manipulating discrete points on a 3D mesh forces the artist to think about an artistic task in a linear, mathematical, left-brained manner, when in fact the loose, free, spatially expressive thought process of R-mode thinking is required. No matter how many sophisticated tools software can provide, subconsciously, many artists will not be satisfied unless they can work with their hands.

Meanwhile, many artists, even experienced artists, struggle to reproduce correct proportions in their artwork. Many focus on the most perceptually interesting parts of a face and neglect the other parts. If technology can alleviate the dilemma, why not allow it? In essence, artists and novices want two things: to have direct, free, hands-on control over what they are making, and to experience the delight of a final product that looks right. Our interface aims to fulfill both requirements.

Chapter 3

Goals

3.1 Stating the Problem

The advantages and disadvantages of CG modeling versus traditional art have inspired the development of our sketch-based parametric modeling user interface, which provides both artistic freedom and precise control over a 3D form.

Our goal is to help artists represent human characters realistically. While we have chosen 3D modeling software as our medium, to clarify the goals of our research we will ignore the medium for a moment and state the problem of realistic character creation in abstract terms. Whether we use pencil and paper, marble and a chisel, clay and bare hands, or a 3D modeler and renderer to create an artistic depiction, we can break the task down into several basic stages:

- **Creating** the shape
- Expressing the **volume**.
- Defining a **pose**.
- Adding **texture** and fine details.

- ”Erasing”, modifying and refining.




Now what happens when we choose the medium? Depending on what tools we use, the stages may be done in a different order or can be intertwined with each other. For example, in sculpture defining the shape is analogous to expressing the volume, whereas in drawing they are two distinct stages: drawing contours to define the shape and shading to bring out the volume. Table 3.1 takes a closer look at how three different media approach the challenge.

We can see that the level of difficulty of each stage varies based on the media and the artist’s preference. In drawing, conveying volume is a complicated task that requires a good eye; it is not easy to visualize how light and shadow falls upon a complex 3D shape. However in sculpture the task is trivial; the volume is conveyed as the shape is created. In 3D modeling, a skilled artist must place lights appropriately to emphasize volume, but shadows are calculated automatically.

Our goal is to simulate the freedom and inherent intuitiveness of hands-on artistry in a 3D environment. While 3D software is difficult to get used to at the basic level, the technology is powerful enough to justify initial roadblocks. When drawing, an artist cannot zoom in on a part of the drawing, cannot move around it in 3D, and cannot copy and paste parts of the drawing. In animation, the time it takes to get past the unintuitiveness of the media pays off richly. Once a character is modeled, rigged, and skinned, it does not have to be re-created for every frame. It would take an impractical amount of time to achieve photorealism in 2D animation, but the reusability of a 3D model leaves time to refine details and create richly realistic textures.

Combining the advantages of 3D software and traditional drawing while minimizing their individual disadvantages, we have developed a unique method that

Table 3.1: Images: 2D walk cycle from [Idl]. Claymation at PWC: from [PWC]. Students at the animation lab at CalArts [Cal]

	DRAWING	SCULPTURE (CLAY)	3D SOFTWARE
STEP 1	Pose: Gesture lines	Volume/Shape: Add and remove clay to create the shapes. Work from general to specific.	Shape: 1) Create a primitive shape 2) Add geometry (extrude and subdivide) 3) Move vertices to shape * Automatically smooth
STEP 2	Shape: 1) Draw basic guide shapes/volumes 2) Draw contour lines		Erase: Fix any topological problems to get a mesh suitable for animation
STEP 3	Erase: remove the basic shapes and guidelines	Erase: Remove parts, smooth with fingers	Texture: Assign a texture map
STEP 4	Volume: Add shading and highlights	Pose: Bend the model into the appropriate position	Pose: 1) Build rig and controls 2) Attach the mesh to rig 3) Position the rig
STEP 5	Texture: Sketch details	Texture: Use tools to add surface detail	Volume: 1) Illuminate mesh with digital lights 2) Render
REFINE	Erase, draw, erase, draw	Add, smooth, remove, smooth	Extrude, tweak, delete, tweak, add points, tweak
EASE OF ANIMATION	Need to re-draw the entire character for every frame. Need to preserve volume.	Character doesn't have to be re-created for every frame, but does have to be repositioned for every frame.	Character reusable for every frame. In-between frames can be interpolated.
			

utilizes anatomical information and implements familiar drawing conventions. We aim to harness the power of 3D technology with the intuitiveness of a pencil.

3.2 Characteristics of an Ideal CG Interface

- **Completeness**

The steps of character creation, modeling, texturing, rigging, and skinning currently exist as separate parts of the animation pipeline. When changes are made at one point in the pipeline, the other parts must be changed accordingly. Traditionally, the necessary adjustments at each stage would be made by hand, making for a tedious iterative process. For example, a minor change in the topology of a mesh will "break" the texture mapping and skinning routines. An ideal system would seamlessly and intuitively carry changes from the modeling stage through the rest of the pipeline.

- **Intuitiveness**

One can provide the user with an exhaustive list of options and sophisticated tools, but if the user cannot find them easily or understand how to use them, they are not useful. Therefore, it is important to define, display, and organize controls in a way that the user can access them easily. Ideally, a successful system should "notice" if the user is about to perform an action that may damage the model. However, too many constraints may be limiting for a clever user, thus the constraints must be chosen wisely. In either case, a user should be able to trust the interface and easily find the tools they need.

- **Flexibility**

No matter how exhaustively one parameterizes a model, there will always be features that an imaginative user cannot create. What if the user wants to add antlers to the characters head, a non-human skin texture, a Pinocchio nose (with the rest of the proportions intact), or a special rig feature? Many programs keep their inner workings hidden from the user, which makes the model "untouchable" in the sense that control points of the mesh cannot be moved manually, and the model is only partially modifiable via sliders. This prevents the user from damaging the model, but can be frustrating for an experienced user. A user should be able to exploit an open architecture without undesirable side effects. Finding a balance between user-friendliness and flexibility is essential to an ideal UI.

- **Interactivity**

With potentially large sets of interdependent measurements, some of which are nonlinear, one faces the problem of making numerous calculations based on each user input. Interactivity needs to be maintained, and thus algorithms must allow users to view results of their work at reasonable feedback rates.

- **Getting the Best of Both Worlds**

An experienced 2D artist knows how to use a pencil and can appreciate the infinite flexibility and intuitiveness. A 3D artist can appreciate being able to reuse work and knowing that he or she does not have to worry about making it appear three-dimensional. An ideal system should maximize the advantages of each method and minimize their disadvantages.

Chapter 4

Previous Work

Keeping in mind our goal of developing a user-friendly character modeling technique, this chapter examines prior works which provide the foundation for our interface. Our ideas conglomerate concepts from several separate areas of research in computer graphics.

First, we must construct a detailed, animatable **prototype** character model. Digitally sculpting a human character requires special design considerations. Thus, Section 4.1 focuses on previous concepts that guide the process of mesh creation and rigging for successful animation.

The second step is to constrain the prototype to maintain anatomical proportions. Section 4.2 explores previous literature about **parametric modeling**, with special attention to those methods that employ anthropometric parameters. A subset of research in this area deals with fitting animatable prototypes to 3D scans, and introduces some of the algorithmic aspects of our work.

The third and final step is to develop an intuitive **sketch-based user interface** for editing the prototype. A relatively new, exciting area of research in computer graphics aims to develop interfaces where a user can physically "draw" 3D ob-

jects. Graphics tablets are used to maximize the advantages of such interfaces. In Section 4.3 we will document work on sketch-based interfaces for modeling.

4.1 The Anatomy of a Prototype Digital Human

4.1.1 Deformable Face Topology

The pioneering work in facial modeling was done by Frederic Parke [Par72]. Creating a complex shape that can deform properly when animated requires a close study of how the original object moves. Parke studied faces and facial expressions of volunteers to get a sense of how the face deforms. Then he experimented by iteratively painting a mesh structure on the faces of his volunteers to determine the optimal layout of vertices, as shown in Figure 4.1. Once he was satisfied, he used a photogrammetric method to recreate the painted mesh in 3D (Figure 4.2).

Not surprisingly, Parke discovered that the optimal configuration of vertices and edges is one that closely follows the lines of underlying facial muscles. This basic heuristic is still used today by virtually all 3D modeling artists to produce realistically animatable faces.

4.1.2 Applications of Animatable Prototypes

The idea that one generic, reusable mesh topology could be used for all faces spawned many exciting applications. This is particularly important since all stages of the animation pipeline, from texturing to rigging and skinning, are dependent on the topology of the model. In practice, these tasks are deferred until the modeling phase is finished. However, if it is known that the topology of the model will not change, the consecutive tasks can be completed simultaneously or even automated.

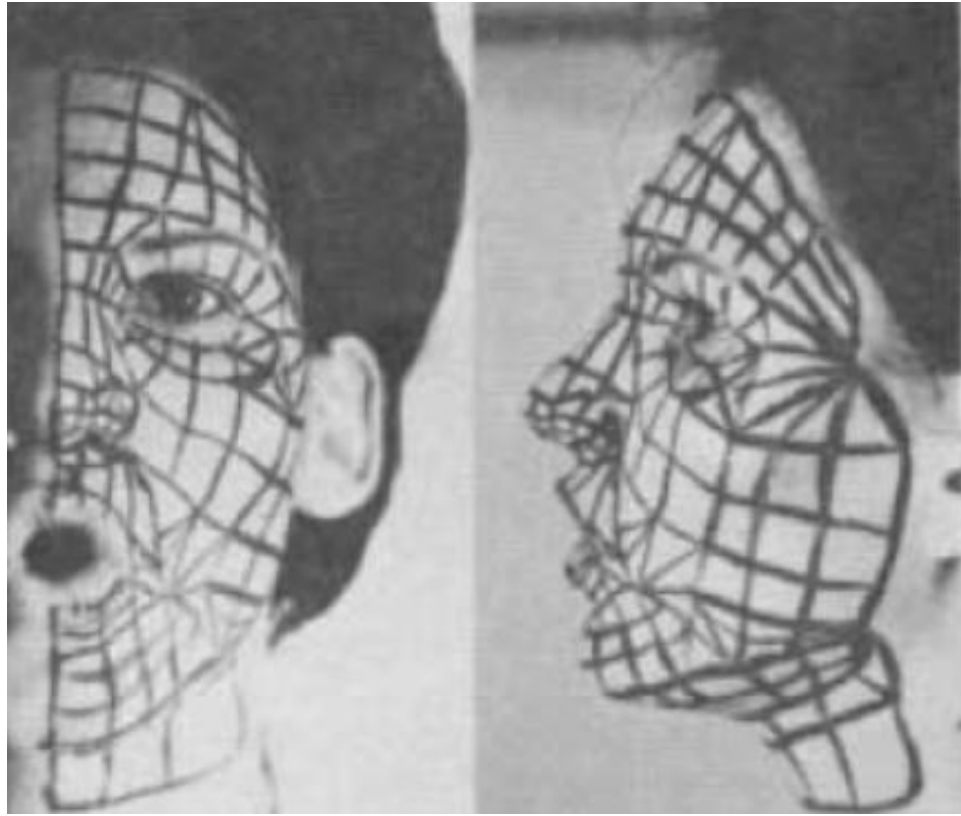


Figure 4.1: A mesh painted on to a volunteer's face. In this case, the "volunteer" happens to be his wife. [Par72]



Figure 4.2: Parke's 3D facial model with optimal topology. [Par72]

This implies that studios could, in theory, build only one character rig, define only one set of skin weights, and unwrap only one UV texture map to fit a large number of characters.

The fact that characters have the same topology does not mean that they must look the same. A user can move the points of a model, without adding or deleting points, to change the shape while maintaining constant topology. While this is clearly faster than creating a new mesh from scratch, researchers continue to search for faster ways to generate a variety of characters. A major area of research focuses on the usage of highly accurate 3D scans of actual faces and bodies, described briefly in Section 2.4.4, for animation.

4.1.3 Fitting a Prototype to a Scan

As mentioned in Section 2.4.4, 3D scans are useful for capturing nuances in complex 3D objects, but have three primary disadvantages:

1. Scanners fail at capturing overlapping geometry, such as the ears and eyelids.
2. The models are of extremely high resolution, making them difficult to modify.
3. The topology is not suitable for animation.

There are several ways to circumvent these obstacles. One possible approach is to manipulate the scanned mesh directly. The mesh could be procedurally decimated to reduce the number of points, and then manually edited to both correct bad topology and reconstruct poorly scanned areas. However, editing the mesh by hand is a laborious process. What if one could develop an algorithm to automatically delete unnecessary vertices and reconfigure them into an animatable

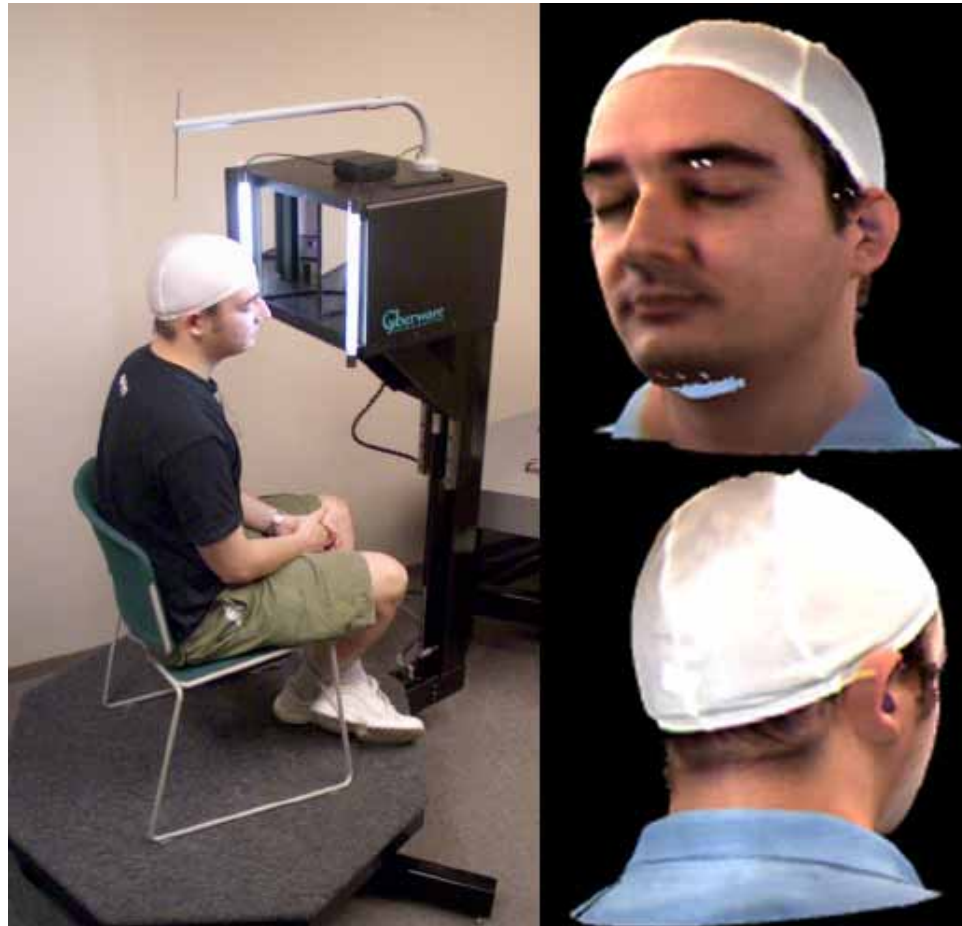


Figure 4.3: The Cyberware 3030/RGB scanhead used at the Program of Computer Graphics at Cornell University. On the right is a typical output. Note the artifacts under the chin and on the eyebrow, and the unnatural blockiness of the ear in the back view. Subjects are often required to wear a swim cap because the reflectance of hair tends to be misinterpreted by the laser beam, producing spikes in the geometry.[Bib04]

topology? It is an intriguing thought, but such an approach would require a heuristic to determine which vertices must be kept. This approach may require solutions to some difficult computer vision problems, not to mention the laborious task of iterating through a highly dense mesh to determine whether each vertex should or should not be deleted. Lastly, one would still need to find a way to deal with poorly scanned areas.

A simpler approach, instead of attempting to change the scan itself, is to use it only as a reference and adapt a preexisting prototype model to fit its contours. This approach tackles all three of the aforementioned challenges:

1. For hard-to-scan parts of the mesh, the scan can be ignored altogether and substituted with the prototype mesh's artist-modeled eyelids and ears.
2. The prototype has a fixed number of points.
3. The topology of the prototype is clean and optimized for animation.

The method was first introduced in 1995 by Lee, Terzopoulos, and Waters at the University of Toronto [LTW95]. They simplified the problem by working with the 2D representations of the scans, and utilizing the depth map generated by the scanner. By fitting a 2D cylindrically defined prototype mesh to this flattened scan, they could then interpolate the 3D structure. They also observed that since a generic animatable topology already follows the muscular structure of the face, the natural next step is to consider developing a generic rig for the face.

4.1.4 Rigging prototypes

In 2004, Jacobo Bibliowicz presented an automated facial rigging and animation system for scans that was based on extensive studies of facial muscle movement,

thus developing a rigging prototype for human faces [Bib04]. The prototype rig used a unique configuration of "bones"¹ and special deformers instead of skinning in order to simulate muscles (Figure 4.4). Bibliowicz isolated the major muscles used for facial expression. Knowing which muscles were used for each expression allowed for the creation of intuitive animation controls, but the user was also given the option of animating individual muscles. Each "muscle" affected a predefined set of points on the face of a prototype model that was affectionately called Murphy.

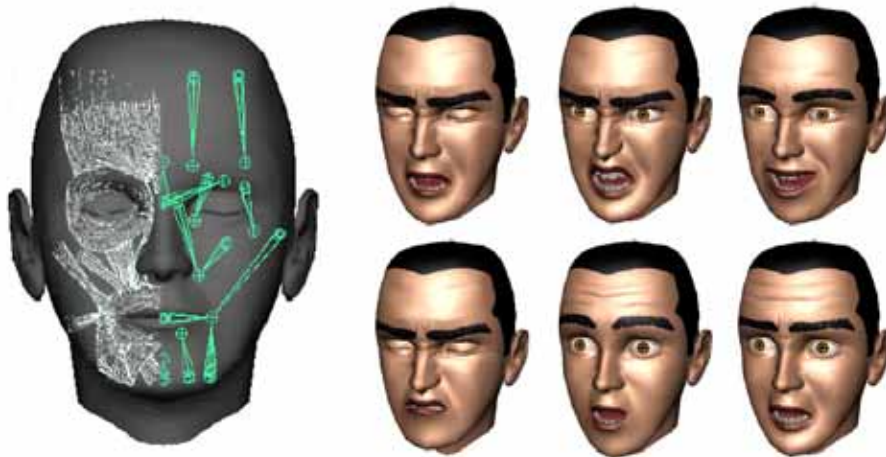


Figure 4.4: The left side of the face is superimposed with a diagram of actual facial muscles. The right side shows the generic facial rig developed by Bibliowicz. Using these muscles, the prototype could realistically convey the six basic emotions: (clockwise from top) Sadness, anger, joy, disgust, surprise, and fear.[Bib04]

The automated rigging process was twofold: First, Murphy was fit to a scan by minimizing the silhouette difference between himself and the scan. The fit was improved by locally aligning distinctive features. Next, the rig was mapped to Murphy's newly positioned vertices. Once the mapping was finished, the sys-

¹The "bones" described in this section refer to bone objects in Maya [Aut], which are not rigid like real, anatomical bones. The fact that Maya bones can be scaled makes them usable for muscle simulations and a variety of effects.

tem provided a user interface for animating the face. Animatable facial models produced by Bibliowicz’s system are shown in Figure 4.5.



Figure 4.5: Four fitted scans posed in the same facial expression using the prototype rig. The top left face is the prototype, Murphy. The top middle is the same face as the top right, with Murphy’s generic texture in place of the scanned texture.[Bib04]

Designing an efficient, robust algorithm for fitting a prototype mesh to a scan is still a difficult problem that needs further research. Nevertheless, algorithms work successfully for a large number of test cases, making them extremely useful for practical application.

4.2 Maintaining Human Proportions in 3D

In Section 4.1, we saw how human facial meshes and rigs can be generalized. This reiterates an important fact we discussed in Section 2.1, which is that all human

faces are remarkably similar. This means that the facial measurements of any individual vary within a limited range of values. With access to large anthropometric databases, in addition to generally accepted artistic proportions, researchers have developed methods to edit character models while faithfully obeying predefined limits.

What makes these systems useful? We tend to describe faces in terms of their parametric measurements, pointing out the shapes, sizes, and placement of facial features. For example, we may say that a person has a "A hooked nose, a large forehead, and wide-set eyes." Thus, a parametric modeling interface equipped with sliders for changing the "hookedness" of the nose or the spacing between the eyes is intuitive because it allows a user to change recognizable, familiar features of a face rather than nameless control points on a mesh.

4.2.1 Parameterized Facial Models

In 1982, Parke created the first parameterized face modeling system [Par82] using his optimal face topology (Figure 4.6). Based on general facial proportions, the intuitive implementation gave users control over the size and shape of individual features. The features could be changed by scaling, rotating, interpolating, and repositioning relevant control points in ways that would maintain a believable facial shape.

Parke organized the system by identifying two general classifications of facial parameters:

- *Conformation parameters*, or aspects of a face that make an individual unique. Bone structure defines most conformation parameters. The colors of the hair, skin, and eyes are also considered conformation parameters.

- ***Expression parameters***, which describe the emotion conveyed by the face. There are six universal facial expressions: Joy, anger, sadness, surprise, fear, and disgust [EF82]. Intermediate expressions are less intense versions or composites of these six. Each expression can be decomposed into movements of specific facial features, particularly the eyebrows and mouth.

Parke's system provided users with five types of interactive operations to modify the face prototype:

1. **Procedural** - Creating a model based on user-defined parameters. This method was used for creating eyeballs: the user would specify a radius and color to generate a customized eyeball geometry.
2. **Interpolation** - Changing the shape by interpolating between extreme values.
3. **Rotation** - Rotating points about a pivot point. To avoid boundary artifacts, the strength of the rotation was dependent on the distance from the invisible "bone" being rotated.
4. **Scaling** - A localized size change of a specific facial feature. Like rotation, the scaling operation's effect diminished with distance to ensure that the scaled feature would smoothly blend into the rest of the model.
5. **Position Offset** - Changing the location of a feature in space.

While Parke's system was easy to use and could generate an impressive variety of facial shapes (Figure 4.6), it was limited by the fact that the optimal set of parameters to produce the full range of possible facial shapes is still unknown. Parke used fifteen expression parameters that encompassed virtually all possible facial

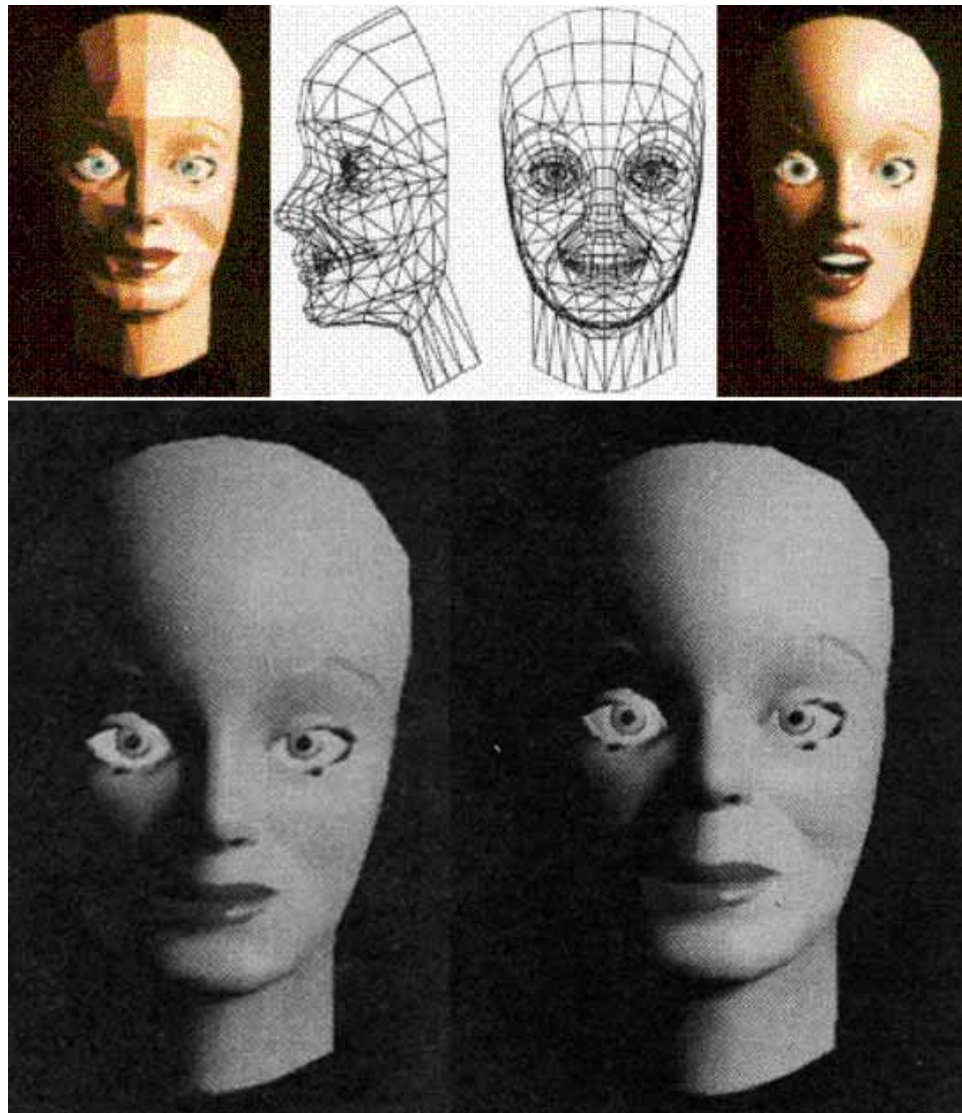


Figure 4.6: The top middle image shows the prototype model used for Parke's system. The top left and right show the same face with different expressions. The bottom two images illustrate changes to the structure, or conformation parameters, of the face: the left face has a high forehead and small mouth, the right face has a shortened nose. Modified from [Par82]

expressions, but defining a full list of conformation parameters is more difficult. To generate this list would require extensive anthropometric research; in theory one would have to recreate a large data base of faces using a given parameter set. Parke's model was not anatomically based and therefore allowed artists to create faces with unrealistic proportions.

4.2.2 Anthropometric Facial Modeling

In 1998, DeCarlo and Metaxas created the first system that used anthropometric parameters to generate faces [DMS98]. Their work is referenced heavily in many later papers on the topic, making it a monumental contribution to the field. Their goal was to quickly generate a large number of realistically proportioned facial models. The system created unique facial models with constraints based on the anatomical statistics gathered by Farkas [Far94].

DeCarlo's system started with a B-Spline prototype mesh modeled by an artist (Figure 4.7). Anthropometric landmarks were manually specified points on the model.

To create a unique face, the algorithm used graph theory to formulate cohesive sets of measurements. It started by randomly generating a set of base anthropometric measurements, using the means and standard deviations from Farkas's database as guidelines. Then, using the proportional relationships, the system generated values for dependent measurements. By representing the measurements as a graph, with the measurements as nodes and proportions as edge weights, they applied a minimum spanning tree algorithm to determine a set of measurements that would stay within the allowable range of facial variation.

Both linear and nonlinear constraints were observed. The linear constraints

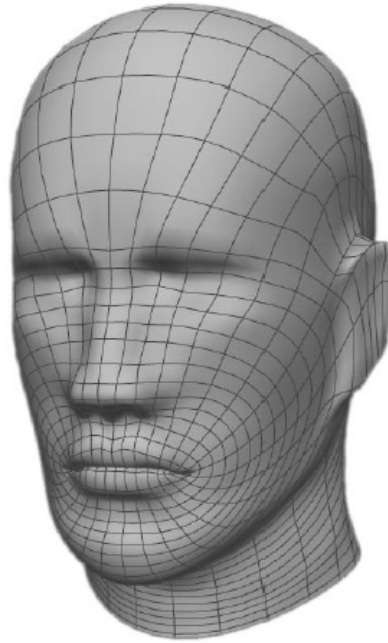


Figure 4.7: The system used a manually modeled B-spline mesh as a prototype. Landmarks were represented as handpicked control points on the mesh.[DMS98]

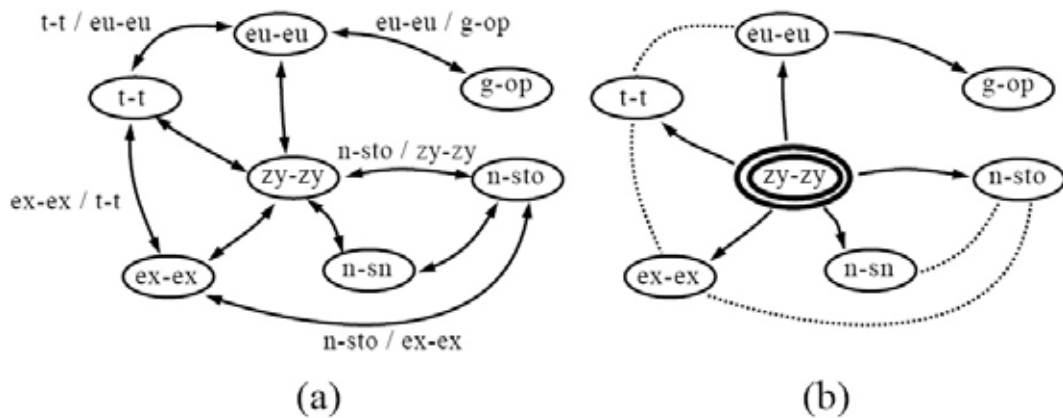


Figure 4.8: Representing Farkas's measurements as a graph for De-Carlo's algorithm. Each node represents a measurement between two landmarks, and each edge weight is the ratio of the two landmarks it connects. In (b), one measurement was designated as an independent measurement, and the graph was reconstructed as a directed graph. This way, a minimum spanning tree algorithm could calculate all the independent measurements.[DMS98]

(axial distances) were maintained using a matrix that kept track of the transformation from the prototype. For each nonlinear constraint (shortest distance, tangential distance, angle between landmarks, and angle of inclination) a different positive function was associated to measure how far the measurement deviated from the prototype. A penalty was associated with each deviation.

After repositioning the landmarks based on the new measurements, the system minimized an objective function to construct a fair surface that fit through them. To generate a plausible solution, the surface-fitting heuristic aimed to keep the geometry as smooth as possible, and as close as possible to the prototype model.

The system was not interactive and required a minute to generate a new random face. Because the system relied on a very sparse set of landmarks, the faces were not realistic, but the varied results proved the integrity of a powerful heuristic for crowd generation. More landmarks would be necessary to generate realistic models.

The smoothness constraint became a limiting factor of the system. Since the curvature of the face was determined mathematically by minimizing surface area, it did not account for nuances of how the skin fits through the landmarks, such as dimples or double chins. These factors make a huge difference in how the face appears. Furthermore, while Farkas's inventory provides a comprehensive list of measurements between landmarks, it does not specify the way that the geometry is curved between the landmarks. Figure 4.10 shows how some of Farkas's measurements can be deceiving. To successfully model a face, parametric systems must acknowledge that anthropometric measurements do not fully specify curvature details.

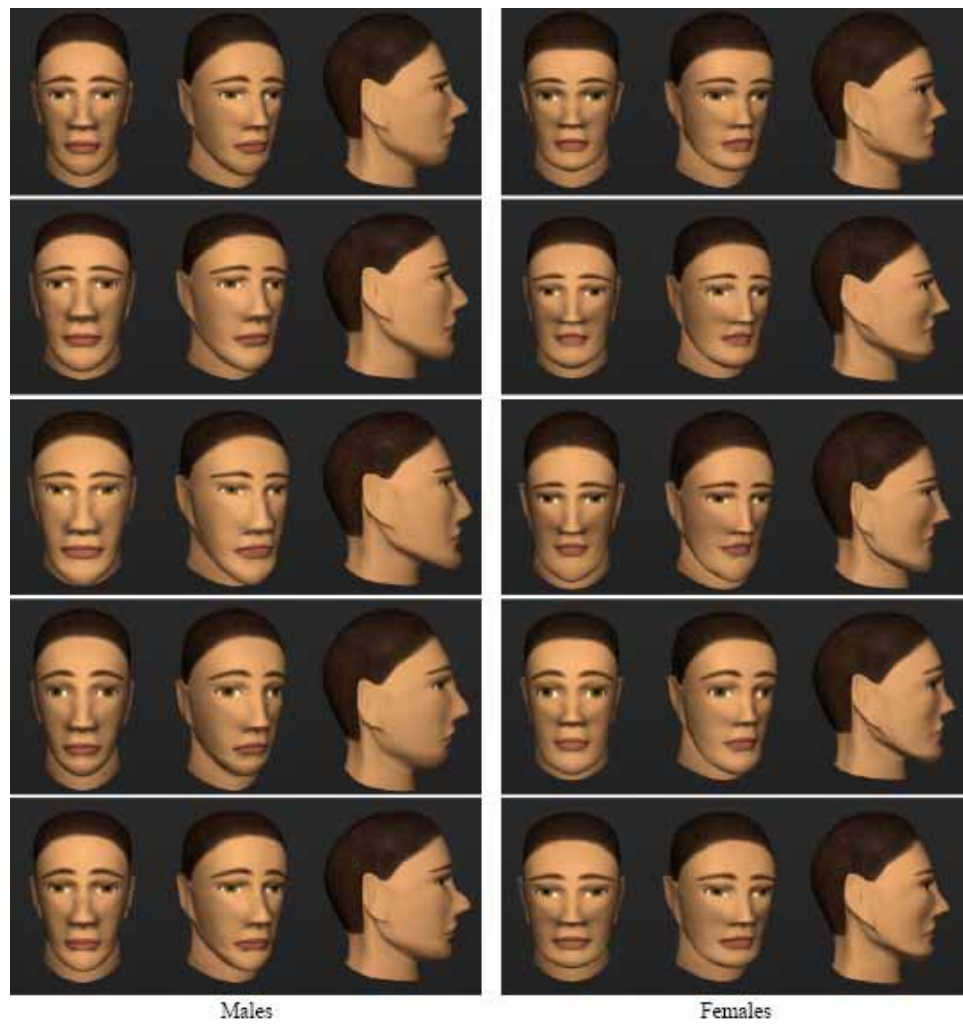


Figure 4.9: Faces generated by DeCarlo's system. [DMS98]

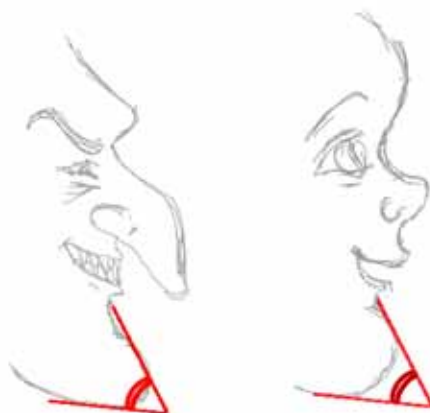


Figure 4.10: Two people with the same chin angle measurement can look very different, considering the curvature of the chin.

4.2.3 Morphable Parameterized Scans

While DeCarlo’s model proved that a wide variety of faces could be generated using anthropometric landmarks, the face model itself was not realistic. Since prototype models can be mapped to realistic 3D scans, why not parameterize the fitted models? In 1999, Blanz and Vetter used the integral concepts developed by DeCarlo to create a morphable model of 3D facial scans [BV99].

Their approach was to take a database of scans that represented a large demographic sample and use simple linear morphing between the models to synthesize new faces and expressions. After using an optic flow algorithm to map scans to a prototype, the system could assume that every model in the database had the same topology, and then easily map points from one model to another.

With access to data describing the average face, Blanz and Vetter were able to isolate specific parameters using **principal component analysis** (PCA). The parameters included both geometric and textural components. Some of the geometric parameters depended solely on size measurements, whereas others had no numerical quantification, such as femininity and masculinity. Also, caricatures

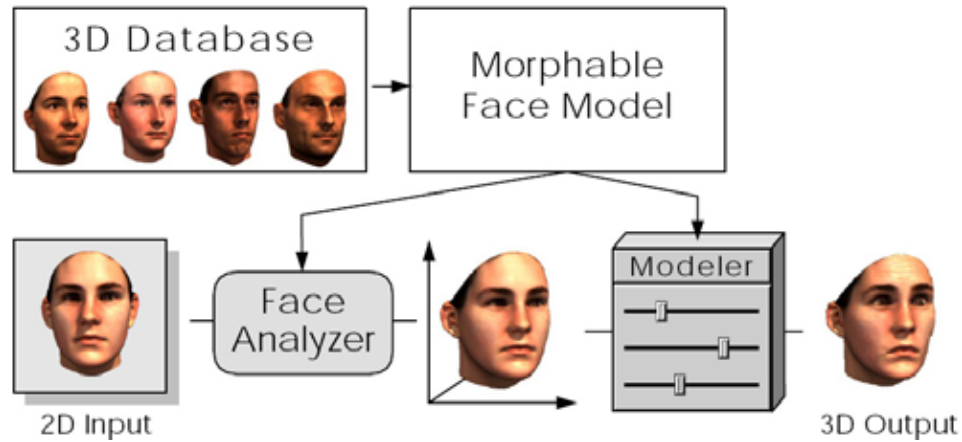


Figure 4.11: A pipeline diagram of Blanz and Vetter’s algorithm. [BV99]

could be created just as they are in 2D: by exaggerating the deviation from the prototype, specifically by multiplying the difference vector by a constant factor.

Another exciting feature of the system was the ability to fit the prototype to a 2D photograph. To do so, the user had to position the prototype model interactively to fit the face in the picture as closely as possible. Then the user would position the lights to approximate the rendering parameters. The system found the best fit to the image by iteratively rendering the prototype face while manipulating parameters, at each stage calculating the difference from the pixels of the original image. When the difference was minimized, the system extracted texture information from the original image and mapped it to the prototype. The uncanny results are shown in Figure 4.13.

Blanz and Vetter demonstrated that the realism achieved by working from existing models did not have to be limiting. Even if the database of faces was limited, the extraction of controllable parameters opened the possibility of synthesizing faces that would not resemble the ones in the database.

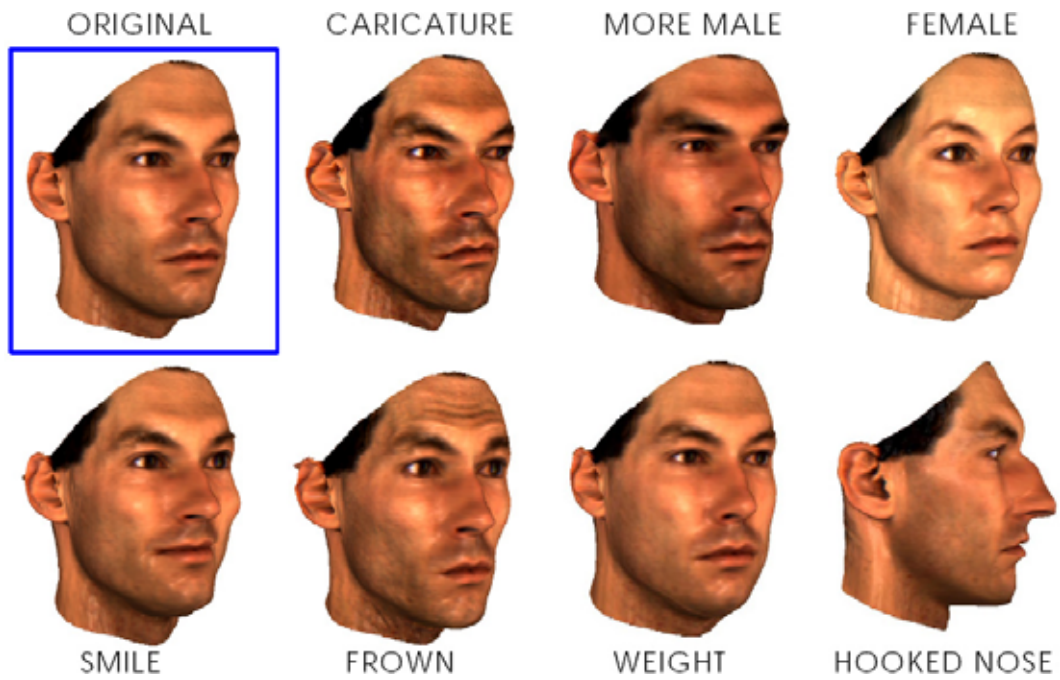


Figure 4.12: Faces generated in Blanz and Vetter's system by manipulating recognizable parameters. [BV99]

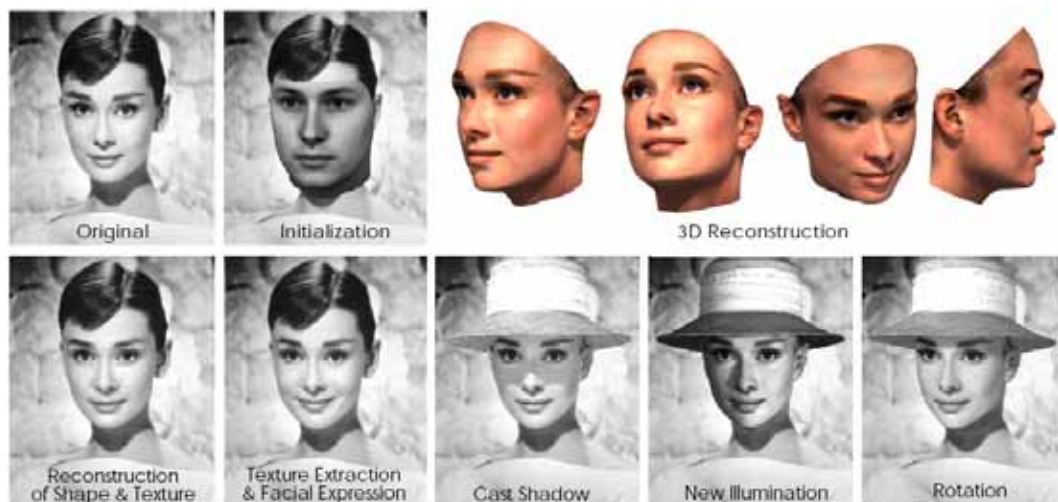


Figure 4.13: Matching a 3D model to a photograph. [BV99]

4.2.4 Parameterizing the Body

Until now, we have only talked about facial models, since as discussed in Section 2.1, the face is the most important part of the body to model realistically. While medical applications that use 3D anatomical models demand "to-the-pore" accuracy, in most artistic applications it does not matter if a leg is not modeled at the same level of detail as the face, as most viewers will not notice. However, the body should make sense holistically. Some parts of the body deserve more attention than others.

One problem in parameterizing the body is that there is no available large source of anthropometric data for the body. Facial anthropometry has received considerable attention, as it is applied in detective work, plastic surgery, and other fields where accuracy is essential. In contrast, body anthropometry is primarily used in ergonomics, which uses large-scale parameters, such as the lengths of the two leg bones, rather than small details such as the bumps on the knee.

With a comprehensive source of anthropometric data for the body, one could create an interface for full-body modeling that is similar to the parameterized facial model. In 2003, Allen and Curless designed a scan-fitting algorithm to generate animatable full-body models and form a parameteric representation based on the data [ACP03]. Their approach not only fit an animatable prototype model to the scans, but also anticipated and handled poorly scanned areas. They used scans from the CAESAR dataset [CAE].

Given a scanned surface \mathbf{D} , and a prototype surface \mathbf{T} , their algorithm determined the set of transforms that had to be applied to \mathbf{T} to fit it optimally to \mathbf{D} . The fit was determined by error minimization based on three criteria.

The first criterion was the data term, which was the sum of the distances

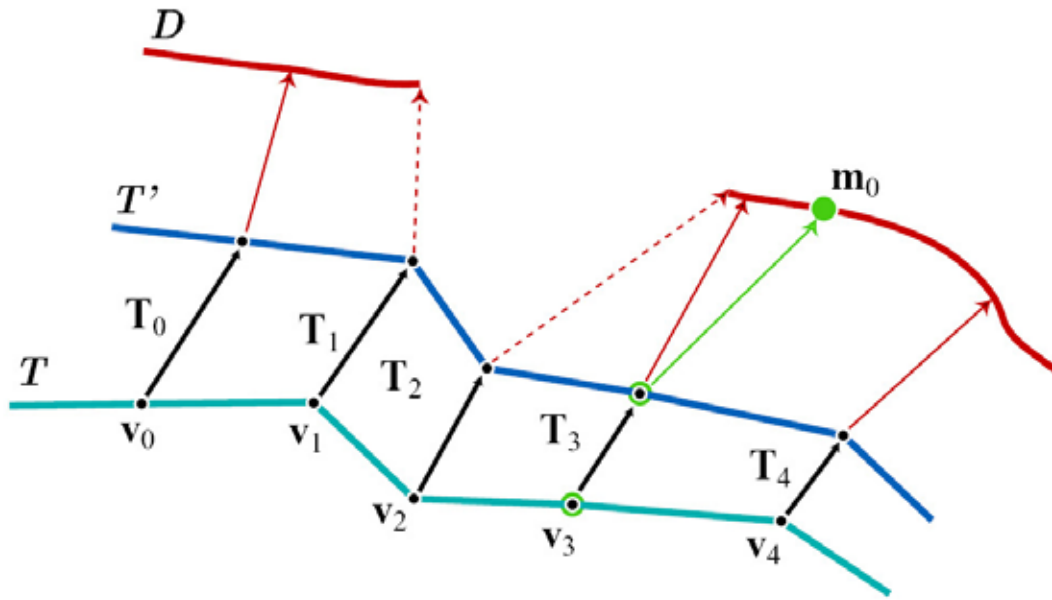


Figure 4.14: Allen’s surface fitting algorithm. [ACP03]

between points on T' (the transformed prototype surface) and the corresponding points on D . Simply put, the data term assumed that the closest point on the scan to a point on the prototype was the best match. The second criterion was the smoothness term, which made sure that neighboring vertices had similar transforms applied to them. This prevented the nose on the prototype from being mapped onto the cheek of the scan. Such a mapping would require a non-optimal amount of calculation. The third criterion was the marker term, used to make sure that anthropometric markers on the prototype were mapped to the corresponding landmarks on the scans. The CAESAR dataset was already landmarked for this purpose. The total error is the weighted sum of the marker, data, and smoothness error. The weights of each error term were input by the user.

Allen and Curless took into consideration that the smoothness error was the most expensive calculation, as it required information from neighboring vertices to calculate the error for each vertex. For the worst-case scenario, when the scan was

very different from the prototype, the calculation became a performance issue. Using a multiresolution prototype model solved the problem. First, a low-resolution version of the prototype was fit to the scan, meaning fewer vertices were considered when calculating the smoothness error. Once a decent fit was achieved, the high-resolution version (parametrically modified to look like the low-resolution version) was fit.

To accommodate parts of the mesh that were scanned poorly, the weights in the data term for those vertices were set to zero so that the prototype model would dominate. Thus, the system was able to maintain detail in areas that are notorious for scanning poorly.

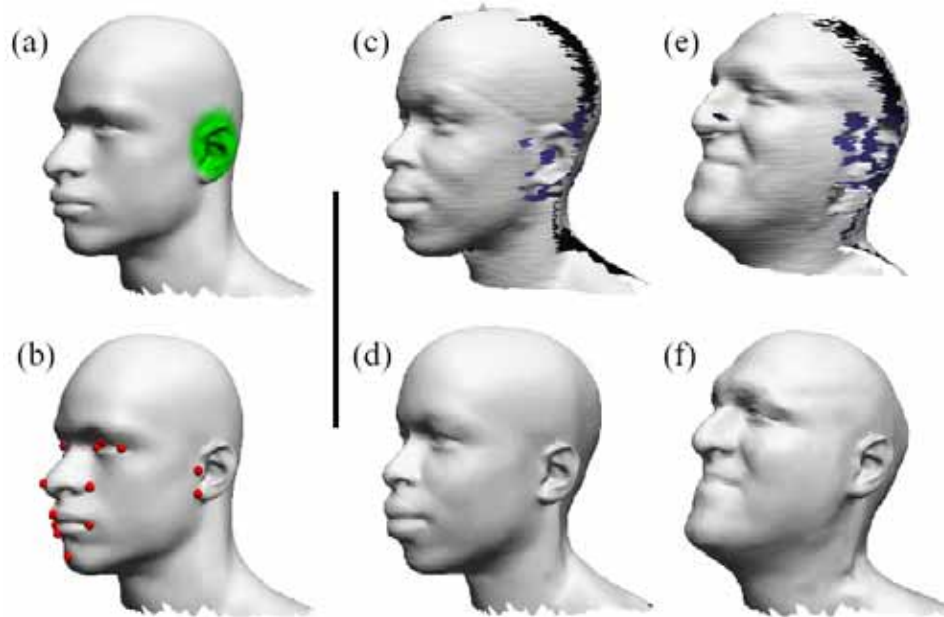


Figure 4.15: Poorly scanned areas, such as the ears and top of the head, are replaced by the prototype model. [ACP03]

Once the prototype was fit, many applications were possible. The point-by-point correspondence made transferring textures or skin weights from one model to another trivial (Figure 4.16). Knowing the locations of landmarks made it easy

to scale a character rig to fit a different character (Figure 4.17).

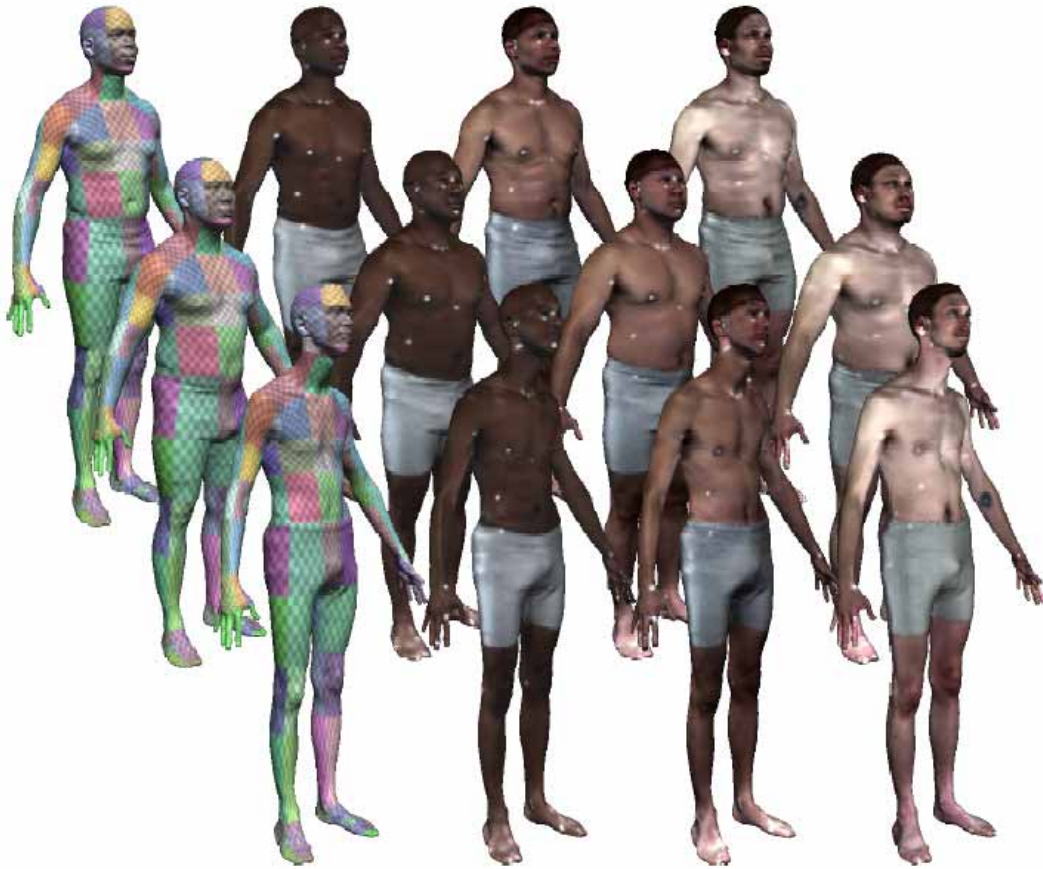


Figure 4.16: Textures are easily transferred from one character to another. [ACP03]

The approach also allowed creation of characters that were not a part of the original data set. Using PCA, they isolated parameters such as height and weight that could be used to transform the model in believable ways. This was important because as the fitness conscious know, fat does not appear uniformly all over the body but rather in certain trouble spots. The PCA approach ensured that if the user wanted to add fat to the body, it would be added in a natural way.

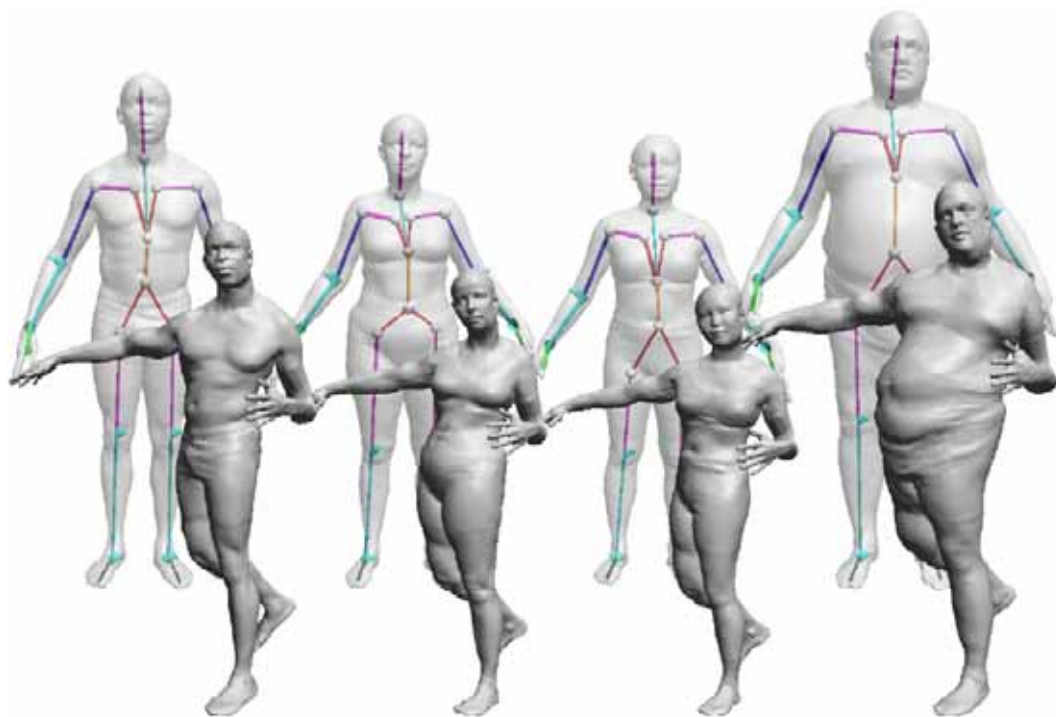


Figure 4.17: Likewise, rigs can be fit easily to new characters by using landmark positions as a guide. [ACP03]

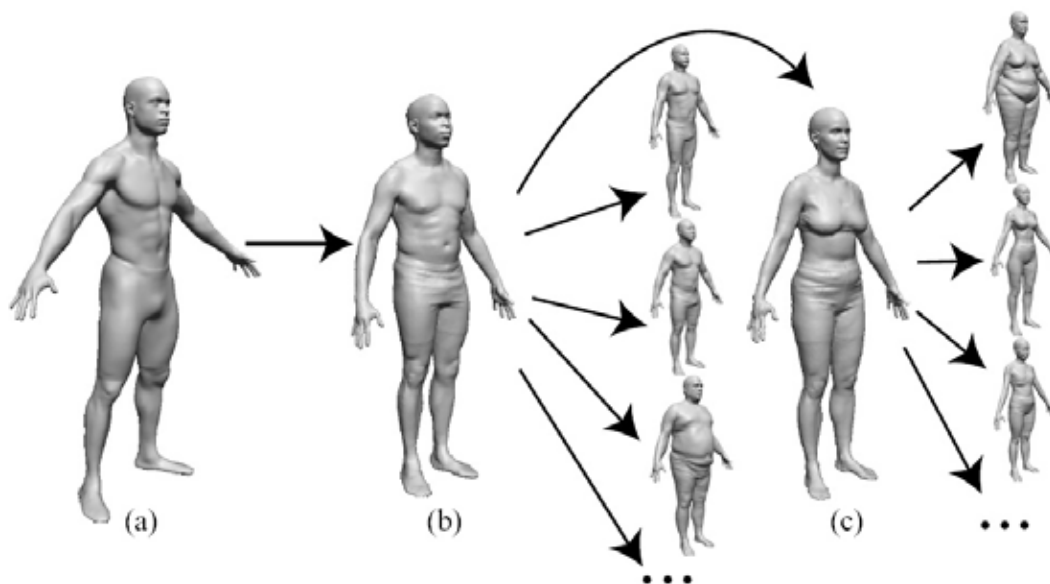


Figure 4.18: The prototype can be parametrically morphed in anatomically meaningful ways. [ACP03]

4.3 Sketch-Based 3D Modeling Interfaces

The invention of the graphics tablet revolutionized user interface design for artistic applications. The **tablet** is a touch-sensitive pad that works in conjunction with a digitizing pen, or **stylus**. By simulating the experience of using pencil on paper, the tablet-and-stylus system facilitates precise artistic tasks that are clumsy or impossible when using a mouse, such as drawing a smooth curve.

Low-end tablets, such as the WACOM Intuos shown in Figure 4.19, are essentially similar to touchpads designed to react only to stylus movements. For many users, this is a sleeker, ergonomic, and intuitive alternative to the mouse. However, low-end tablets still carry one major disadvantage, which is the physical distance between the user's hand on the tablet and the drawing that appears on the screen.



Figure 4.19: The WACOM Intuos provides a smoother alternative to mouse functionality, which makes it useful for tasks such as photo editing, shown above. [Int]

To facilitate the hand-eye coordination needed for detailed drawing, high-end tablets, such as the WACOM Cintiq shown in Figure 4.20 have built-in displays.

They may be pressure-sensitive, allowing the user to vary the darkness of a single stroke. By facilitating a truly hands-on experience and providing exceptional visual feedback, a monitor-tablet combination is the tool of choice for photographers and digital painters.



Figure 4.20: Tablets with built-in displays, such as the WACOM Cintiq, give users direct feedback and most closely simulate the experience of using a pencil. [Cin].

As tablets became popular among 2D digital artists, many wondered if their advantages could be extended to creation or modification of 3D models. The main challenge of interpreting 2D sketches as 3D forms is currently being examined by many researchers. In this section, we will describe the rise of **sketch-based user interfaces** in 3D graphics.

4.3.1 Freeform Design from Sketches

Systems such as SKETCH [ZHH96] and Teddy [IMT99] were among the first to present the idea of a "sketchy modeling". They were based on **gesture recognition**, which maps "expected" 2D sketches to predefined primitive 3D forms. For example, a roughly drawn circle would be interpreted as a sphere of the same diameter.

SKETCH, designed by Zeleznik, Herndon, and Hughes in 1996 at Brown University, defined gestures to generate a number of primitive shapes. As shown in Figure 4.21, the input strokes were simplified yet descriptive versions of what a user would draw on paper. After creating objects, the user could edit or combine them using another set of strokes. Using this interface, one could cleanly and efficiently "draw" a number primitive 3D shapes with usable topology. The set of creatable objects, however, was strictly limited. Also, it was necessary for users to take some time prior to using the system to learn the acceptable strokes.

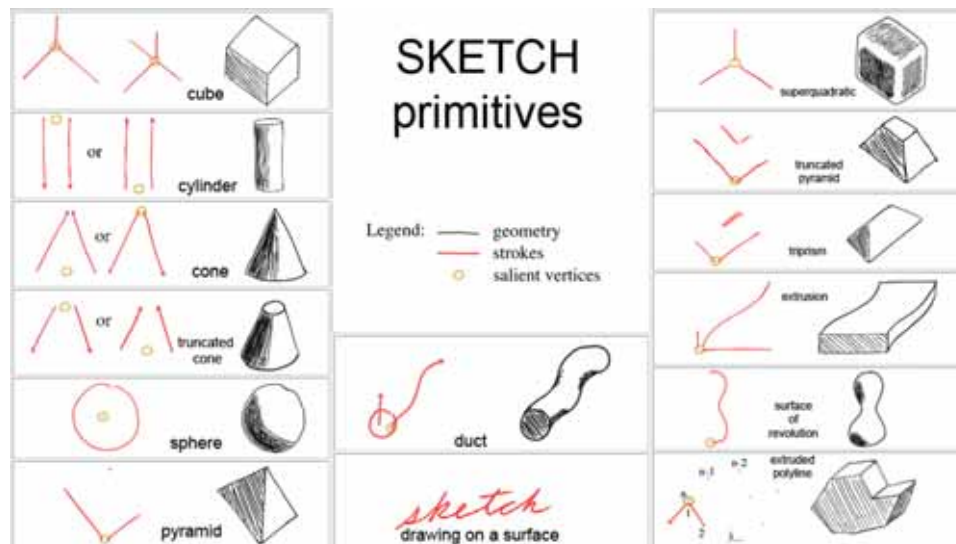


Figure 4.21: 3D Shapes that can be drawn using SKETCH. The gestures on the left produce the primitives shown on the right. [ZHH96]

The Teddy Interface was developed by Igarashi, Matsuoka, and Tanaka at the University of Tokyo to extend the idea of gesture recognition in a way that allowed for arbitrary curves. They implemented a **stroke reconstruction** algorithm that took 2D coordinates from the user's sketch and made an educated guess about the third dimension coordinates to generate a smooth, closed surface. Gesture recognition was used only for editing shapes once they were created; the system defined a small set of strokes for mesh operations such as extrusion, cutting, and deleting. By limiting the number of gestures that a user would have to learn, Teddy demonstrated that even a small child could "draw" an appealing 3D character in a short time, as shown in Figure 4.22.

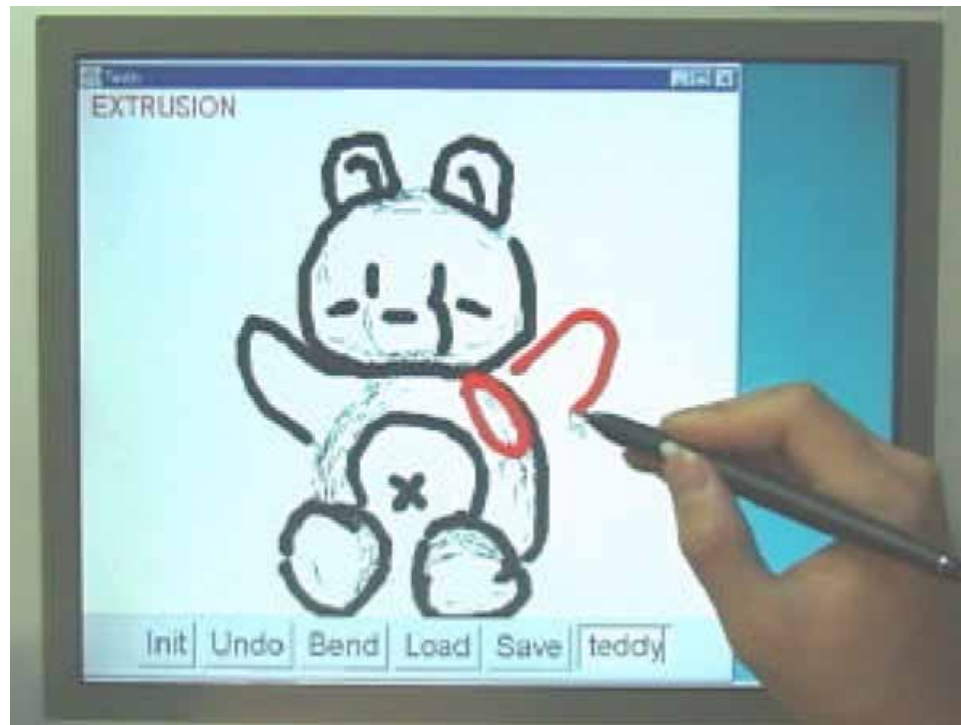


Figure 4.22: Teddy: A system is so simple that "even a child can use it." [IMT99]

While freeform stroke reconstruction interfaces offer artists a sense of unlimited freedom and flexibility, with the flexibility comes a lack of control over the topology.

Curve-reconstruction can be useful for static meshes, but the uniformly generated topologies generated on the fly by such algorithms are not generally suitable for animation. As intended, these systems greatly reduced the amount of time needed to create the rough shape of a mesh, but one must revert to traditional 3D modeling tools in order to fix bad topology and add fine details.

A notable cosmetic feature of these two systems was the usage of a **non-photorealistic rendering** (NPR) technique that only displays the outlines of an object to the user, with minimal shading (as shown in both Figure 4.21 and Figure 4.22). This made the screen look like a 2D drawing, further reinforcing the psychological experience of drawing on a flat surface. While the NPR display was visually appealing, it concealed topology problems in the generated meshes. The user could only see these problems after importing the mesh into a conventional modeling program.

4.3.2 Sketch-Based Mesh Editing

The limits on the amount of detail achievable by novel applications such as Teddy prompted a search for better ways to harness the power of sketch-based editing. In 2004, Nealen and Sorkine addressed some major challenges of manipulating an *existing* mesh with sketches [NSACO05]. Their interface provided a small set of tools that offered a large range of possibilities.

Nealen and Sorkine visualized the problem of sketch-based modeling as **”inverse NPR.”** Specifically, while non-photorealistic rendering transforms 3D objects to 2D space, their method translated 2D curves into appropriate positions in the 3D world. These algorithms require the detection of **silhouette edges** in the model, or edges that define the border between a front-facing polygon and a

backfacing polygon. After detecting the silhouette of the user-specified region of the model, the system calculated the transformations needed to map the silhouette onto a user input 2D curve. The 2D transformations were then mapped to the model. The transformation in the third dimension was interpolated by making a "guess" about the user's intention which was dependent on the position of the camera and the normals along the mesh.

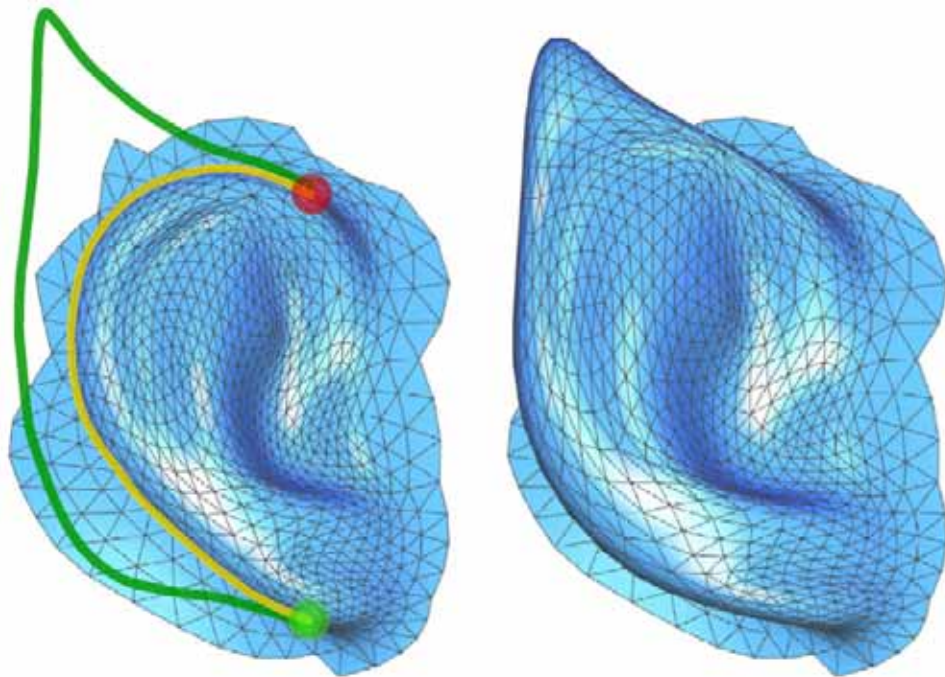


Figure 4.23: Mapping a 3D mesh to a 2D curve. [NSACO05]

Figure 4.23 demonstrates the precision of the silhouette-to-silhouette mapping. The system additionally allowed users to vary the precision of the fit, which introduced the option of sketching a global shape change without being concerned with details, as shown in Figure 4.24. Varying the level of constraint for high- and low-frequency mesh features could also be used to preserve local details while modifying the overall shape of the selected region on the mesh, as shown in Figure 4.25.

Another subset of tools facilitated sketching on the surface of the model to

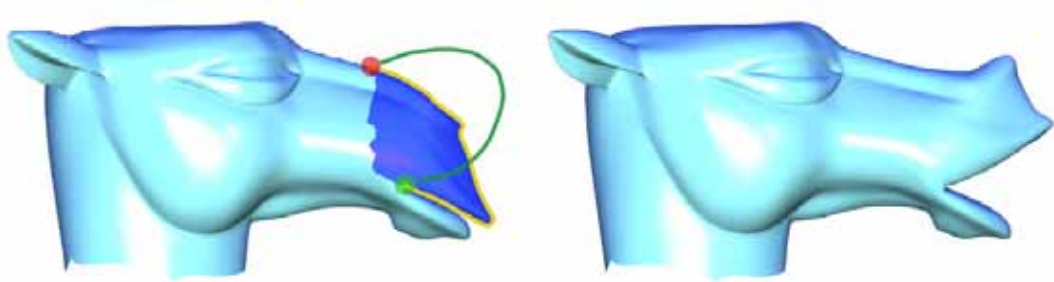


Figure 4.24: Loosening constraints allows users to sketch approximate curves for the new shape.[NSACO05]

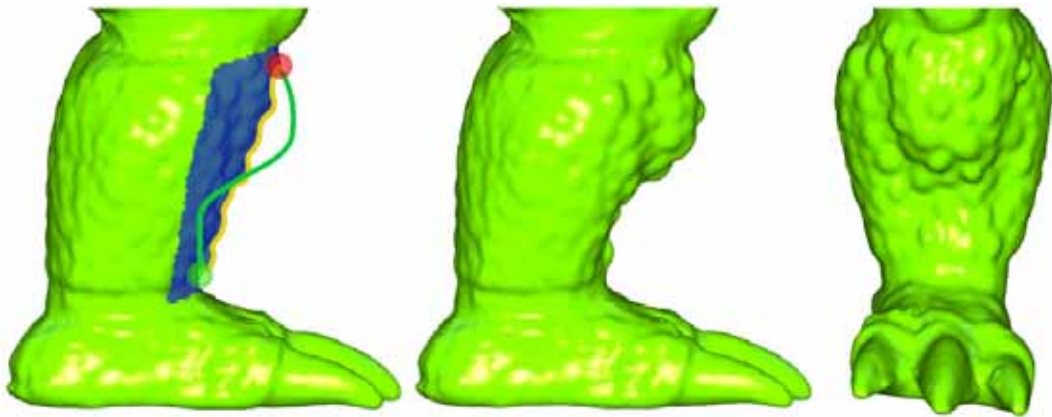


Figure 4.25: The leg retains its texture while the global shape is changed.[NSACO05]

create creases or ravines (Figure 4.26). The method recognized that the geometry needed for the desired surface feature might not exist on the original model. Rather than creating new geometry, the system rearranged the existing geometry by detecting the edges closest to the drawn curve, and repositioning them to follow the input curve. These edges could then be pulled out to create a crease or pushed inward to create a ravine.

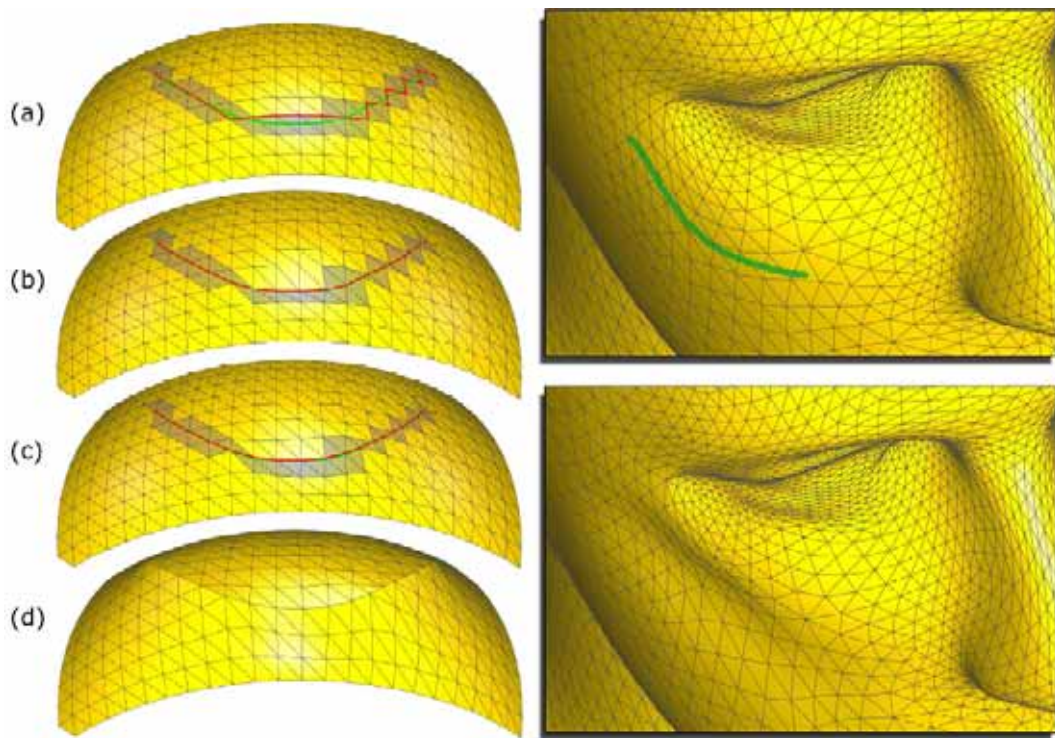


Figure 4.26: Geometry is rearranged to create the crease while maintaining a consistent, uniform topology. On the right, a Poser™ model has been given a more defined cheekbone by "sketching" a ravine. [NSACO05]

With an elegant, intuitive interface and impressive results, Nealen and Sorkine demonstrated that the power of sketch-based interfaces is not just for novel applications, but can revolutionize the way we use detailed meshes.

Chapter 5

The Prototype Model

To reiterate, the goal of our research is to simplify the process of creating a proportionate, animatable 3D human character by simulating the free expressiveness of 2D figure drawing. Sketch-based techniques are key to our implementation, as they achieve the latter part of our objective. However, we have seen that curve reconstruction modeling interfaces such as Teddy [IMT99] cannot easily be used to model meshes as complex as the human body, which require specific topology considerations. Thus, our system provides a topologically optimized prototype model, which is also equipped with a basic rig and simple customizable texture. In developing the prototype, our aim was not to detract from the artist’s role in the creation process, but to expedite the less creative, more procedural phases of modeling while providing a palette of possibilities. This chapter describes the first step in the development of our interface: the creation of ”Frankie”.

5.1 Geometry

Much literature has been published, both in print and on the web, to guide 3D artists through the painstaking process of creating a human model. For comprehensive advice and instructions on anatomical modeling and rigging, an interested reader can refer to *Building a Digital Human* by Ken Brilliant [Bri03], *Maya Character Creation: Modeling and Animation Controls, First Edition* by Chris Maraffi [Mar04], or the multitude of tutorials at 3DTot.com [3DT]. Our models, rigs, and animation were produced using Maya 6.5.

Frankie, named after another character who was brought to life by a well-meaning experiment, is our generic model. After exploring the advantages and disadvantages of various modeling methods in Section 2.4, we have opted to model Frankie using subdivision surfaces due to their flexible topology and the ability to generate smooth models from a relatively small number of points.

In the interest of providing a comprehensive prototype model that would not require further topology changes, Frankie was constructed at a reasonably high level of detail. The outer mesh, or **cage**, has 8,226 vertices. One level of subdivision produces a model that appears smooth; this smoothed mesh has 32,821 vertices. At this resolution, the model can still be manipulated interactively.

5.1.1 Face and Head

Figure 5.1 shows Frankie’s facial mesh. Emulating Parke’s optimal topology heuristic [Par72], the layout of edges simulates facial muscles.

The eyes and teeth were modeled separately from the body in order to facilitate their rigging. Individual meshes for the sclera, cornea, and tear duct form the eye.

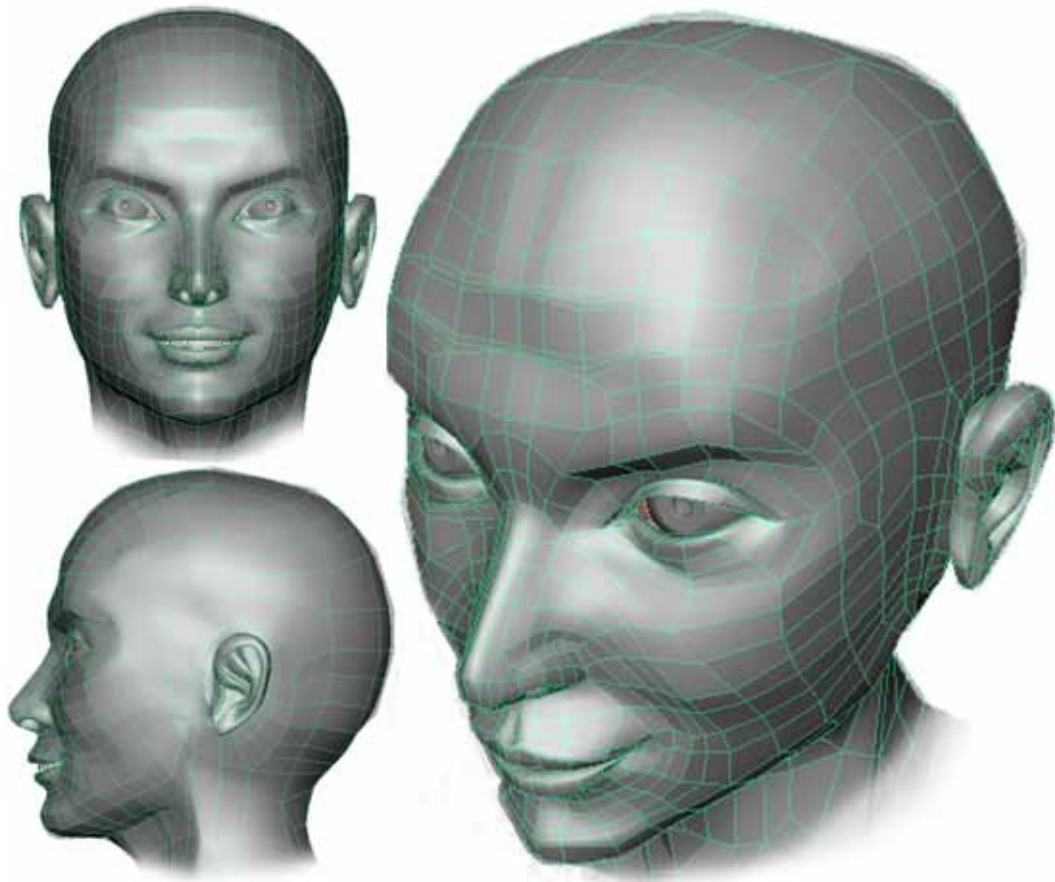


Figure 5.1: Frankie's facial mesh. The lines are the edges of the cage mesh, which is semitransparent in these images so the smoothed mesh, which is the cage mesh with one level of subdivision, can be seen. Radial patterns around the mouth and eyes ensure that the model will deform in the right places when animated.

A **geometry constraint**, which constrains one object’s movement to the surface of another, is applied to the tear duct to conform it to the corneal surface. The gums and the individual teeth were modeled as separate objects, and then combined to form just two meshes for easy manipulation: the upper and lower teeth.

While the inside of the mouth and tongue are also often modeled as separate objects, or sometimes left out for their infrequent visibility in animation, we have modeled them as parts of the facial mesh for added realism. These parts, along with the eye and teeth models, are shown in Figure 5.2.

5.1.2 Body

The full body mesh is shown in Figure 5.3. Due to its lower perceptual importance, the body was modeled in slightly less detail than the face. However, for texturing purposes, it was important to keep the sizes of polygons relatively consistent throughout the mesh. Anticipating the 3D scan-fitting potential of our system, detail is concentrated in areas that are known to scan poorly: eyelids, ears, fingernails, and toes. Figure 5.4 demonstrates the importance of detail in these areas for scan-fitting applications.

Like the face, the body’s vertices are streamlined along the muscular contours.

5.2 Rigging, Deformation, and Parameterization

Rigging was a crucial step in our process because the aim was not only to build usable animation controls, but more importantly to develop interactive handles upon which we could build our parametric modeling implementation. First, we will describe the generic animation skeleton and rig. Next, we will discuss how we configured rigging controls to facilitate our parametric, sketch-based modeling

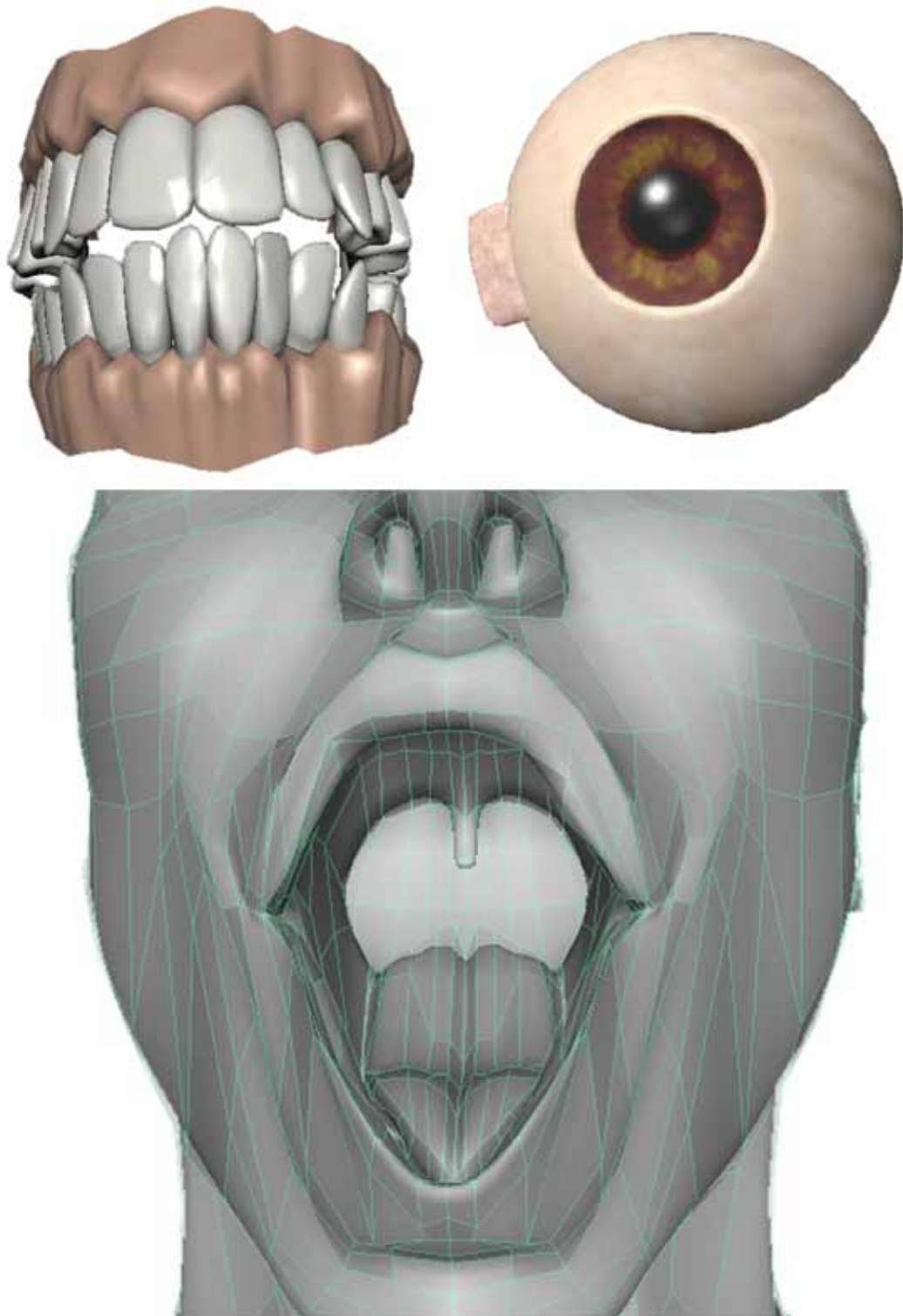


Figure 5.2: Frankie's teeth and eyes are separate from the main mesh. The inside of the mouth, including the tongue, is part of the main mesh.

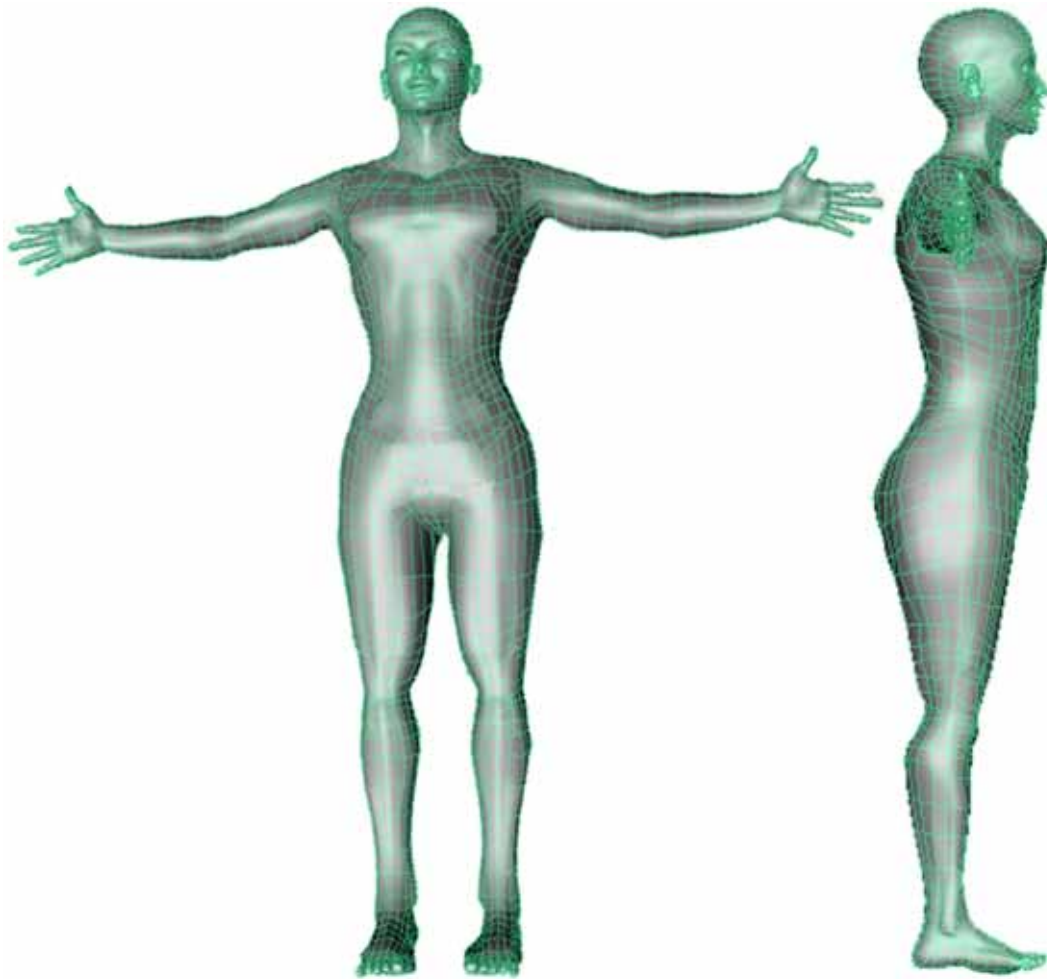


Figure 5.3: The full body prototype mesh. The edges are arranged to suggest muscular contours, including the biceps, quadriceps, pectorals, and abdominals.

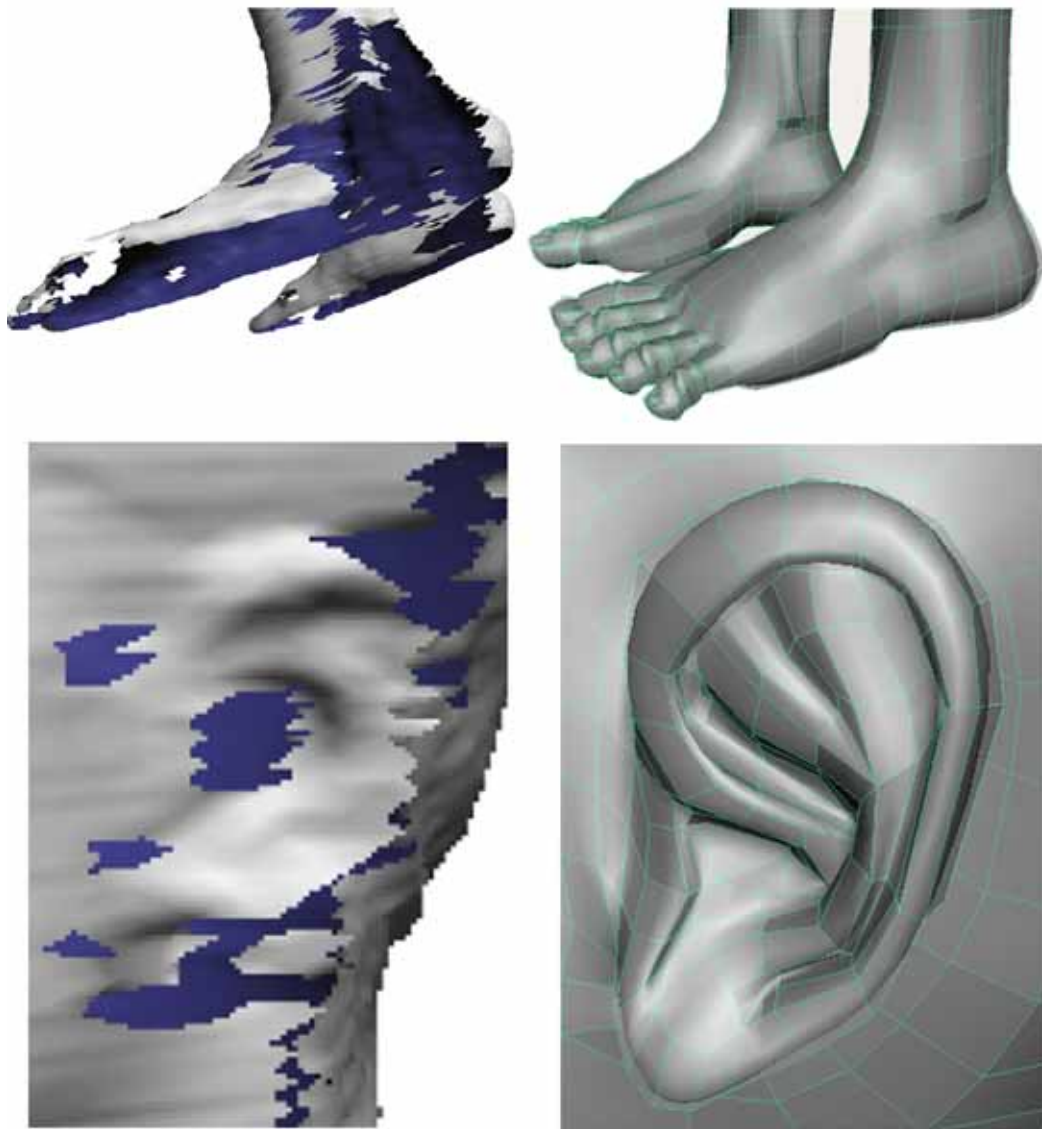


Figure 5.4: While 3D scanners can capture uncanny facial models, they often fail at capturing complex, concave, or overlapping details. Our prototype mesh provides details that could be used to replace these areas. Scans shown are from the CAESAR dataset [CAE], the images of the scans are courtesy of [ACP03].

method.

5.2.1 The Animation Controls

Maya provides an array of controls and deformers that a creative user can piece together to form a robust character **rig**. The most basic rigging mechanism is the **joint**. Essentially, joints are transformation nodes that can be attached to other joints hierarchically. Visually, a hierarchy of Maya joints looks like a skeleton, but depending on how the user chooses to define controls, they may or may not behave like human bones. These "joints" can be rotated, but unlike real joints, they can also be scaled. Also, when a Maya joint is moved in space, its parent joint retains its position, which changes the distance between the parent and the child. Note that this is different from scaling.

Frankie's skeleton, shown in Figure 5.5 is a simplification of the human skeleton that includes only the joints that have a visible effect on movement, such as the knees and ankles. Internal bones, such as the ribcage, are not necessary.

A rig composed of joints only can be posed using **forward kinematics**. This simply means that the user must rotate each joint individually, starting with joints highest in the hierarchy and working down to the leaves to achieve the desired pose. This is not intuitive for longer joint chains; when animating a walk cycle an artist would prefer to position the feet first, which are leaves of the leg hierarchy, rather than rotating the hip, knee, and ankle first. **Constraints** can be applied to the skeleton in order to build intuitive animation handles.

An **inverse kinematics**, or IK, control facilitates the use of a leaf joint to change the positions of joints above it on the hierarchy. Attaching an IK handle to a hip and ankle bone allows the user to position the foot bone with the hip



Figure 5.5: Frankie’s skeleton, on the right, is shown next to an anatomical skeleton. In an animation mesh, only moveable bones are necessary.

joint in place, while automatically calculating the rotation of the in-between knee joint. The new 3D position of the knee is the **IK solution**. However, a difficulty arises because the IK problem may have an infinite number of solutions. Given specific locations of the hip and ankle, the *direction* of the knee is not specified by IK; thus, the knee could be anywhere on a sphere of points between the hip and ankle. This could cause the knee to bend backwards during animation. The user must add additional constraints to the angles between bones to avoid these unwanted effects, but depending on the degrees of freedom, there may still be a large number of possible solutions.

Another problem with an unaltered skeleton is that because bones lie inside the mesh, they are hard to see and access. For this, **dummy objects**, or non-renderable locators and curves, can be placed outside the mesh and used as handles for the unreachable bones. They can be attached to the bones using several types

of constraints:

- **Point Constraints.** These allow the position of a target object to control the position of a source object, either exactly or relatively.
- **Orientation Constraints.** Rotating the target object rotates the source object.
- **Aim, or "Look At" Constraints.** As the name implies, the source object is constrained to "look at" the target object, which means the translation of the target controls the rotation of the source. These are often used to animate eyeballs rolling.
- **Mirror Constraints** We implemented mirror constraints for our interface in order to localize symmetry on our model. It takes four inputs: a source object, a mirror object, a reflected object, and an axis of reflection (x, y, or z). The reflected object is constrained to stay the same distance away from the mirror object as the source, in the opposite direction along the specified axis.

Our rig provides IK handles for the limbs, aim constraints for the eyes, a jaw rotation control, and controls for rotating the head, neck, spine, and waist. The rigging needs of individual characters can be quite unique, but most human character rigs require these standard controls. Users may choose to add extra rigging controls on top of our setup.

To attach the mesh to the bones, a **skin deformer** was used. The skin deformer associates each joint with a set of vertices on the mesh. The initial assignment is performed by Maya and then must usually be altered by the user to achieve realistic

deformation effects when a joint bends. There are two methods of skinning: rigid and smooth. In a **rigid skin**, only one joint can influence each point. In a **smooth skin**, a point can be influenced by more than one joint. Each joint has a weighted influence on the given point such that the sum of all weights equals one.

Rigid skinning results in sharp creases on the mesh at the transitions between the underlying joints, which makes smooth skinning preferable for organic models. However, it takes a substantial amount of time and error-checking to specify skin weights in a smooth skin, and there are some effects that are impossible to achieve with skinning alone. For example, the skin deformer cannot realistically capture the way the skin on the inside of the elbow deforms when the elbow bends. A rigid skin causes the mesh to self-intersect; a smooth skin causes the elbow geometry to "cave in" as it is bent. To solve this problem, auxiliary deformers can be applied on top of skin.

For Frankie, we used smooth skinning, but did not apply any auxiliary deformers to the mesh. Our modeling interface, which Chapter 6 will discuss in depth, can be used to undo unwanted skinning artifacts.

5.2.2 Parametric Modeling Setup

We have designed a novel setup, which blends rigging and modeling, to facilitate sketch-based changes to the prototype interactively at any point in time, enabling changes to Frankie's geometry to flow seamlessly through the rest of the animation pipeline. Our setup considers a few key observations about the protocol for modeling a digital character:

- *As a rule, characters are modeled in the standard pose.* For a human character, this means the character is standing facing forward with

arms outstretched at right angles to the body, and with feet flat on the ground, as shown in Figure 5.3. Typically, the Z axis in the 3D world is defined as "front" and the Y axis is "up". This setup makes the model easily accessible for rigging.

- *While modeling a character, artists generally make the most significant changes while looking at the front, side, and top views, using the perspective view for tweaks and corrections to the 3D structure. In a studio, concept artists provide drawings in these canonical views.* Betty Edwards observed that many common drawing mistakes occur in perspective views, as shown in Figure 2.7, because we subconsciously rotate 3D objects to the most descriptive canonical views. This implies that we understand their structure best in these views [Edw79]. Thus, modeling in standard pose carries the additional advantage of giving artists access to silhouettes of the model that most clearly and concisely describe it. Our system makes use of these primary contours and gives users constant access to the standard pose.
- *The order of deforming operations matters.* For example, bending an object before twisting will result in a different shape than if one were to twist it first and then bend it. In relation to our setup, this means that if we want a user to be able to manipulate a model in the standard pose while the model is being animated, any modeling deformer must be applied before the skin deformer, not necessarily in time, but in the mesh's deformation chain.

Given these observations, we propose a modeling "rig" known as the **Gesture Sketch Harness**.

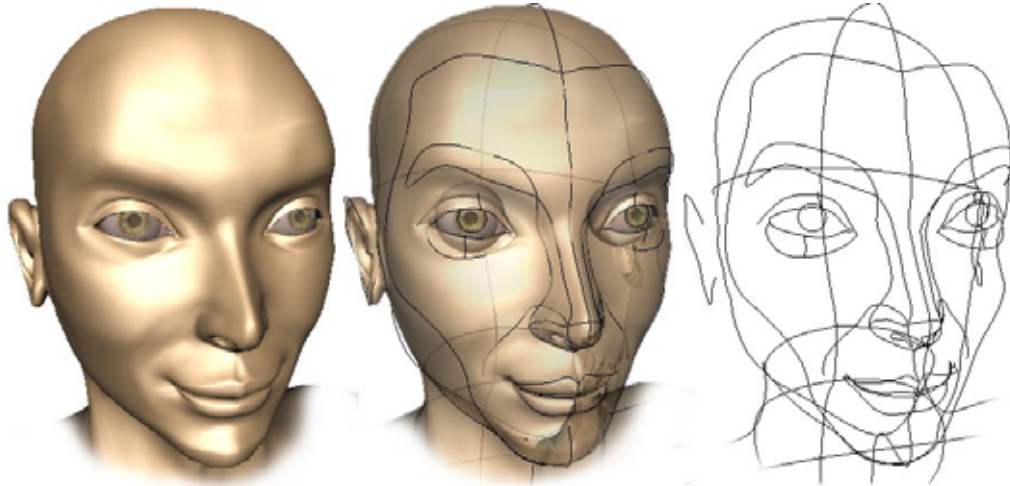


Figure 5.6: The Gesture Sketch Harness for the face, shown on the right, was constructed by drawing 3D curves along the major silhouettes of the original mesh, shown on the left. The middle image demonstrates their correspondence.

The Gesture Sketch Harness (GSH)

To bridge the gap between 2D and 3D "drawing", the foundation of our modeling technique is a "gesture sketch" representation of Frankie. As the name implies, the Gesture Sketch Harness (GSH) looks like an artist's quick 2D line drawing of a person, which captures the basic volumes and descriptive contours (Figures 5.6 and 5.7). It functions on several levels:

- **Visualization.** When viewing the wireframe of a detailed polygonal mesh, one may easily become overwhelmed by the plethora of vertices. The wireframe is displayed whenever an object is selected, which means the user must constantly select and deselect the model in order to see the true effect of vertex modifications on the smooth mesh. Teddy [IMT99] and Sketch [ZHH96] used NPR to interactively display their models as 2D drawings, acknowledging the perceptual benefits of making a sketch-based interface look "sketchy". On an abstract level, the GSH functions as a simple, elegant

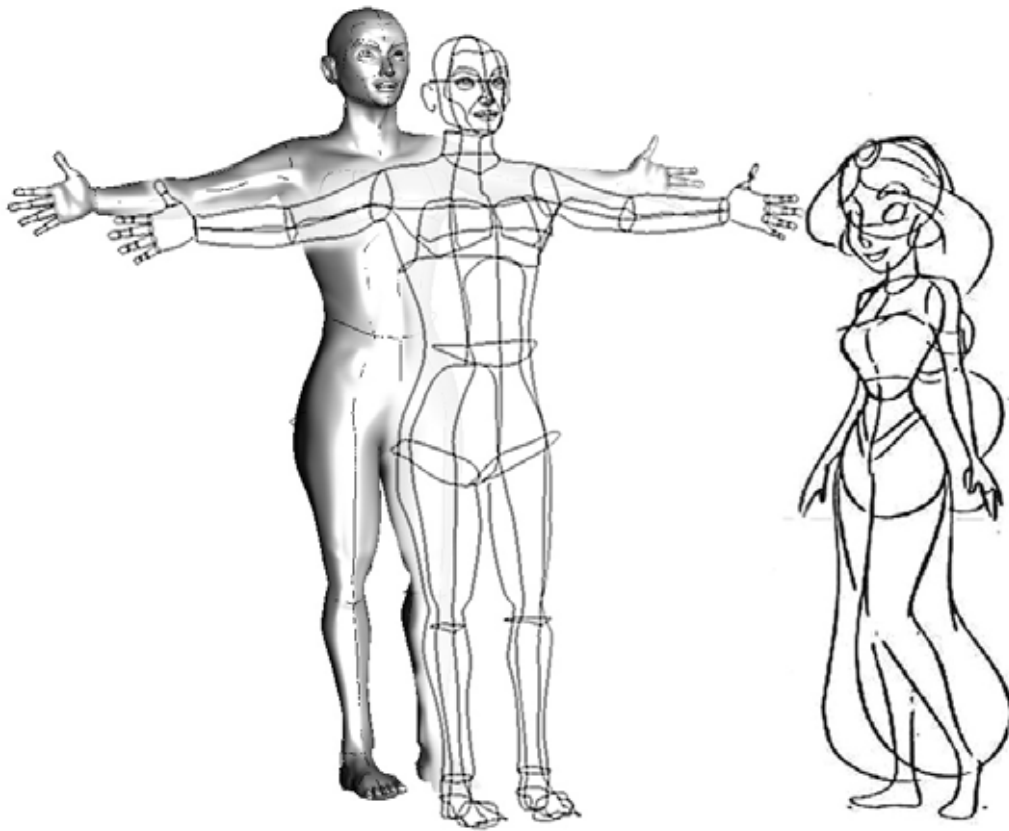


Figure 5.7: In the background, Frankie is shown with the full-body Gesture Sketch Harness superimposed. The GSH appears in the foreground. It is composed of 3D curves, but in still images it is reminiscent of a quick, preliminary figure drawing. For comparison, a concept sketch of Jasmine from Disney's *Aladdin* is shown. Note the similarity in the way volumes and silhouettes are conveyed. Concept sketch courtesy of [Dis92].

alternative visualization that displays the model in recognizable silhouettes. Although in reality the GSH is composed of 3D curves and thus does not show exact perspective projections, it works well in the orthographic views. In 3D views, it also reduces the complexity of a wireframe view by providing a more concise, understandable visualization without expensive NPR calculations (Figure 5.8).

- **Deformation.** Each "silhouette" in the GSH is actually a 3D NURBS curve. The curves were initially drawn through precise vertices on the mesh, and then applied as **wire deformer**s to the cage mesh. A **wire deformer** is essentially a "magnetic" curve associated with a region of a mesh that "attracts" nearby vertices to conform to the shape of the curve, with varying degrees of influence on surrounding vertices. When the shape of the curve is changed, the region of influenced vertices conforms to the reshaped curve. Thus, by moving the control vertices on the curves of the GSH, the user can change the outlines of the mesh precisely and smoothly, as shown in Figure 5.9. The subdivision cage is already a generalization of the smooth model; the GSH builds a second level of generalization on top of the cage with a drastically lower number of points.
- **Automated Rigging.** To convey volumes, some curves in the GSH are drawn around the circumferences of significant joints: the hips, knees, ankles, shoulders, elbows, wrists, waist, neck, and skull base. Our automated rigging system reads the positions of these roughly circular curves and then generates a skeleton by mapping a predefined set of joints to the geometrical centers of corresponding curves (Figure 5.10). The associations between the joints

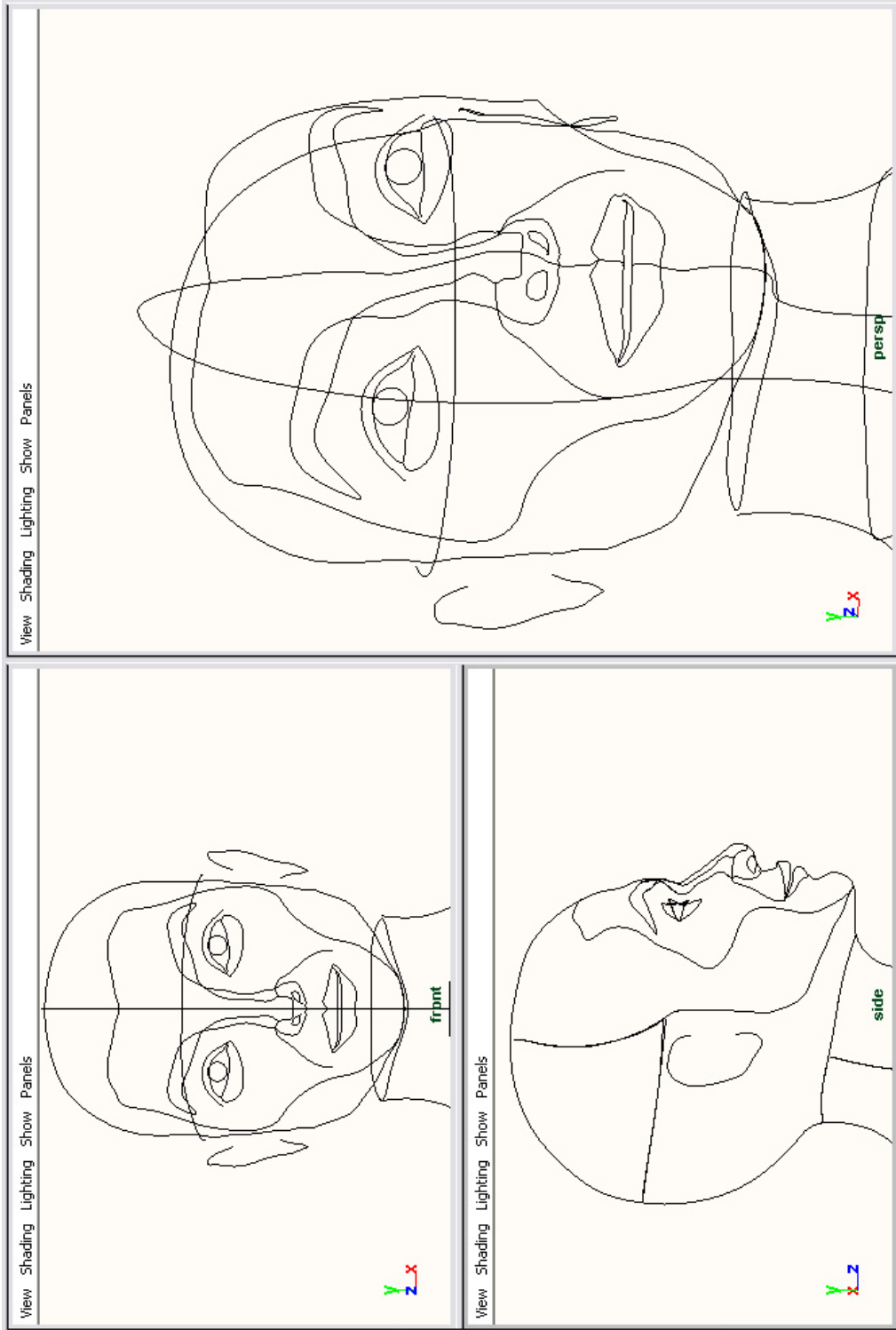


Figure 5.8: The GSH face in front, profile, and perspective views of the GSH setup are shown, clarifying the 3D structure of the curves. Visually, it is a "drawing" that can be rotated in 3D space.

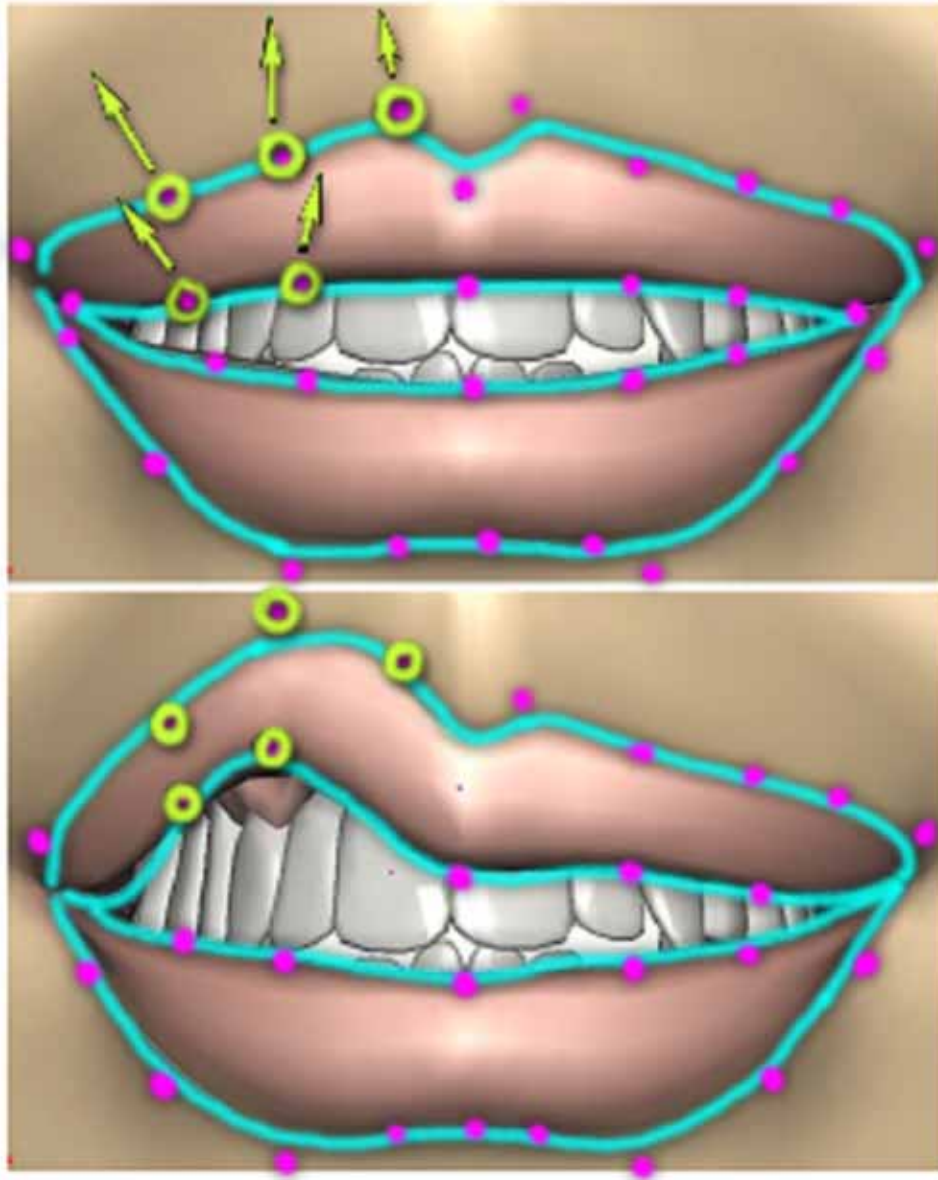


Figure 5.9: When applied to a mesh, a wire deformer can be used to interactively reshape the region of the mesh within its range of influence to conform smoothly to the target NURBs curve. In this example, the NURBs curves displayed in blue and represent the silhouettes of Frankie’s lips; the control points of the curves are displayed in magenta. By moving just a few control points, the user can make Frankie impersonate Elvis.

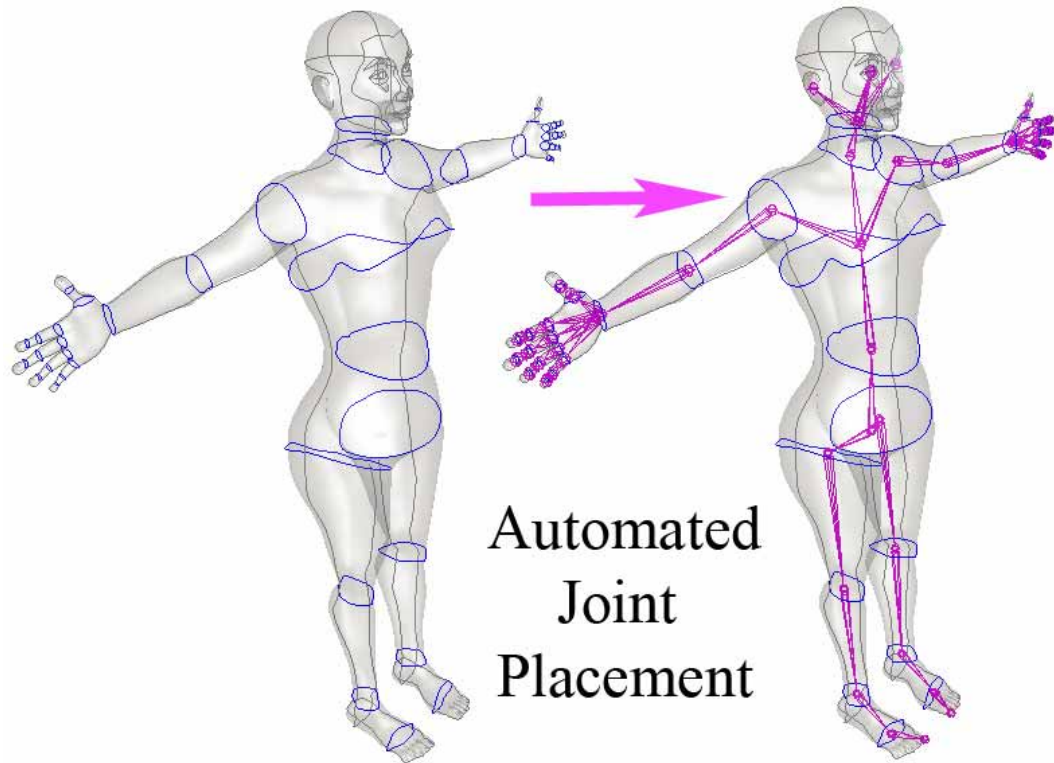


Figure 5.10: Joints are automatically placed at the geometrical centers of radial GSH curves, highlighted in blue, whose names correspond to the names of the joints.

and the curves are maintained in order to facilitate automatic repositioning of joints when the size of a limb is changed.

- **Concurrency in the Pipeline.** Since the user may wish to continue manipulating the model's shape in the standard pose while it is possibly being textured, skinned, and animated by others, the wire deformers in the GSH are applied before the skin deformer. Consequently, the GSH maintains standard pose even when the rigged model is moving.

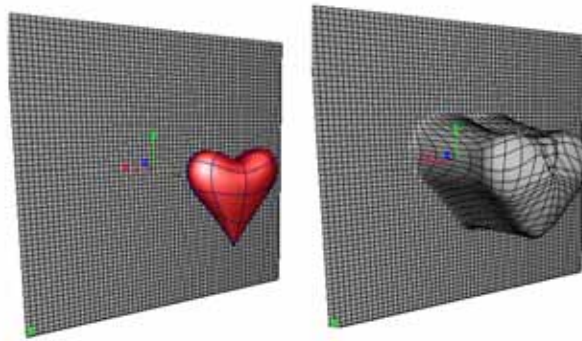


Figure 5.11: The heart shape, which is actually a modified sphere, can be used as a sculpt deformer on the plane to create a popular cartoon effect. In our setup, sculpt deformers are used to preserve volume.

Preserving Volume

In old cartoons, to emphasize a lovestruck character’s rapid heartbeat, an animator would literally make the character’s heart pop out of his chest. In Maya, this effect can be achieved with a modified **sculpt deformer** (Figure 5.11). The sculpt deformer consists of a spherical base object, defined as the **sculptor**, that is used to push out points on another object (a mesh or a curve). The artifacts in the heart example demonstrate that while the sculptor works best with perfectly spherical models, slight deformations of the sphere are permissible.

To preserve the anatomical coherence of our model, the body must maintain a minimum volume, and sculpt deformers can be used to achieve this. The ”chopped off skull” error, discussed in Section 2.2.1, can be avoided in 3D by placing a sculpting sphere inside the skull. We can use modified, stretched spheres to maintain the limbs’ volumes. Since sculptors work by calculating intersections between each affected point and the sphere, they can become expensive. To avoid excessive computation and still get reasonable results, we applied sculptors to the ”lightweight” GSH curves rather than the dense mesh. Sculptors were placed at the skull, joints,

abdomen, and chest, where the bone structure would prevent a character from getting any thinner. They were also applied to the eyelid GSH curves to prevent intersection with the eyeballs, and to the lips to prevent intersection with the teeth.

For his facial rig, Bibliowicz implemented a useful generalization of the sculpt deformer known as the **push-to-front deformer**, which could use an arbitrary, non-spherical mesh as the sculptor [Bib04]. Theoretically, the push-to-front deformer could be used to model an entire bone and muscle geometry underneath the model, which would be valuable for medical simulations. Intersecting an arbitrary surface, however, is more expensive than intersecting a sphere. For our model, Maya’s spherical sculpt deformers suffice.

Landmarks

To further prune the number of controls and provide handles for the parametric modeling component of our system, we implemented **Landmark locators**. For clarity, whenever we refer to "Landmarks" with a capital "L", we are referring specifically to Landmark locator objects. Landmarks function as placeholders for significant anthropometric locations on the face and body. We used a subset of the landmarks described by Farkas [Far94] (Figure 5.12). Because anthropometry does not represent all the perceptually significant areas of skin curvature, it was necessary to add a few non-anthropometric Landmarks, including the deepest part of the eyelid crease and the "bulge" of the cheekbone.

Landmarks provide another way to control the shape of the model. For efficiency, Landmarks are applied as deformers to the GSH rather than the mesh. Using a configuration similar to Bibliowicz’s wire-based rig, [Bib04], weighted cluster deformers were applied to groups of points on GSH curves, and then the clusters

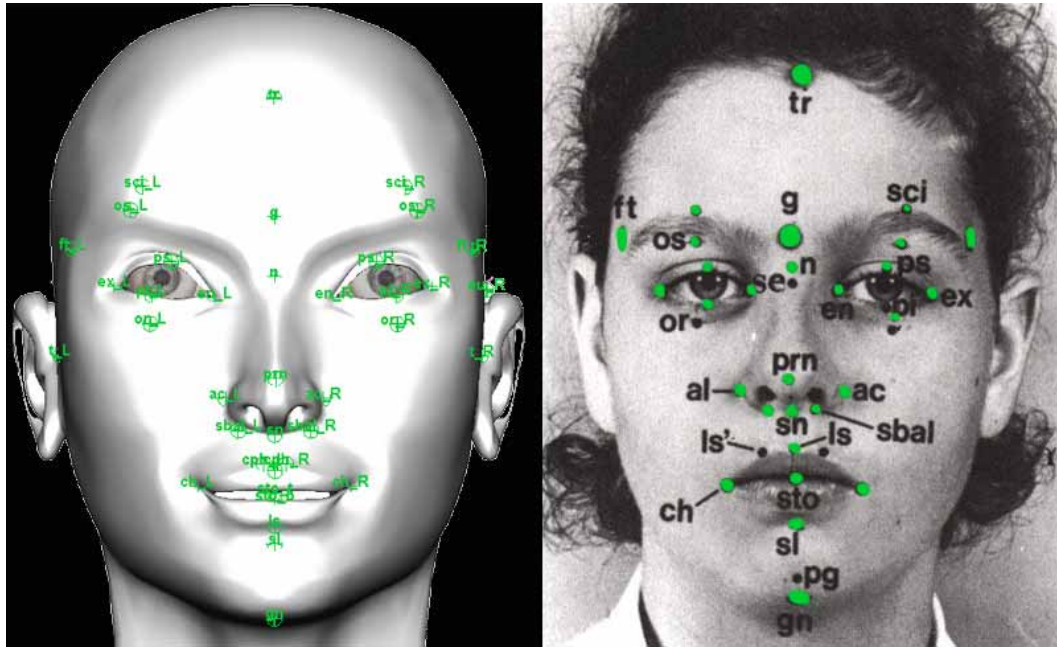


Figure 5.12: A side-by-side comparison of Frankie’s facial Landmarks and the anthropometric markers defined by Farkas [Far94].

were parented to corresponding Landmarks. To maintain smooth transitions, the weight of a cluster deformer’s influence on a given control point is inversely proportional to the control point’s distance from the Landmark (Figure 5.13).

Thus, the user can move Landmarks interactively to deform the GSH (and consequently the smooth model) in anthropometrically meaningful ways. For example, to change the length of the nose, a user can simply move the ”prn” Landmark ¹ forward, as shown in Figure 5.14.

While Landmark-based modifications to the mesh allow the user to control many anthropometric parameters, they do not encompass the full range of variation of the human face. Altering both the Landmark locations and the control points of the GSH curves between them, however, can produce a vast variety of results. Once again, this is much simpler than working with vertices on the dense mesh. To

¹”PRN” is the abbreviation for *pronasale*, which is the anatomical name for the tip of the nose.

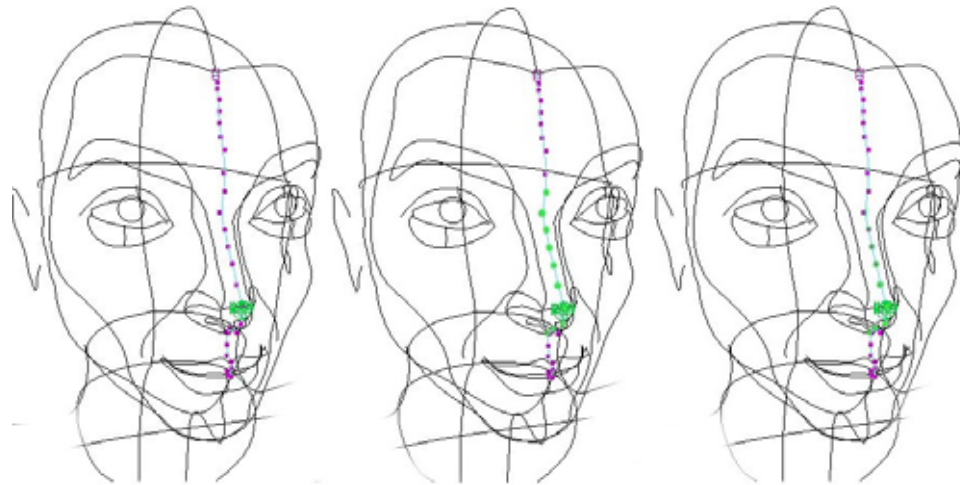


Figure 5.13: This figure demonstrates how Landmarks are associated with GSH curves. On the left image, the control points of the facial profile curve of the GSH are displayed in magenta, and a Landmark has been placed at the tip of the nose. In the middle image, the control points highlighted in green are associated with the Landmark. Finally, on the right image, the influence on each associated control point has been weighted with a falloff rate inversely proportional to the point's distance from the Landmark. The "greener" the control point, the more closely it will follow the Landmark's movement.

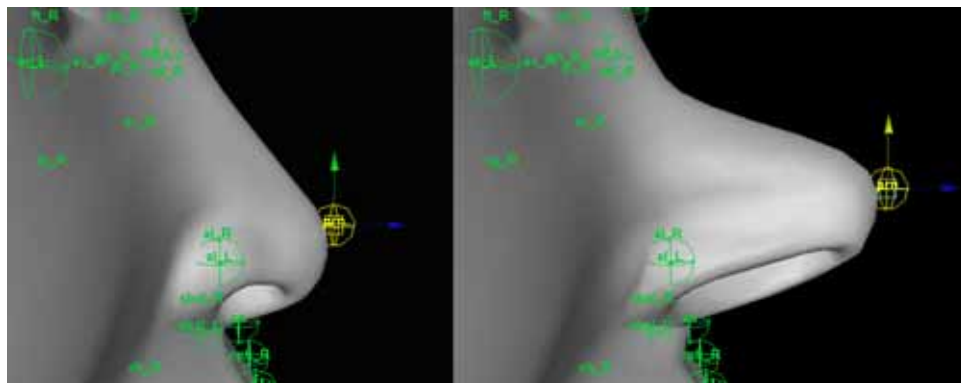


Figure 5.14: The user can smoothly "grow" Frankie's nose by moving just one landmark.

build a final level of simplification, our sketch-based interface will further condense the Landmark and GSH manipulation modes into one intuitive step.

5.3 Texture

Parke’s parametric face model considered color as a significant **Conformation Parameter** [Par82]. Because color affects the way light reflects off a surface and consequently influences our interpretation of a facial geometry, our model is equipped with a customizable texture. The prototype model’s unwrapped UV texture map is shown in Figure 5.15. Note that the face and arms in the UV map are enlarged in proportion to the rest of the body, as they are generally textured in the most detail. The torso and abdominal area are also relatively large in the UV map to allow for clothing.

Maya provides a convenient interface for interactive UV texturing using Adobe Photoshop™; the UV map was first unwrapped in Maya and then exported as a PSD file. Maya can automatically generate Photoshop layers for each texture attribute, including color, bump, specular, transparency, translucency, etc. Changes to the maps in Photoshop can be viewed interactively in Maya.

We provide a number of predefined textures to encompass a wide range of hair and skin color variations, as shown in Figure 5.16, and for the eyes, shown in Figure 5.17. Our ”texture chooser” UI allows efficient customization of the model’s global coloring (Figure 5.18). The user may also directly alter any of files to create more complex textures and delete unwanted layers. For example, one may wish to delete the eyebrow, hair, and clothing maps in order to apply more realistic hair or cloth simulations. The layered structure of a Photoshop file gives local control over each subset of texture parameters.

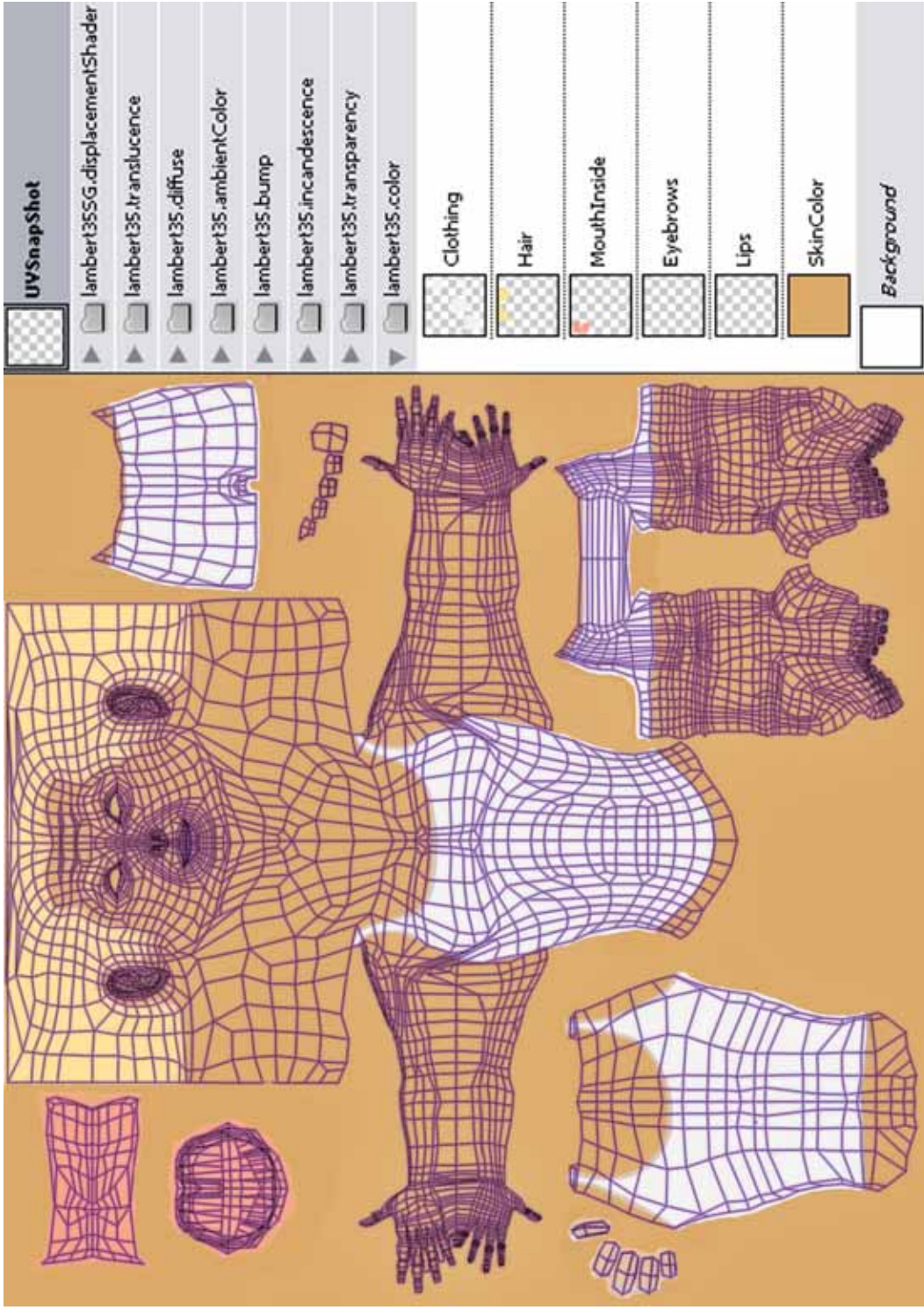


Figure 5.15: Frankie's UV map. Various texture attributes, listed on the right, can be painted by the user.

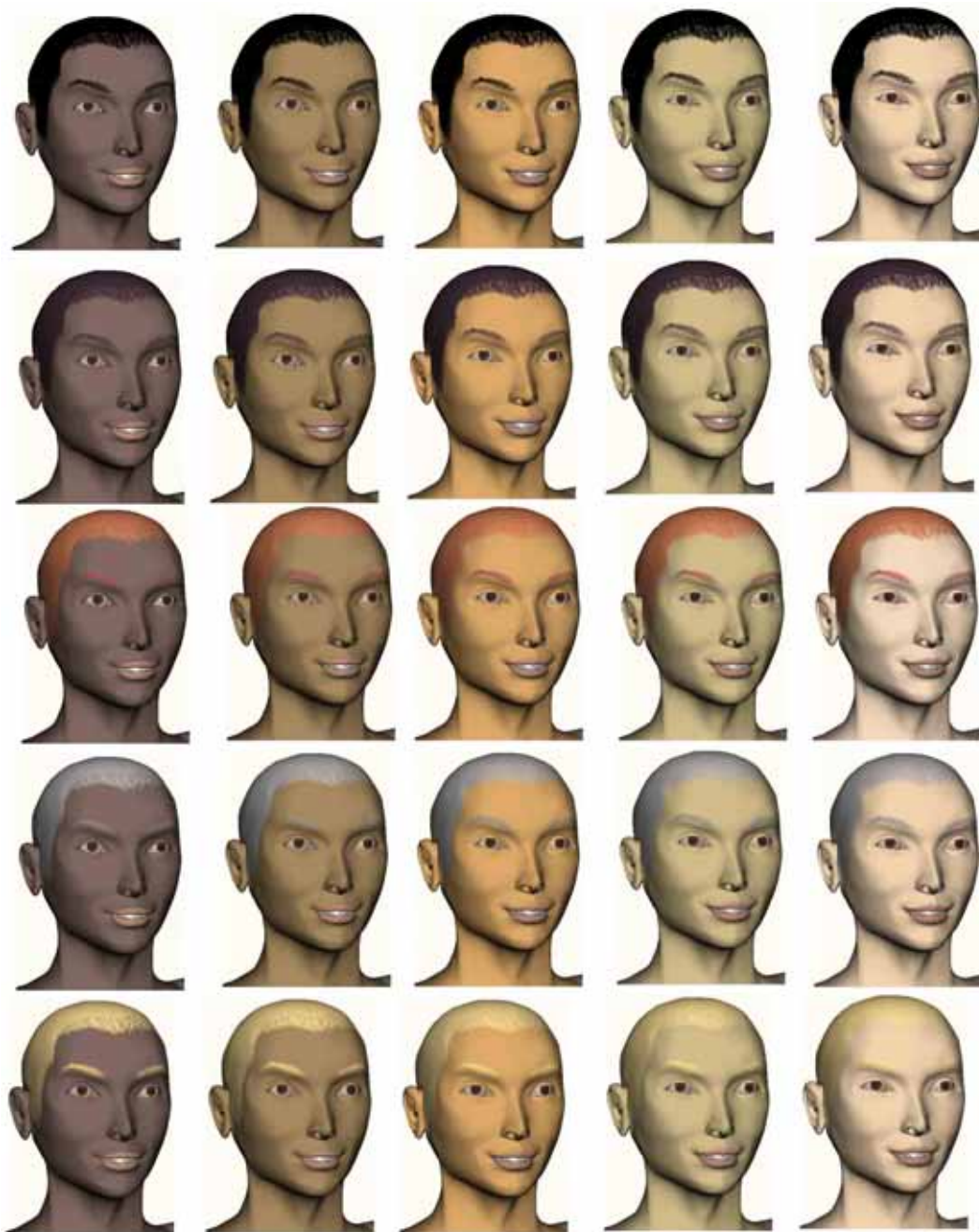


Figure 5.16: Variations to Frankie's skin and hair colors. These textures are provided by our system.

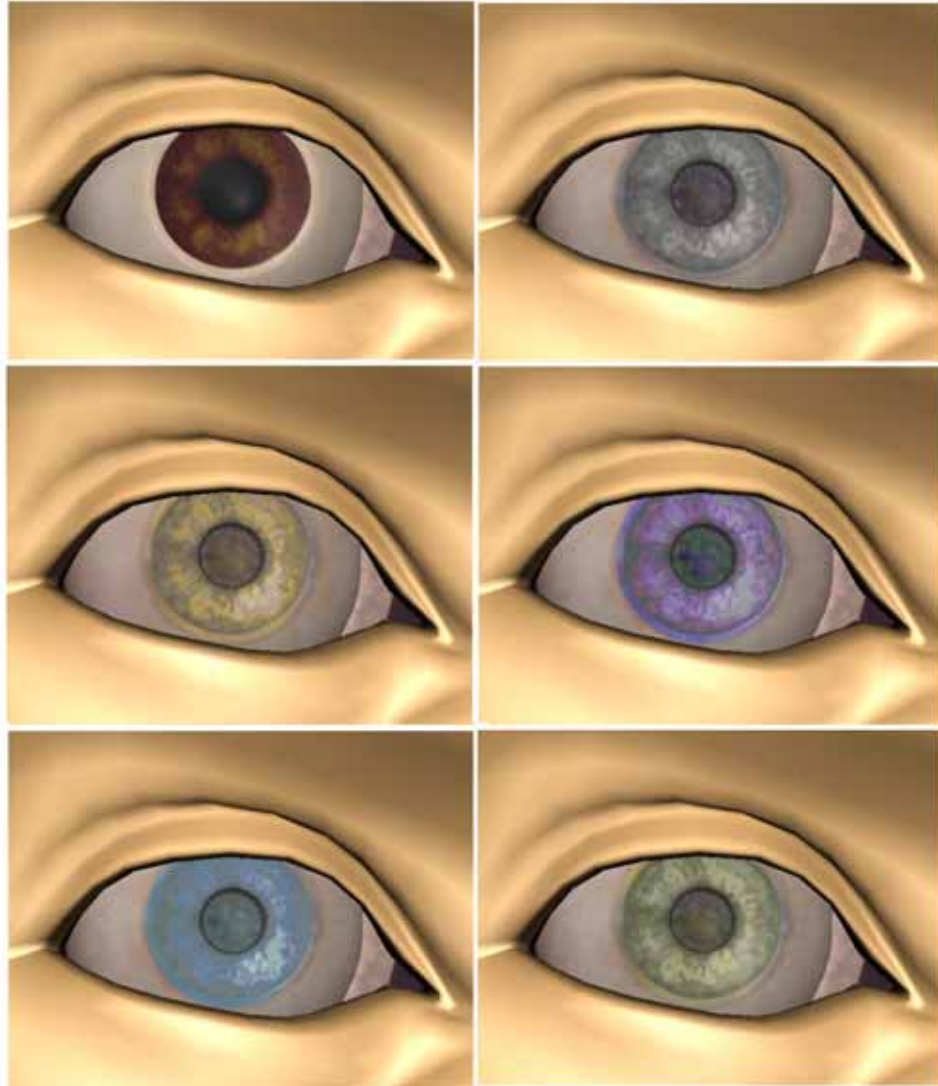


Figure 5.17: Predefined eye textures for Frankie.

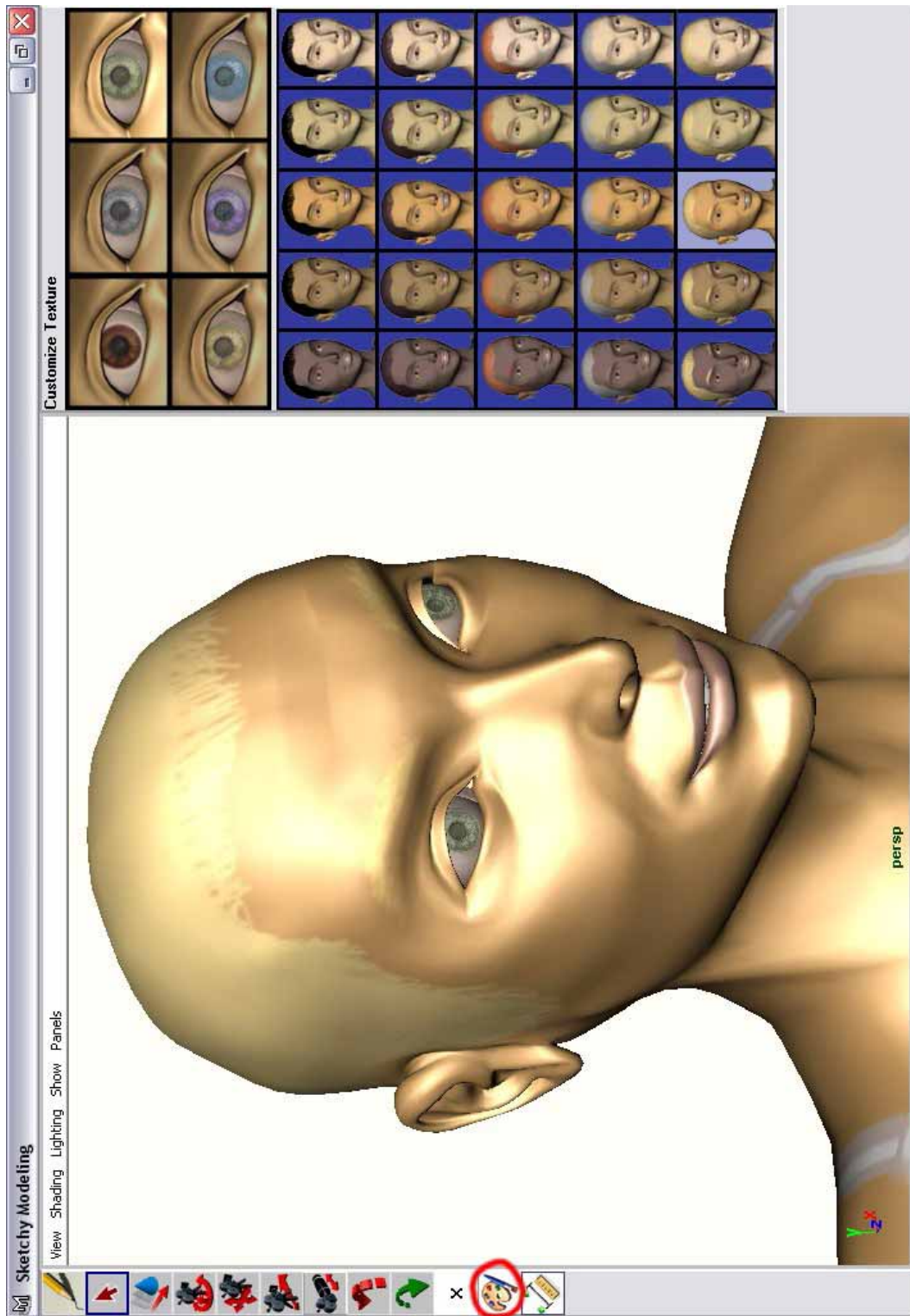


Figure 5.18: Our interactive texturing UI is accessed by pressing the "palette" button circled in red.

5.4 Summary

The time it takes model, rig, skin, and texture a high-quality mesh is nontrivial, motivating the search for better methods to reuse a well-constructed prototype. The configuration of our prototype already simplifies much of the process by building layers of abstraction over the original vertex set. The Gesture Sketch Harness provides a compact, recognizable, easy-to-manipulate level of generalization on top of the complex subdivision mesh. Landmark locators on top of the GSH further shrink the number of points that the user must work with in order to create a usable new geometry. The final level of simplification, the sketch-based interface, will be covered in the next chapter. The focus of this chapter was to alter the **representation** of points to be moved; our sketch-based interface aims to streamline the **process** of moving them.

Chapter 6

Implementation

Our implementation of the "Model-Sketch" system includes tools and data structures for

- Representation of anthropometric landmarks.
- Visualization of anthropometric constraints.
- Sketch-based editing of a prototype model.

This chapter decomposes the prototype system into its components and describes the fundamental concepts and algorithms in depth.

6.1 Maya Architecture

The software was implemented in C++ as a plugin module for Maya 6.5; the GUI design used Maya's Embedded Scripting Language (MEL). To explain the basis for our implementation, this section briefly introduces the architecture of Maya.

All objects in a Maya scene file are represented as **nodes** in a graph. Nodes include shapes, cameras, animation curves, textures, lights, unit conversion calcula-

tors, and transformation nodes, among others. Each node has a set of **attributes**, which can be defined as **inputs** or **outputs**, and a **compute** method, which calculates the output attributes based on the inputs. For example, a node defining a sphere may have its radius and center defined as input attributes, and then use them to calculate locations of vertices that compose a spherical mesh, which is the output attribute.

Output attributes of one node can be connected to input attributes of others, provided the attributes are of the same data type. For example, to make a mesh follow a locator's movement, one would connect the locator's "translate" attribute to the mesh's "translate". Nodes can be connected to many others. Internally, a Maya scene is represented as a directed graph of the nodes with attribute connections as the edges. The graph structure of the scene, known as the Dependency Graph or DG, can be visualized in the **Hypergraph** window. Different views of the Hypergraph are shown in Figures 6.6, 6.7, and 6.8.

Attribute values can be manipulated directly using **tools** and MEL **commands**. Commands can be typed into the command line for interactive feedback. For example, the command "move -r 0 1 0 sphere1" moves the sphere1 object one unit in the y direction, relative to the current location. The "-r" is a command **flag** which stands for "relative"; to move the sphere to an absolute location in space, one would replace the "-r" with "-a". Some commands are connected to **tools**, or **contexts** in Maya terminology, which use feedback from the mouse or stylus to manipulate objects. To use the move tool, the user clicks on the object and drags it to a new location in space. Internally, "move -r <mouse release point> sphere1" is executed. Some tools use only the mouse cursor position as input, but others have auxiliary visual representations or handles, called **manipulators**. For

example, the "move" tool manipulator is composed of three arrows representing the x,y, and z axes. Like scene nodes, a manipulator is selectable, however it does not appear in the DG and only exists when the node it is attached to (the sphere in the move tool example) is selected.

The creative user can arbitrarily build connections between nodes, producing an unlimited range of effects. This flexibility, coupled with direct access to the graph structure of a scene (which some 3D software conceals from the user), is what makes Maya the software of choice for many animation studios. The "open architecture" allows developers to build "plugins" that can seamlessly interact with existing Maya objects.

The Maya Plugin API facilitates the creation of custom nodes, commands, tools, and file translators. The objects are implemented by extending existing Maya classes. Because some parts of a Maya file are most easily accessible through the high-level language of MEL, the plugin API allows calls to MEL commands from C++ files. Maya GUIs must be implemented in MEL.

Two excellent resources for those interested in developing Maya plugins and UIs are *Complete Maya Programming: An Extensive Guide to MEL and C++ API* [Gou03] and *Complete Maya Programming, Vol. II: An In-Depth Guide to 3D Fundamentals, Geometry, and Modeling* [Gou05], both by David Gould.

6.1.1 The Model-Sketch Plugin

The Model-Sketch plugin includes custom, nodes, commands, and tools, whose functionality is outlined in Table 6.1, Table 6.2, and Table 6.3 respectively. Additionally, we implemented a *file translator*, derived from MPxFileTranslator, that reads a text file of anthropometric data and translates it into a network of Para-

Table 6.1: Custom Maya Nodes for Model-Sketch.

Node	Description	Parent Class
Landmark	Represents an anthropometric landmark. Can be connected to any number of Parameter nodes. Displays the location as a sphere.	MPxLocatorNode
Parameter	Calculates an anthropometric measurement based on two connected Landmark locations. Three types: Shortest Distance, Axial Distance, Angle of Incline.	MPxNode
ParameterManip	Attached to a Parameter node to visualize it. Manipulators are virtual nodes which only exist when the connected node is selected.	MPxManipContainer
ParametricModel	Connected to all Landmarks and Parameters associated with a set of Anthropometric data.	MPxNode
MirrorConstraint	Takes a source point, mirror location point, and an axis of reflection as input. Outputs the source point's reflected 3D position about the mirror.	MPxNode

meter and Landmark nodes.

6.2 Landmarks and Parametric Visualization

To maximize artistic freedom, our current approach to parametric modeling does not strictly enforce anthropometric parameters, but simply alerts the user if a modeling operation results in a violation of anthropometric proportions. The "alert" is an intuitive visualization of the linear constraints on a Landmark's location based on anthropometric limitations. Knowledge of the limits can be used to guide future drawing by the user. To calculate proportion constraints, we have

Table 6.2: Custom Maya Commands for Model-Sketch.
 These are all derived from MPxCommand.

Command	Function
matchLandmarks	Maps Landmarks to corresponding points on a user-drawn curve.
matchCurves	Maps the control vertices of an input curve to the control vertices of the drawn curve. It rebuilds the drawn curve so it has the same number of control vertices as the input.
node creators*	Each node defined in Table 6.1 has a corresponding command to create it in the DG. The command name is the same as the node name, but starts with a lowercase letter.
epCurve	This command is from Bibliowicz’s implementation [Bib04]. It takes a set of 3D points as input and generates a NURBs curve that passes through them.

Table 6.3: Custom Maya Commands for Model-Sketch.
 These are all derived from MPxCommand.

Tool	Function
modelSketchContext	The fundamental component of our system. This tool provides the "select-and-sketch" functionality. It switches between modes each time the mouse is released. When the mouse is released after a sketching operation, the "matchLandmarks" command is invoked, followed by "matchCurves".
curveOnPolyContext	An auxiliary tool for drawing a NURBS curve directly on the surface of a polygonal mesh by dragging the mouse along the surface.
polyVtxCurveContext	Used to draw a curve through specific vertices on a mesh. The user must select vertices individually and then press enter to create the NURBS curve. This tool was used to draw curves of the GSH.

implemented three types of nodes: Landmark nodes, Parameter nodes to calculate anthropometric measurements, and a Parametric model that holds all the relevant information.

6.2.1 Parameters

A Parameter node holds information for a single anthropometric measurement. Its inputs are:

- Measurement Type. These include Shortest Distance, Axial Distance, and Angle of Incline.
- Mean = μ
- Standard Deviation = Δ_d
- Axis. Used for the Axial Distance and Angle of Incline measurements.
- Landmark Locations. L_1 and L_2 . For Axial Distance measurements, L_1 must be the right of, above, or in front of L_2 , depending on whether the axis is x, y, or z respectively.

Based on these values, the system generates the following outputs:

- Value = V . This is the calculated value of the measurement based on the two landmark locations.
- Minimum Value = $V_{min} = \mu - \Delta_d$. For non-angular measurements, the minimum value is clamped to zero to avoid negative distances.
- Maximum Value = $V_{max} = \mu + \Delta_d$. To keep the minimum from exceeding the maximum, the lower limit for V_{max} is $V_{min} + 0.001$.

- Difference = $\Delta_V = V - \mu$
- Violation. This is the amount by which V exceeds V_{min} or V_{max} for the measurement. It is zero if the constraint is satisfied.
- Landmark Limits (2). For each landmark, a set of linear constraints that will enforce the anthropometric constraint are calculated. During limit calculation for each landmark, it is assumed that the other landmark is fixed in space. For Axial Distance, constraints are only generated for one axis; for Shortest Distance and Angle of Incline, which are nonlinear, linear approximations are used to generate three dimensional limits, with the assumption that a landmark's orientation will not change dramatically between two Model-Sketch operations.

The **Attribute Editor** in Maya, shown in Figure 6.1, can be used to interactively view these calculations as a Landmark is moved, and also to edit attributes, such as mean and standard deviation, that are not dependent on other nodes.

Parameters are internal Dependency Graph nodes; they are not visible during the Model-Sketch process. However, our system provides a way to visualize them. Each type of measurement has a unique visualization:

- **Shortest Distance.** If it is known that a point A is a certain distance d from a point B , but no other information is given about their locations, this means that geometrically, B may lie anywhere on the surface of the sphere centered at A with radius d . Thus, we visualize a shortest distance constraint as two cocentric spheres centered at the L_1 with radii equal to the Maximum Value and Minimum Value for the anthropometric measurements. The volume between the two spheres defines the limits for L_2 . The sphere

Sketch:Eye_Height_L | FarkasNorms_ParametricModel

Parameter: Sketch:Eye_Height_L

Caching

Node State: Normal

ParameterName: Eye_Height_L

Enabled

Landmarks

Landmarks[0]:

Landmarks[1]:

Measurement Type: ShortestDistance

Mean	10.900
Standard Deviation	1.200
Difference	-10.406
Value	0.494
Parameter Min	7.300
Parameter Max	14.500
Violation	-6.806
Axis	NONE
Axis Order	NONE
Flexibility	3.000

Figure 6.1: The Attribute Editor for a Parameter Node. The mean, standard deviation, and two landmark location attributes are used as inputs to calculate anthropometric measurements on the mesh.

that passes through L_2 is also displayed; colored green if the constraints are satisfied, red if not. (Figure 6.2)

- **Axial Distance.** This measurement is drawn in the viewport as a simple line segment along the specified axis, with square-shaped panels located at the maximum and minimum values, and at the location of the second Landmark. (Figure 6.3)
- **Angle of Incline.** The display includes the canonical axis and the ray that originates at L_1 and passes through L_2 , along with two rays demonstrating the maximum and minimum angles. (Figure 6.4)

6.2.2 Landmarks

Landmark nodes are derived from Maya **locator** nodes. The Maya locator node's most significant attribute is its position; it is visible to the animator but not renderable, and can be moved arbitrarily in space by the user. Thus, while we use Landmarks as modeling controls, generic locators are often also used as a rigging handles. By default, Maya locators are displayed as crosshairs. Our plugin overrides the locator's "draw" method to display Landmark nodes as wireframe spheres with a name label that can be toggled on or off.

Each Landmark's position attribute is connected to a number of Parameter nodes, each of which independently determines linear limits for the Landmark. Provided that the predefined Parameter set is valid, the limits produced by all the Parameter nodes connected to a single Landmark will form an overlapping region in space. For each set of limits produced by a Parameter node, a connection is made back to the Landmark node. A set of linear limits in the x, y, and z

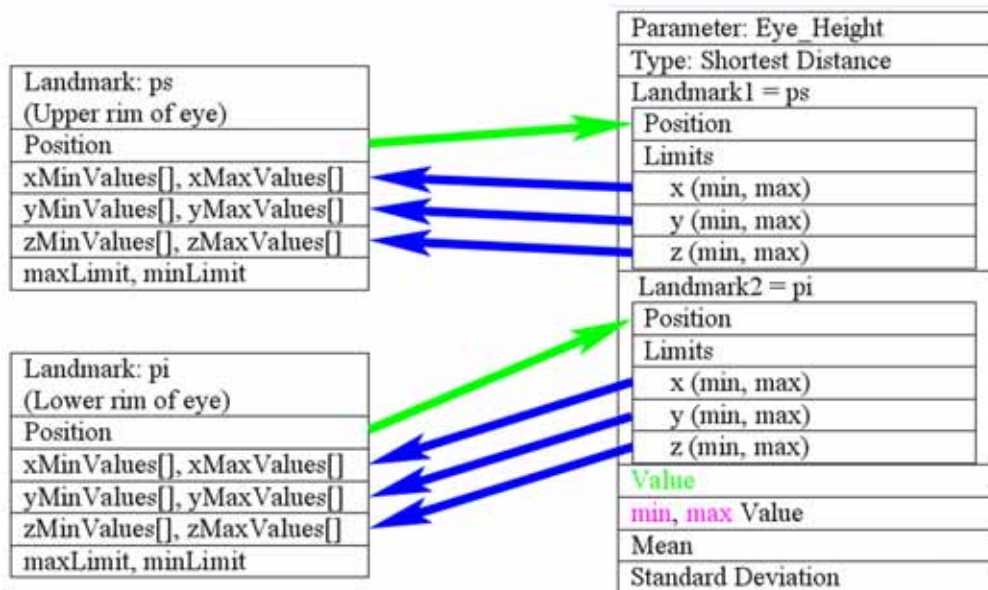
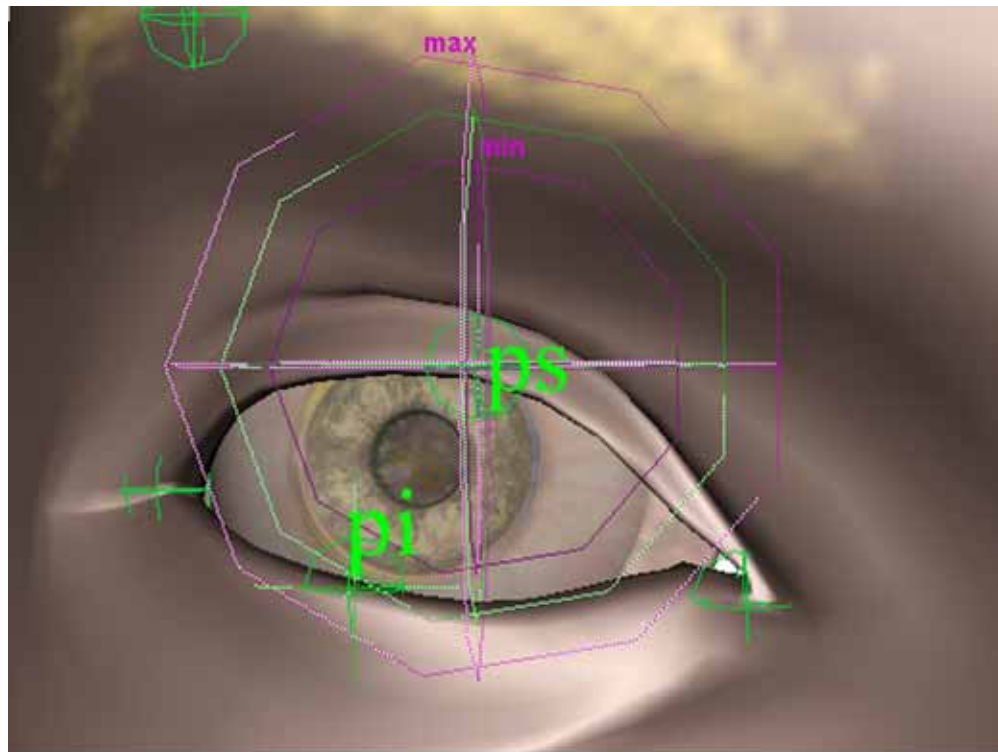


Figure 6.2: The shortest distance measurement is visualized using two cocentric spheres representing the maximum and minimum values of the parameter. The green sphere represents the current value. To ensure that anthropometric constraints are satisfied, "ps" can be placed anywhere outside the minimum sphere and inside the maximum sphere. A simplified schematic diagram of the Parameter and Landmark nodes, along with their connections, is shown below the screenshot.

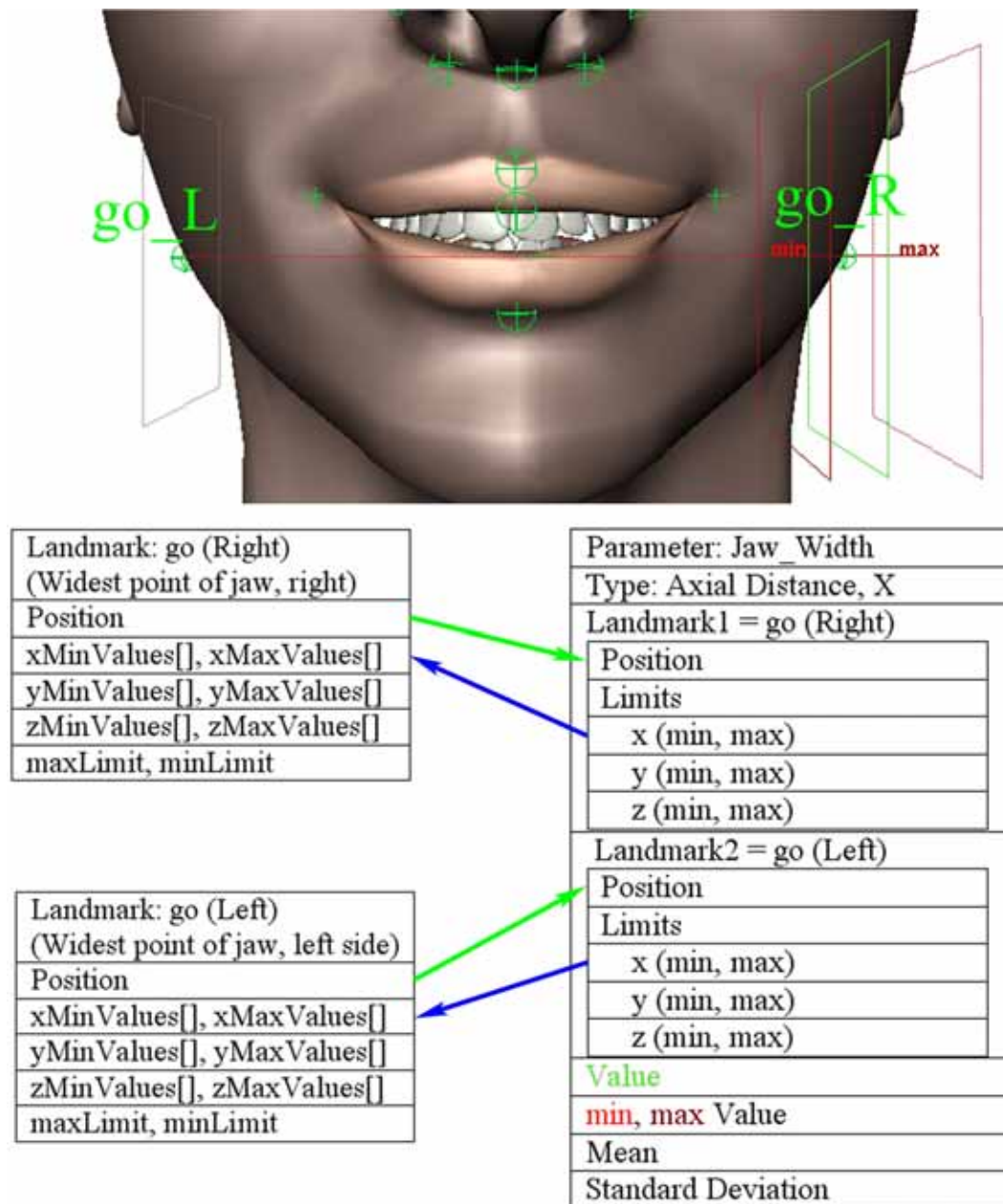


Figure 6.3: The axial distance measurement is visualized with a simple axis-aligned line segment between the two Landmarks. The maximum and minimum values are represented with square-shaped panels. In the schematic diagram, note that only the limit for the specified axis is connected to the Landmark nodes.

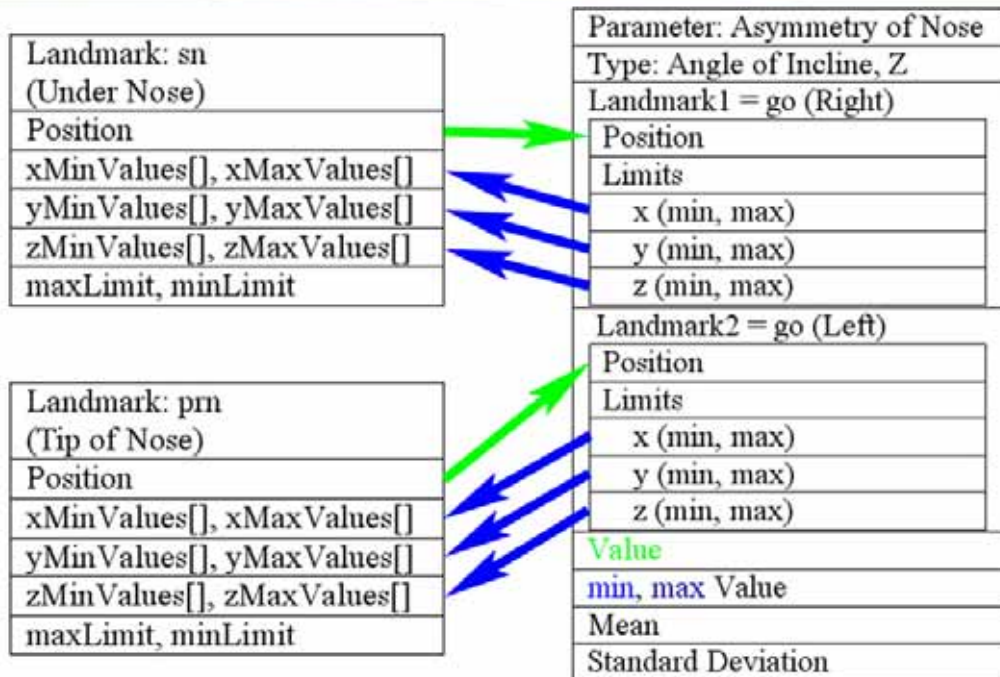
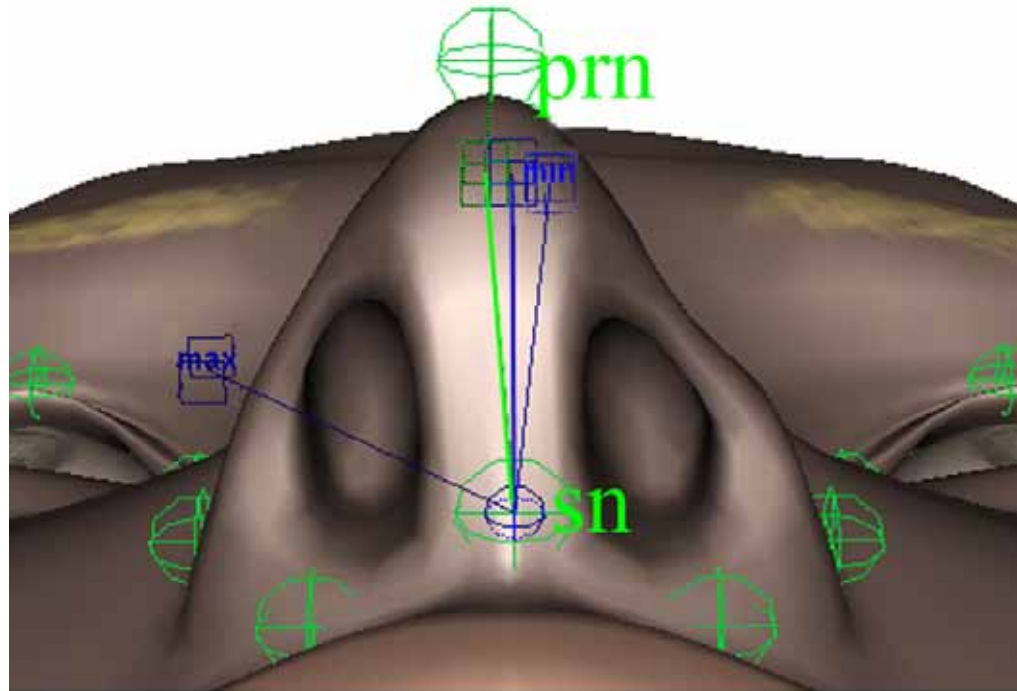


Figure 6.4: The angle of incline measurement is represented with four rays: the canonical axis (the z-axis in this example), the axis-aligned ray between the two landmarks (shown in green), and the two rays demonstrating the maximum and minimum angles. Like the shortest distance measurement, the angle of incline measurement is a nonlinear constraint. Our system uses a three-dimensional linear approximation to the constraint.

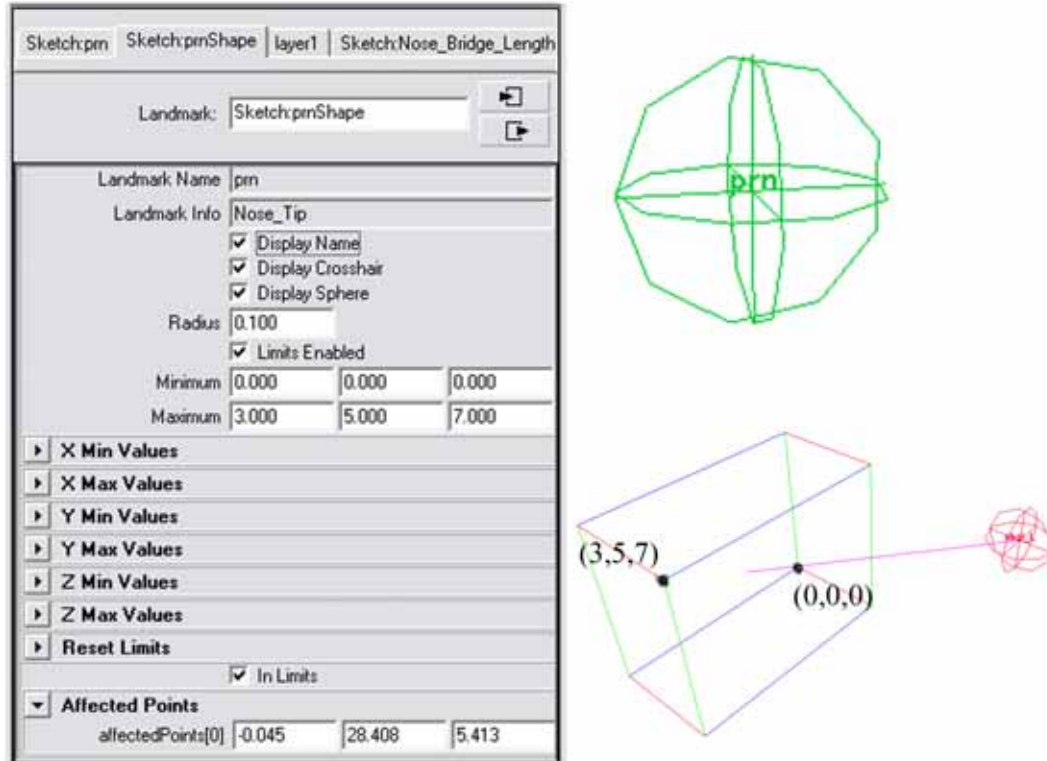


Figure 6.5: The right image shows the same Landmark node; when it satisfies linear constraints, it is green; when it violates them, it is highlighted red and the constraints are displayed in the form of a box. In order to fix the violation, the user may choose to move the Landmark into the box. On the left is the Attribute Editor view of the Landmark.

directions can be visualized as a box. With access to limits generated by all the Parameter nodes it affects, the Landmark calculates the intersection of all these "boxes" to generate its own absolute maximum and minimum attributes. Finally, the Landmark returns a boolean attribute that tells whether or not the current location is within these limits.

The Landmark is displayed in green if the limits are satisfied, red if not. The system provides an extra visual guide for fixing problematic Landmarks. When limits are violated, the limits "box" is displayed with an arrow pointing from Landmark to the box's center, as shown in Figure 6.5.

Landmarks lie on specific points along the "paths" created by the GSH. To

represent these associations, the closest **edit point**¹ on each curve the Landmark crosses is connected to the Landmark's input attributes. A single Landmark can be located at the junction of several GSH curves. The main function of the edit point connection to the Landmark is to indicate its relative location to other Landmarks along a given curve. This becomes useful when constraining selection, as we will discuss in Section 6.3.3.

6.2.3 Parametric Model

A third node, the ParametricModel, provides a hub for all the Landmarks and Parameters associated with a single Parameter set. Connected to Landmark locations and Parameter values, the Parametric Model's main purpose is to provide accessibility to the fairly complicated graph of anthropometric measurements, as shown in Figure 6.8.

6.2.4 Translating Anthropometric Data

Figure 6.8 shows that the parameter graph is quite complex. Our system is equipped with a text file translator that can generate the graph automatically from a user-defined **parameter file**. This file is a spreadsheet that lists details about anthropometric measurements and landmarks, along with pertinent information for automated rigging. The file follows a specific format so it can be read by the translator and successfully translated into a network of Maya nodes.

The first row in the spreadsheet contains the label "ParametricModel" followed by a unique name for the parameter set. The next two lines specify a

¹A parametric t-position that marks the end of a **span** on a NURBs curve. [PT97].

Parameter Node

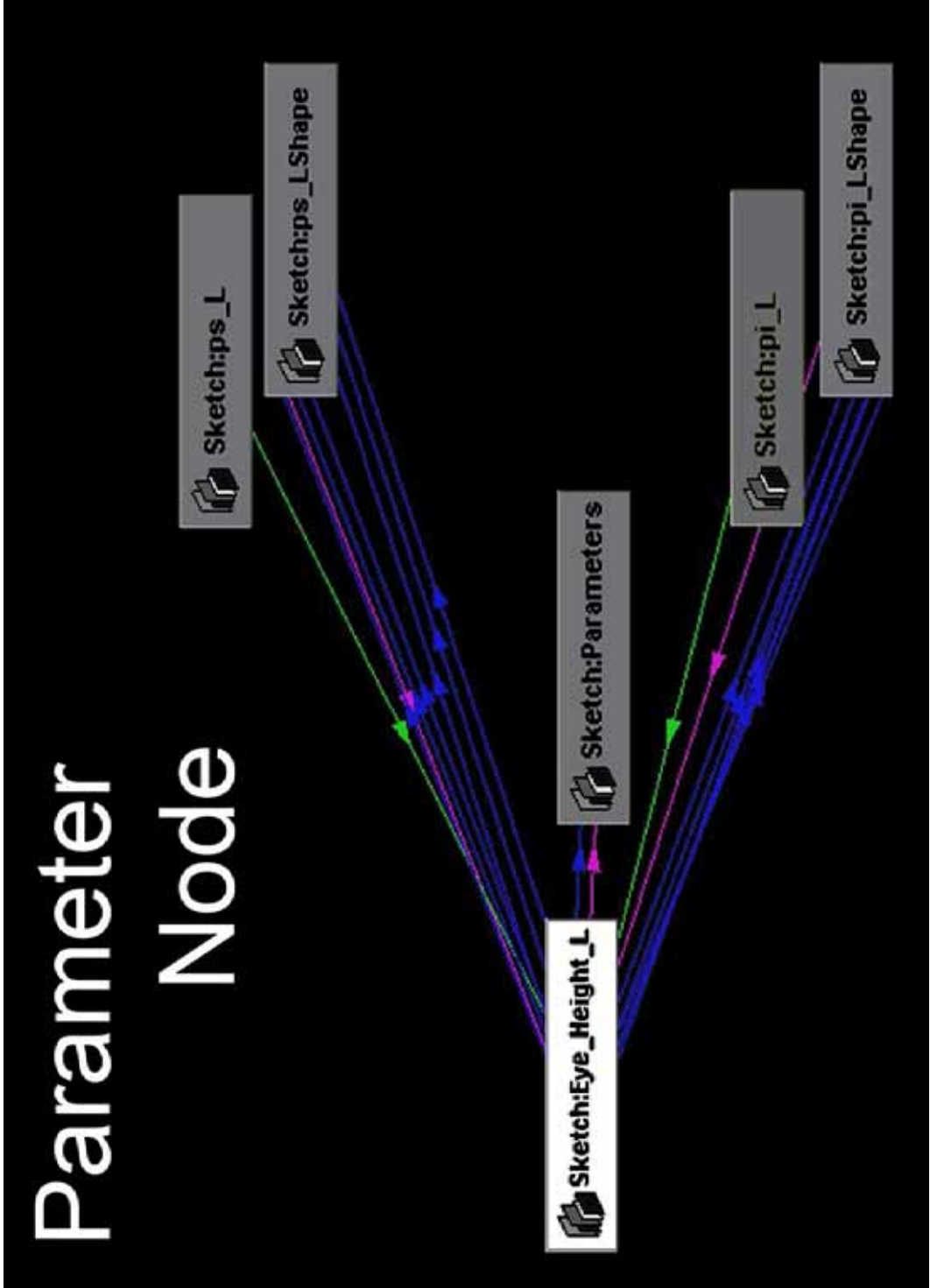


Figure 6.6: Each Parameter node is connected to two Landmark locations.

Landmark Node

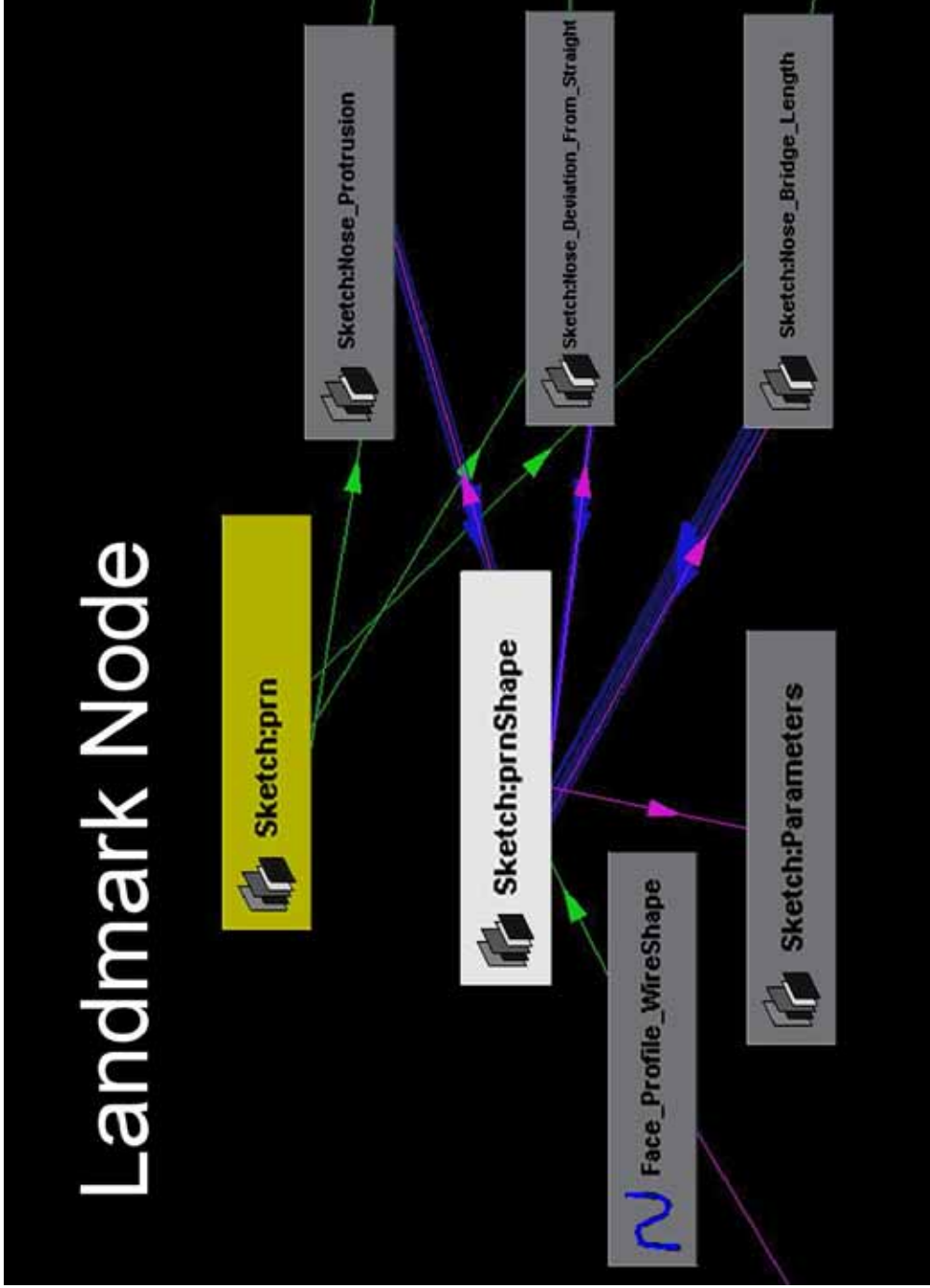


Figure 6.7: A Landmark can be connected to an arbitrary number of Parameter nodes (on the right). It is also connected to one or more GSH curves. In this example the GSH curve is "Face Profile Wire". Note that the Landmark is composed of two nodes: a Transform node (yellow) and a Shape node (white).

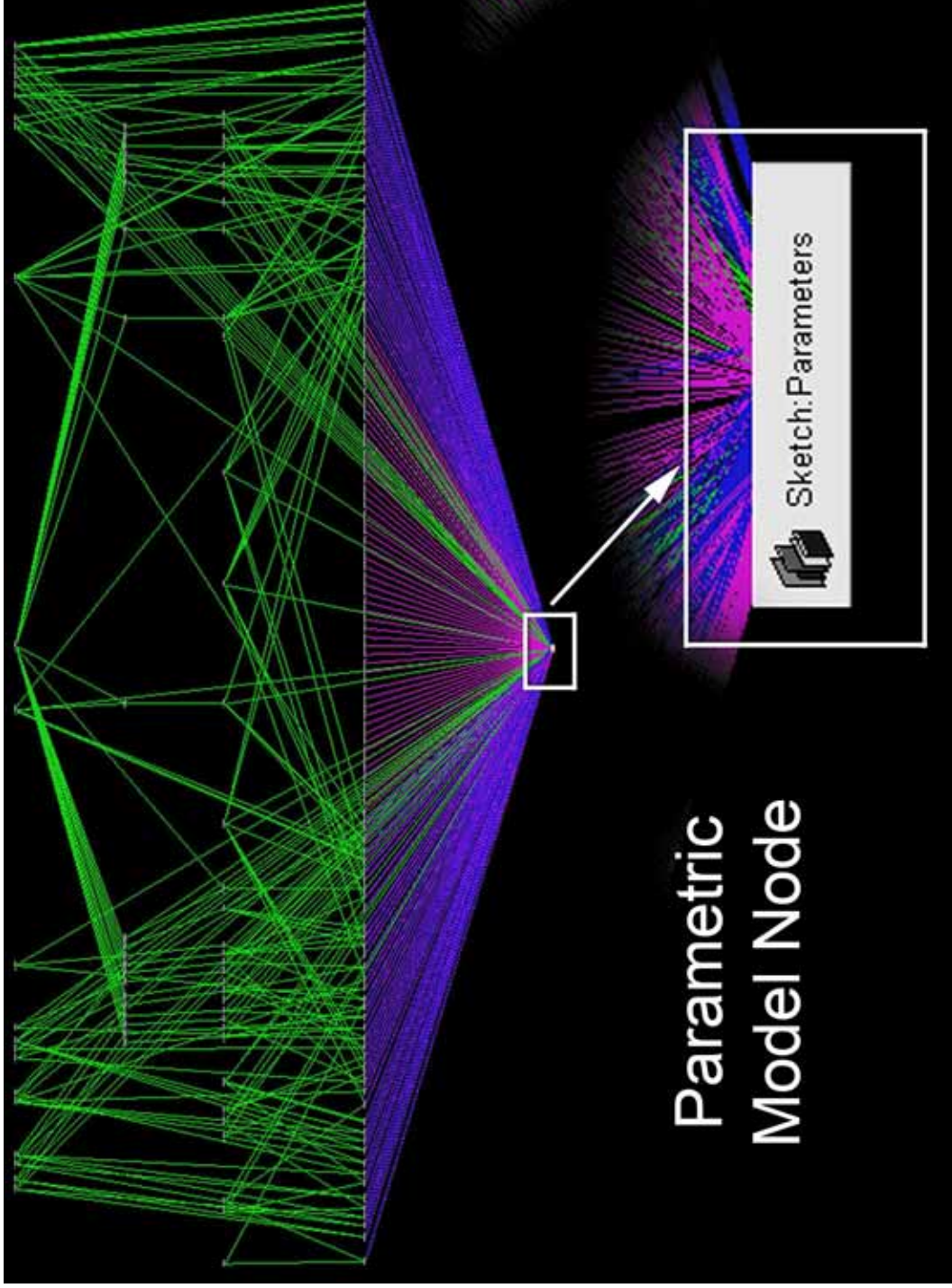


Figure 6.8: The complex network of connections to this ParametricModel node represents a subset of Farkas's measurements. [Far94]

Table 6.4: Format of the Parameters File.

	A	B	C
1	ParametricModel	North_American_Caucasian_Facial_Norms	
2			
3	MirrorMethod		0
4	CalculationMethod	Ratio	
5			
6	Planes		
7	PlaneType	PlaneName	isMirror
8	XY	Frankfort	0
9	YZ	Sagittal	1
10	ZX	Transverse	0

”MirrorMethod” followed by a 0 or 1, which tells whether the model should be symmetric from the negative axis to the positive, or vice versa, and a ”CalculationMethod”, which can be ”Exact” or ”Ratio”. If the calculation method is ”Exact”, the anthropometric measurements are calculated absolutely; ”Ratio” indicates that parameter values should be normalized with respect to a given base parameter. In ”Ratio” mode, the first specified Measurement will be used to normalize all other parameters.

Next, the file contains a list of ”Planes”. Because many anthropometric measurements are taken with the face or body aligned to one of the anatomical planes, a user may wish to constrain landmarks to them; for example, the feet could be constrained to a floor plane. The second column specifies whether the plane should be used as the mirroring plane. Only one of the planes can be used as a mirror. The mirror plane is required, all others are optional. (Table 6.4)

The following section is labeled ”Parts” and hierarchically lists the names of body parts (Table 6.5). These are used to set up a joint hierarchy for rigging and ”parents” for landmarks. We recall that some curves of the GSH are loops around significant joint locations. The naming convention for these curves is <joint name>_Wire. If the file translator finds a corresponding joint wire for a joint listed under ”Parts”, it will create a joint at that location, otherwise it creates the joint

Table 6.5: Joints specified by the Parameter File.

	A	B	C
10	ZX	Transverse	0
11			
12	Parts		
13	Name	Parent	Paired
14	Pelvis	NONE	0
15	Waist	Pelvis	0
16	Chest	Waist	0
17	Neck	Chest	0
18	Head	Neck	0
19	Forehead	Head	0
20	Jaw	Head	0
21	Eye	Head	1
22	Lips	Head	0
23	Nose	Head	0
24	Ear	Head	1
25	Shoulder	Chest	1

at the origin.

For each part, the user must specify the parent part, using "NONE" for the root joint. The file must define the hierarchy starting at the root and working downward. Since the translator creates joints in the order they appear in the file, joints whose parents have not been created yet would not be attached to the skeleton.

The next two sections of the file are responsible for creating the actual Parametric architecture. The "Landmarks" list generates Landmark nodes based on their name, description, parent joint, whether or not they are paired (if a Landmark is paired, two nodes are created with a mirror constraint between them), and an optional "PlaneConstraint", which can be "NONE" or the name of one of the planes defined in the "Planes" section. If the plane exists, a constraint is applied to keep the Landmark on the surface of the plane. (Table 6.6)

Table 6.6: Landmarks specified by the Parameter File.

	A	B	C	D	E
49	Toe4	Ball	1		
50	LittleToe	Ball	1		
51					
52	Landmarks				
53	Name	Description	Paired	ParentPart	PlaneConstraint
54	v	Head_Top	0	Head	Sagittal
55	op	Head_Back	0	Head	NONE
56	eu	Head_Outer_Edge	1	Head	NONE
57	en	Eye_Inner_Corner	1	Eye	NONE
58	ex	Eye_Outer_Corner	1	Eye	NONE
59	pi	Eye_Lower_Rim	1	Eye	NONE
60	ps	Eye_Upper_Rim	1	Eye	NONE
61	or	Eye_Socket_Lower_Rim	1	Eye	Frankfort
62	os	Eye_Brow_Lower_Edge	1	Eye	NONE

Finally, the file includes a list of "Measurements", their names, the two associated Landmarks (which should be defined earlier in the file), "Type" ("NONE" for shortest distance measurements), the axis of measurement², a unit of measurement, and a mean followed by a standard deviation. The last two columns are for a second mean and standard deviation if the measurement is paired; if not, the user must again specify "NONE" for both of these fields. If the measurement is paired, the values for the left side should come first. (Table 6.7)

Our current setup for Frankie uses Farkas's facial anthropometric data for North American Caucasian females [Far94]. Measurements for the body were derived from artistic proportions. In art, the body is often measured in "heads". Referring to Farkas for the average height of a head, we interpolated a set of "average" measurements.

Because the system does not strictly enforce anthropometric limits, the fact that we used measurements for a specific gender and race will not prevent the user

²For shortest distance measurements, since some of them are almost axial, one must specify the axes in order of their importance. For example, "ZYX" would be specified for the depth of the jaw, which varies most prominently in the z direction and very little in the x direction.

Table 6.7: Measurements specified by the Parameter File.

	A	B	C	D	E	F	G	H	I	J	K
121	mpft	Knee_f	1	Knee	NONE						
122											
123	Measurements										
124	Name	Paired	LM1	LM2	Type	AxisInfo	Unit	Mean1	Dev1	Mean2	Dev2
125	Head_Height	0	v	gn	AxialDistance	Y	mm	215	7.9	NONE	NONE
126	Head_Depth	0	g	op	AxialDistance	Z	mm	186.8	6.8	NONE	NONE
127	Head_Width	0	eu	eu	AxialDistance	X	mm	144.1	5.1	NONE	NONE
128											
129	Hairline_to_Top_of_Head	0	v	tr	AxialDistance	Y	mm	47.4	8	NONE	NONE
130											
131	Forehead_Height	0	tr	g	AxialDistance	Y	mm	52.7	6	NONE	NONE
132	Forehead_Width	0	ft	ft	AxialDistance	X	mm	111.5	4.4	NONE	NONE
133	Forehead_Incline	0	tr	g	AngleIncline	Y	deg	-5.9	5.2	NONE	NONE
134											
135	Jaw_Width	0	go	go	AxialDistance	X	mm	94.5	5	NONE	NONE
136	Jaw_Depth	1	gn	go	ShortestDistance	ZYX	mm	81.8	3.7	81.3	3.7
137	Jaw_Height	1	t	go	AxialDistance	Y	mm	62.2	4.7	61.6	5
138											
139	Chin_Height	0	sl	gn	AxialDistance	Y	mm	27	2.5	NONE	NONE
140	Chin_Incline	0	sl	pg	AngleIncline	Y	deg	9.1	9.6	NONE	NONE

from creating others, but merely provide alerts when the model starts to appear less "Caucasian" and "female." We hope to include parameter sets for males and various races in future versions of the system, along with anthropometric data for the body.

6.3 Sketch-Based Modeling

The interactive select-and-sketch interface is simple to use. The artist draws a curve through some landmarks, draws a second "modeling" curve, and then sees the model fitting to the curve in real time. We have implemented a single tool that switches modes each time the user lifts the pen so that the user does not need to switch tools after every operation. To deliver interactive results, we designed an efficient curve fitting algorithm that is a generalization of the error minimizing and multiresolution heuristics used by Allen and Curless for fitting a prototype model to scans.[ACP03]

After recording the two interactively input curves, the fitting algorithm works in two steps which are illustrated in Figure 6.9:

1. Moving Landmarks to corresponding points on the sketch.

2. Reshaping the GSH curve between the landmarks to conform to the sketch.

We recall that a 3D object's structure is best understood in canonical views. Thus, to deliver predictable results, each "sketchy modeling" operation is two dimensional. That is, one canonical coordinate is held constant when landmarks or curve control points are moved.

6.3.1 Moving Landmarks

A selected set of landmarks defines a GSH silhouette on the model. For clarity, we will refer to this as the "silhouette" and refer to the user input curve as the "sketch".

Determining the new locations of selected landmarks requires a heuristic for identifying "landmark points" on the sketch. Farkas's definition of landmarks states that they are the most distinctive points of the face, analogous to the local minima and maxima of facial feature outlines in one of the canonical views [Far94]. Thus, in most cases one can assume that the local maxima and minima of the sketch correspond to potential landmark locations.

First, we iterate through the points of the sketch curve to build an ordered list of potential "landmarks". The start and end points of the curve are added by default. All points of maximum **curvature** are also added to the list.

L is the set of selected landmarks, and P_L is the list of potential landmarks. To achieve a one-to-one mapping from L to P_L , our system, like Allen and Curless, uses a summation of error criteria to find the the best fit. The criteria are:

1. Distance.

We calculate the distances from a given point p in L to all points in P_L . To

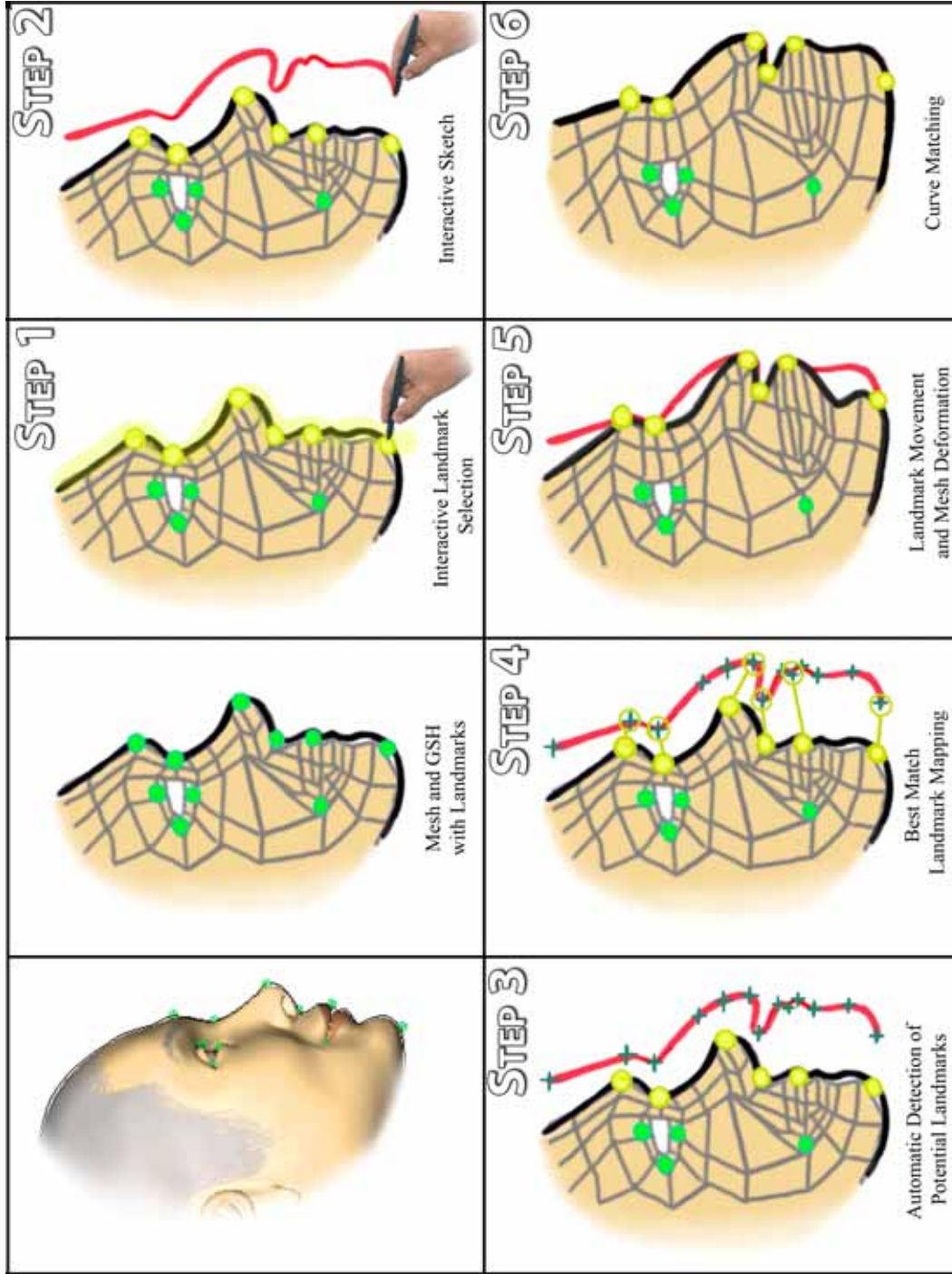


Figure 6.9: Our curve fitting algorithm. Landmarks are moved first, followed by the control vertices of the corresponding curves to achieve a smooth fit.

ensure that the distance error would not obliterate other error criteria, we normalized these distances with respect to the point in P_L farthest from p .

2. Extrema Type and Axis.

There are four types of extrema: start points, end points, local maxima and minima. Local maxima and minima are simply all the points of maximum curvature along the curve. Each maximum and minimum is associated with an axis, which is the canonical axis closest to the point's normal. We project both the highlight curve and the sketch onto one of the standard 2D planes before calling the curve fitting commands to ensure that one axis will be held constant.

Each error criteria is weighted; in our implementation the distance term is considered most important. We iterate through L , calculating the error for each currently available "landmark" in P_L . Since we traverse L in order, eliminating points in P_L if they have already been mapped to a Landmark on the silhouette, we prevent two Landmarks from being mapped to the same point. This, and the requirement that the endpoints are mapped to endpoints, gives us a good initial approximation to the sketch in most cases.

Once the new landmark locations are calculated, the Landmarks are moved in two dimensions to these locations, keeping the third coordinate constant.

6.3.2 Curve Fitting

Two curves can easily be matched if they have identical topologies: The control points can simply be aligned. However, since the silhouette is part of a larger curve on the GSH, and the sketch can have an arbitrary number of control points,

our implementation performs some piecewise curve editing operations to match their topologies. Because we cannot modify the topology of the silhouette, which is a wire deformer on the mesh, we reconstruct the sketch curve. Maya provides a "match topology" function that works by holding one curve constant and adding control points uniformly along the second curve if it initially had fewer points than the source, or by removing the control points with the least effect on the shape if the second curve has fewer points.

Simply executing "match topology" may not produce desirable results, as the new configuration of control points may not necessarily identify landmarks with the same control point indices as the source curve. To minimize this problem, we consider each region between two landmarks individually. The sketch curve is split into several curve segments along the points identified in the landmark mapping step, and "Match topology" is invoked for each curve segment. The same operation is performed for an isolated duplicate of the silhouette curve. The pieces of the sketch curve are then reattached, and then the "match topology" operation is invoked one more time using the original silhouette as the source. This produces a more suitable curve for the final operation, which directly moves each control point on the selected GSH silhouette to corresponding control points on the reconstructed sketch curve.

As a result, the model deforms to fit the user's sketch, completing the operation.

6.3.3 Landmark Selection Constraints

One problem with the selection step, which is common in many 3D modeling applications, is the occlusion problem. Undesired Landmarks, such as those behind the model in the current view, can accidentally be selected. The user may not

notice until after the shaping step, which will bring these points to the front, producing unanticipated and sometimes ghastly results. Occlusion culling, which removes occluded objects from view, is implemented in some modeling programs. However, in Maya it is only a viewing parameter; occluded objects become invisible but they can still be selected.

Another, more subtle problem is that since the fitting algorithm first moves Landmarks and then repositions curve points independently, Landmarks might no longer be aligned with their associated curve points after the curve is reshaped. For example, suppose a user selects the Landmarks which represent the top and bottom of the nose, ignoring the Landmark at the tip of the nose. Not knowing that a Landmark exists between the two selected, the curve fitting algorithm will reposition the two Landmarks and then reshape the curve between them. The reshaped mesh appears correct to the user, however, the nose tip landmark has been left behind. The problem is shown in Figure 6.10. The next time the user wants to reshape the nose, the curve fit will not produce intuitive results due to the offset of the Landmark. Moreover, the Parameter values dependent on the skipped Landmark will no longer be accurate.

Thus, we enforce some constraints on Landmark selection to ensure that in-between landmarks are not ignored. Recalling that the GSH encompasses the most recognizable (and often drawn) "paths" along the face and body, we know that each Landmark lies on one or more of these "paths". Technically, each Landmark is linked to one t-position on each NURBS curve it affects. We have built our model such that each Landmark appears on at least one and no more than four curves.

Restrictions on selection are performed in three steps: The first step is inter-

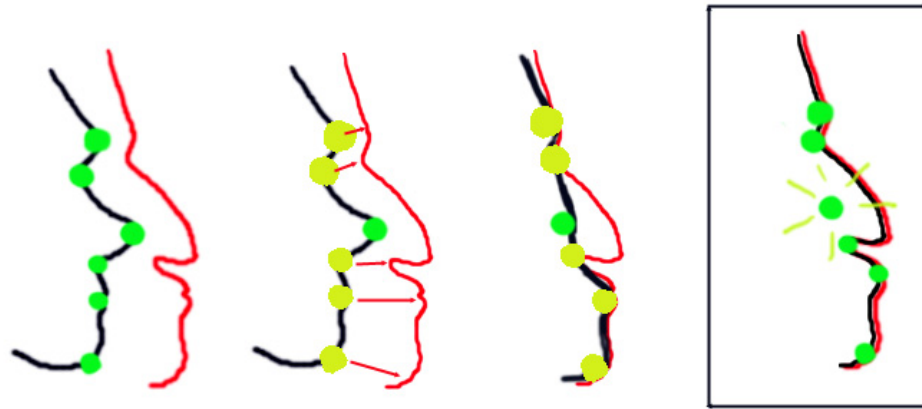


Figure 6.10: When a user skips a landmark on a curve, the resulting mesh will look correct, but the skipped landmark no longer identifies an anthropometric position on the mesh.

active; as the user draws, only currently selectable Landmarks are added to the selection list (and highlighted in yellow); the system determines on the fly which landmarks are "selectable". The next step sorts the selected landmarks in the order of increasing t position on a curve that they have in common. The final step automatically selects and highlights any "in between" Landmarks which were forgotten by the user.

Interactive selection constraints are applied as follows:

1. The first Landmark on the highlight curve is always added to the selection list.
2. All the curves connected to the first Landmark are added to a "potential curves" list.
3. If there is only one curve in the "potential curves" list, the curve becomes the "designated curve"
4. For each subsequent Landmark highlighted as the user drags the stylus,

the system checks whether it is connected to one or more of the "potential curves". If so, the Landmark is added to the selection list, and the "potential curves" list is narrowed. (usually, it takes only two landmarks to designate a curve).

5. When the pen is lifted, the selection is modified.

The modification step "cleans up" the user's selection to form a selection list acceptable by the curve fitting algorithm:

1. If there is more than one "potential curve", it is necessary to designate the first curve. In our model, this step is unnecessary because no two landmarks are connected by the same two curves.
2. Get the t-positions along the designated curve of the selected landmarks, and sort the landmarks in order of increasing t.
3. Iterate through the curve's edit points between the maximum and minimum t positions of selected Landmarks, checking if the edit points are connected to unselected Landmarks. If so, add these Landmarks to the selection list.

By restricting landmark selection to predefined paths along the model, we prevent selection of occluded objects, provide the curve fitting algorithm with inputs that will not cause unpredictable results, and ensure that Landmarks remain aligned with the points they affect. Consequently, the user can select a range of Landmarks by only selecting the two endpoints of the range.

6.4 Summary

By fusing an artistic tool with mechanisms for interactive analysis of the "drawing"'s proportions, we have developed a novel system that provides accurate parametric modeling functionality without stringent constraints. Our approach differs from previous approaches to both parametric modeling and sketch-based interfaces in its unique combination of internal constraints that keep track of anatomical relationships for the user and minimize the instability of sketch-based editing; "constraints" which paradoxically facilitate and encourage free creative expression.

Chapter 7

Changing Faces

This chapter presents the compelling results produced using our Model-Sketch interface. We measure success in terms of both the time it takes to produce a new, animatable model, and the quality of individual models. While quality is a subjective measurement, we illustrate the versatility of the process by defining a range of variation for human facial shapes based on anthropometric and common facial description parameters, and then testing the interface to ensure that all the minimum desired shapes can be reproduced. Given that a certain range of variation is achievable, it then becomes the artist's responsibility to mentally piece them together and "sketch" their interpretations in 3D to create unique models.

7.1 Visualizing Contours

Recall that the GSH is the set of "gesture sketch" curves that conform to the most recognizable silhouettes of the model. Subsets of GSH curves represent individual features, as shown in Figure 7.1. Because the GSH is updated interactively, it functions as a "line drawing" representation of any new face created (Figure 7.2).

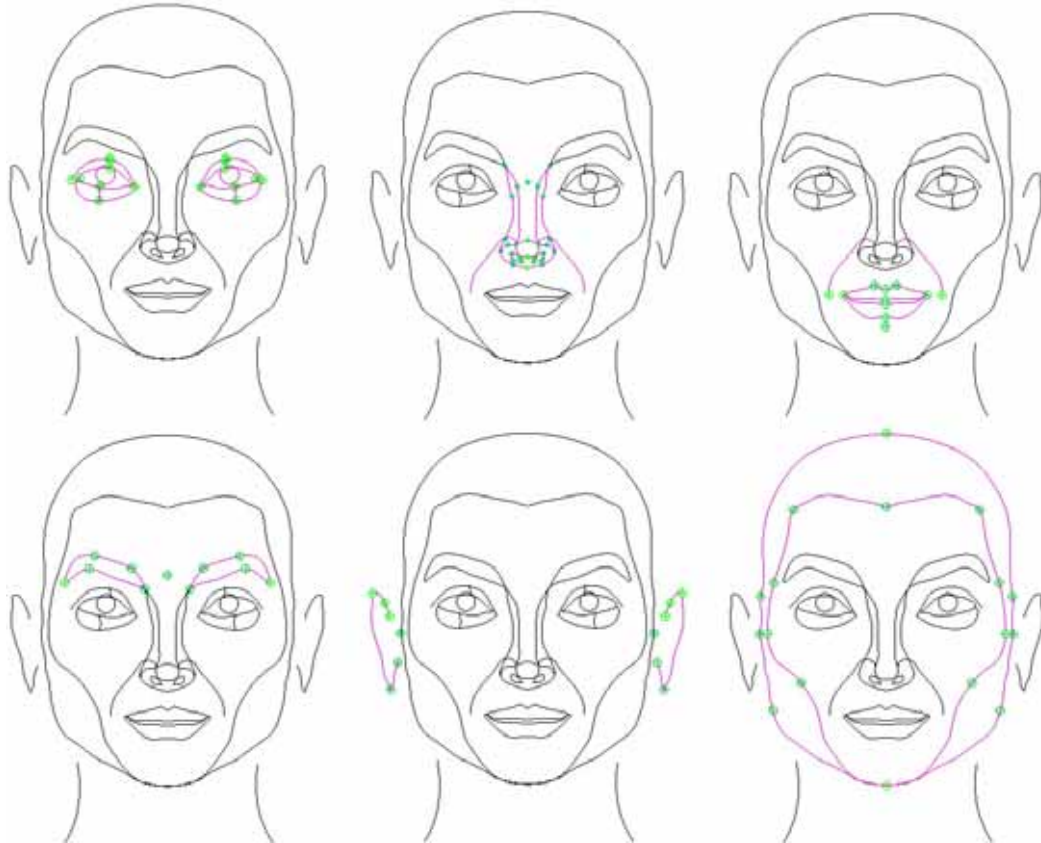


Figure 7.1: The landmarks and GSH curves corresponding to specific facial features. For efficient fitting using the Model-Sketch interface, Landmark selection is limited to sets that are consecutive along a single shared GSH curve.

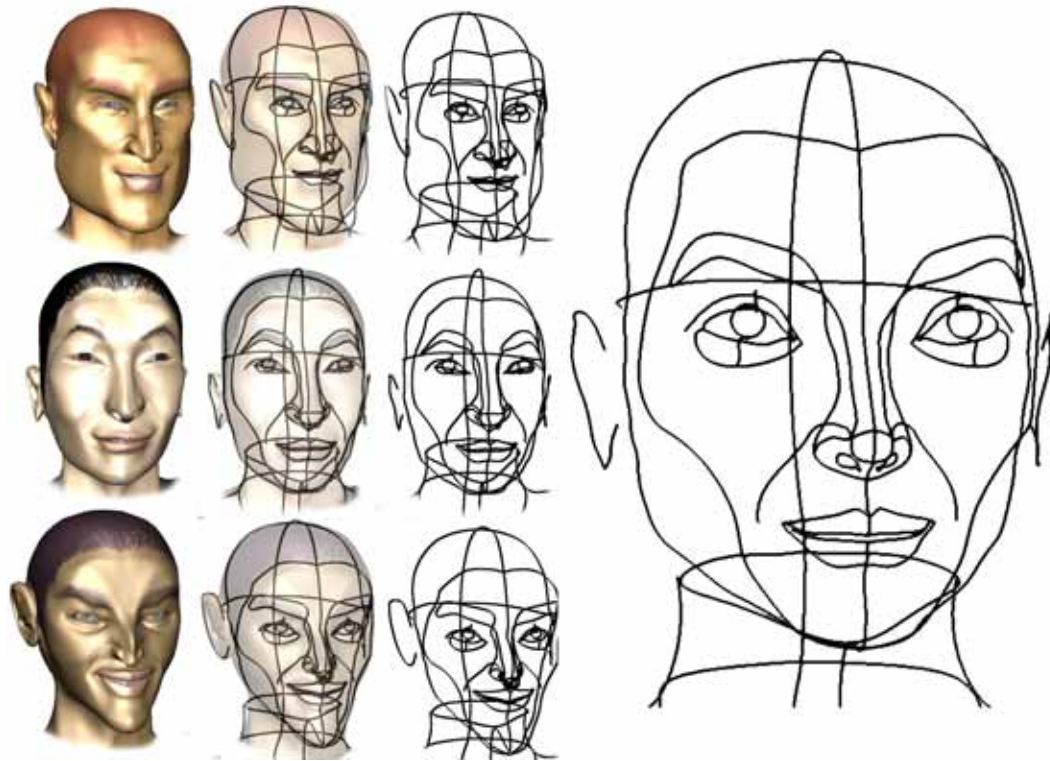


Figure 7.2: The GSH is a useful visualization because it captures the major descriptive silhouettes of the character. On the left are some characters created using the system shown next to their GSH curves. Frankie’s GSH is shown on the right for comparison.

This allows the user to view contours that may be occluded by the model, such as the crease of the eyelid, and also to see the relative locations of Landmarks along the GSH curves. The artist should be somewhat familiar with the layout of the GSH-to-Landmark correspondences, because, as described in Section 6.3.3, a set of selected Landmarks must be connected by a single shared GSH curve.

7.2 Procedure

In Figure 6.9, we illustrated our algorithm for fitting the prototype mesh to a user-drawn curve. Figures 7.3 and 7.4 demonstrate results of the fitting algorithm applied to the actual prototype model.

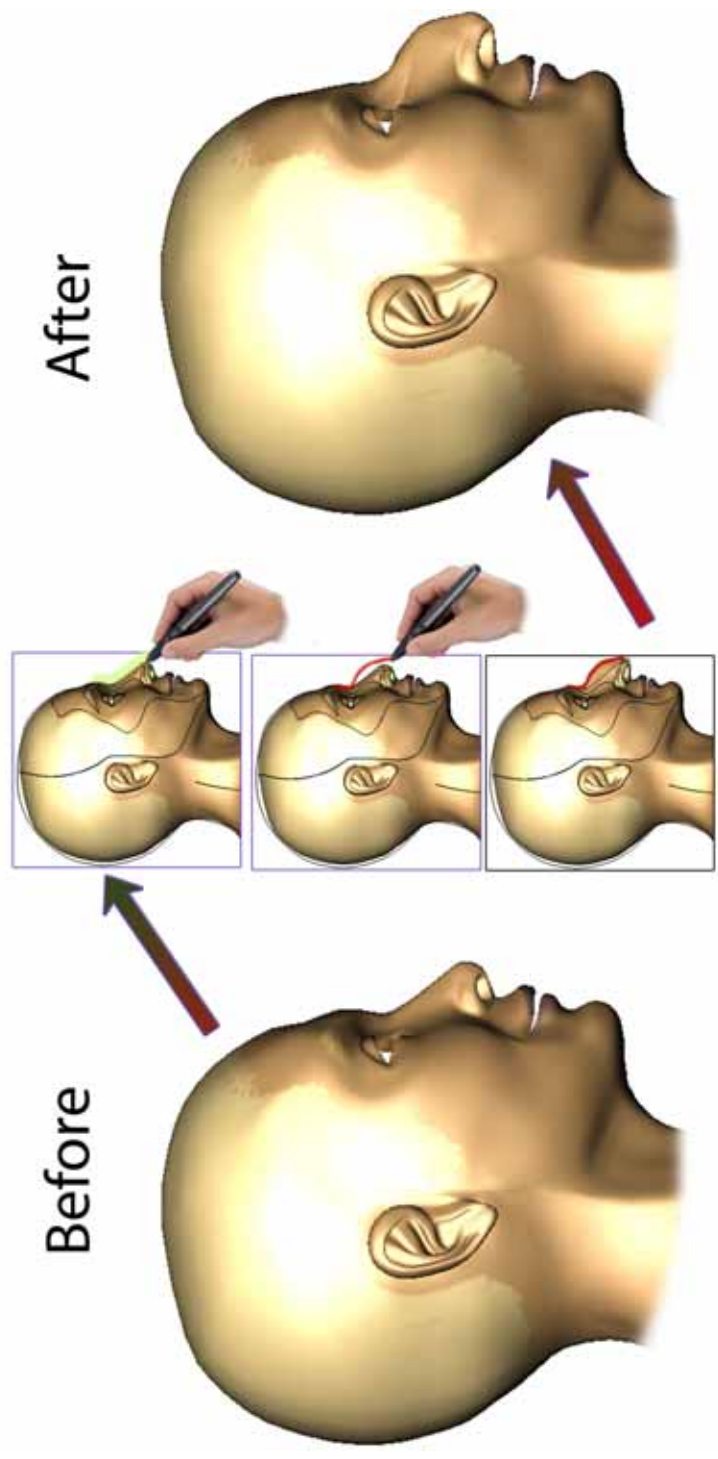


Figure 7.3: Demonstration of the usage of the Model-Sketch tool. The artist performs a simple select-and-sketch operation to reshape the nose interactively.

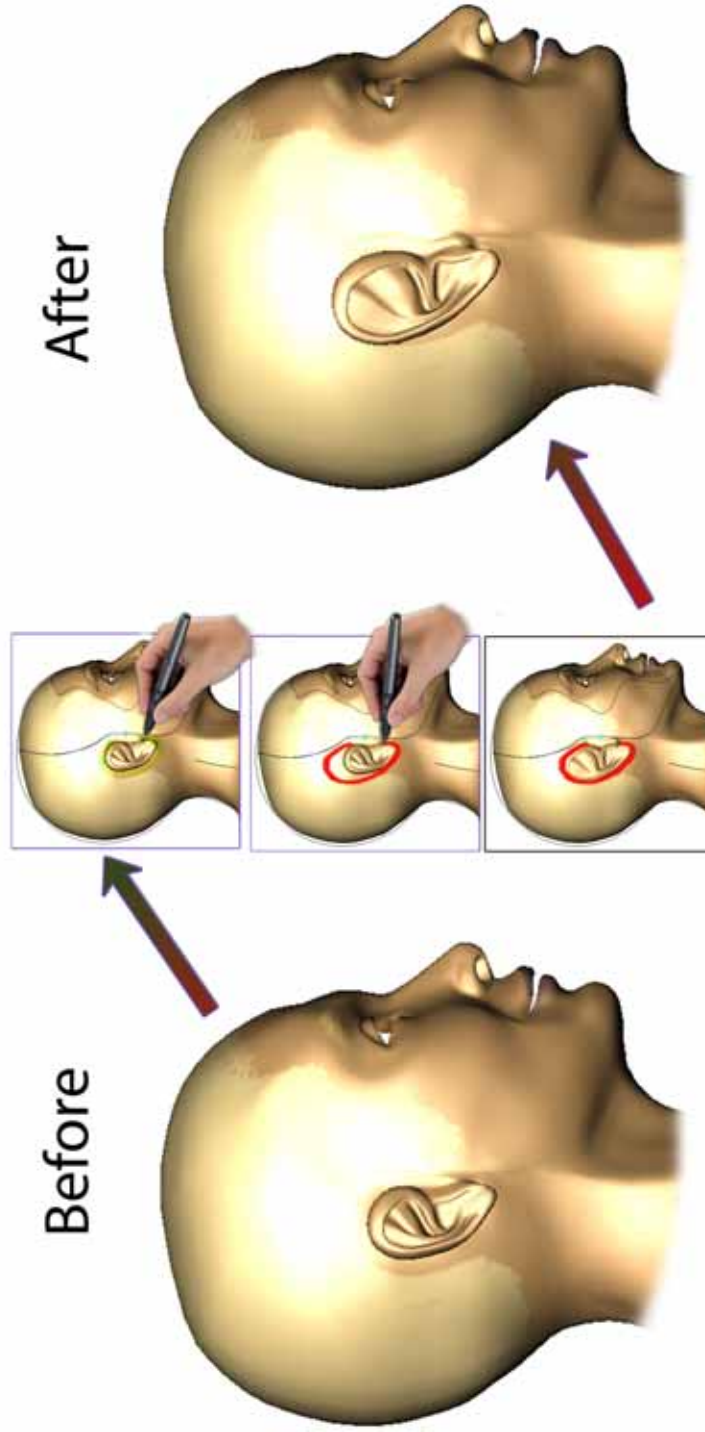


Figure 7.4: Another example of a Model-Sketch operation (see Figure 7.3, used to reshape the ear).

Preliminary trials of the interface revealed that mapping the entire selected region to a new location at once can be visually jarring, and not always intuitive. To minimize this, we added the constraint that the first selected Landmark acts as an "anchor". When two or more Landmarks are selected, the sketch curve automatically starts at the location of the first or last Landmark, whichever is closest to the pen. When only one Landmark is selected, no curve fitting is calculated and the Landmark is simply mapped to the center of the sketch.

7.3 Facial Features

To test the versatility of our system, we have decomposed the facial features in terms of commonly recognized variations. The classifications of facial features are based partially on anthropometric parameters, but also on "shape" parameters, which have no scientific basis but are useful in describing a face. Shape parameters are often used in cosmetic applications, for example, choosing haircuts to flatter one's face shape. The anthropometric parameters we have tested are those that have counterparts in the colloquial vocabulary of facial description; for example a person may have "wide-set" or "close-set" eyes depending on the anthropometric measurement of the space between the eyes.

7.3.1 Eyes

The eyes are the most perceptually important part of the face. Geometrically, they are also the most complex; the overlapping geometry of the eyelid, along with the fact that changes to the eyelid must interact properly with the eyeball and vice-versa, require special considerations to fit into the Model-Sketch framework without requiring a switch to a different tool. Thus, we have tailored a set of

unique constraints for the eyes.

Our efficient eyeball setup conforms the eyelids on the surface of the eye without requiring any extra deformers on the mesh itself. To represent the thickness of the eyelid, we have encapsulated the eyeball geometry inside an invisible "dummy" sphere of a slightly larger radius. The eyeball is parented to the dummy sphere. The eye fissure landmarks, excluding the inner corner of the eye, and the eyelid crease landmark are geometrically constrained to conform to the surface of the dummy sphere. The sphere also acts as a sculpt deformer on the the two GSH silhouettes of the eye fissure and eyelid crease, ensuring that control vertices remain outside the sphere. Finally, the dummy sphere and all the eye landmarks were parented to the "en" Landmark, which denotes the inner corner of the eye.

Thus, by using a single Model-Sketch stroke to move the "en" Landmark, the entire eye including the eyeball geometry can be repositioned. The effect of simply repositioning the eye is demonstrated in Figures 7.1 through 7.3. Also, scaling the "dummy" eye rescales not only the eyeball, but also deforms the eyelid geometry to conform to it. Eyeball scaling is a rare operation and the only one which requires the use of a tool other than the Model-Sketch tool. For a realistic adult character, the eyeballs need not (and should not) be scaled; the operation is useful in modeling the proportionately large eyes of babies or anime characters.

Eye shapes are often classified by eye cosmetics or eyeglass specialists. The common eye types are illustrated in Table 7.4. Note that the single-eyelid is a special case where the Frankie model actually has more geometry than is needed to produce the feature. To create the appearance of a single eyelid, the eyelid crease landmark can be aligned with or constrained to the upper eye fissure landmark to hide the double lid.

Table 7.1: The space between Frankie's eyes can be increased or decreased by using one stroke of the Model-Sketch tool. The landmark on the inner corner of the eye is used to change the global position.

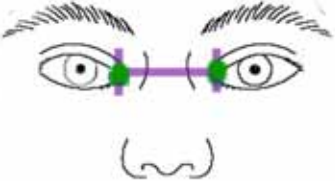





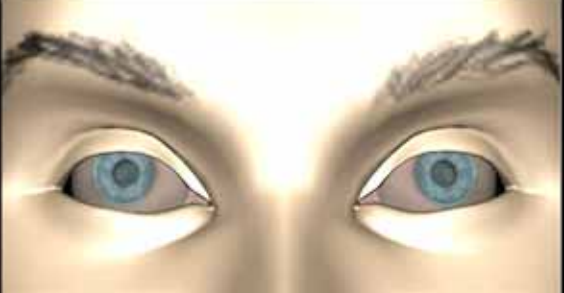
<h1>Eye Spacing</h1> 		
Landmarks: en(Left)-en(Right)		
		Maximum = $3.18 + .23 =$ 3.41 cm
		Mean = 3.18 cm Standard Deviation = .23 cm
		Minimum = $3.18 - .23 =$ 2.05 cm

Table 7.2: The maximum and minimum anthropometric heights of the eyes.

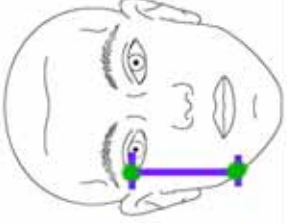



<h1>Eye Height</h1>			
<p>Landmarks: ex-go</p>			
<p>Maximum= 9.3+.45= 9.75 cm</p>	<p>Mean= 9.3 cm Standard Deviation= .45 cm</p>	<p>Minimum= 9.3-.45= 8.85 cm</p>	
			

Table 7.3: The maximum and minimum anthropometric depths of the eye.

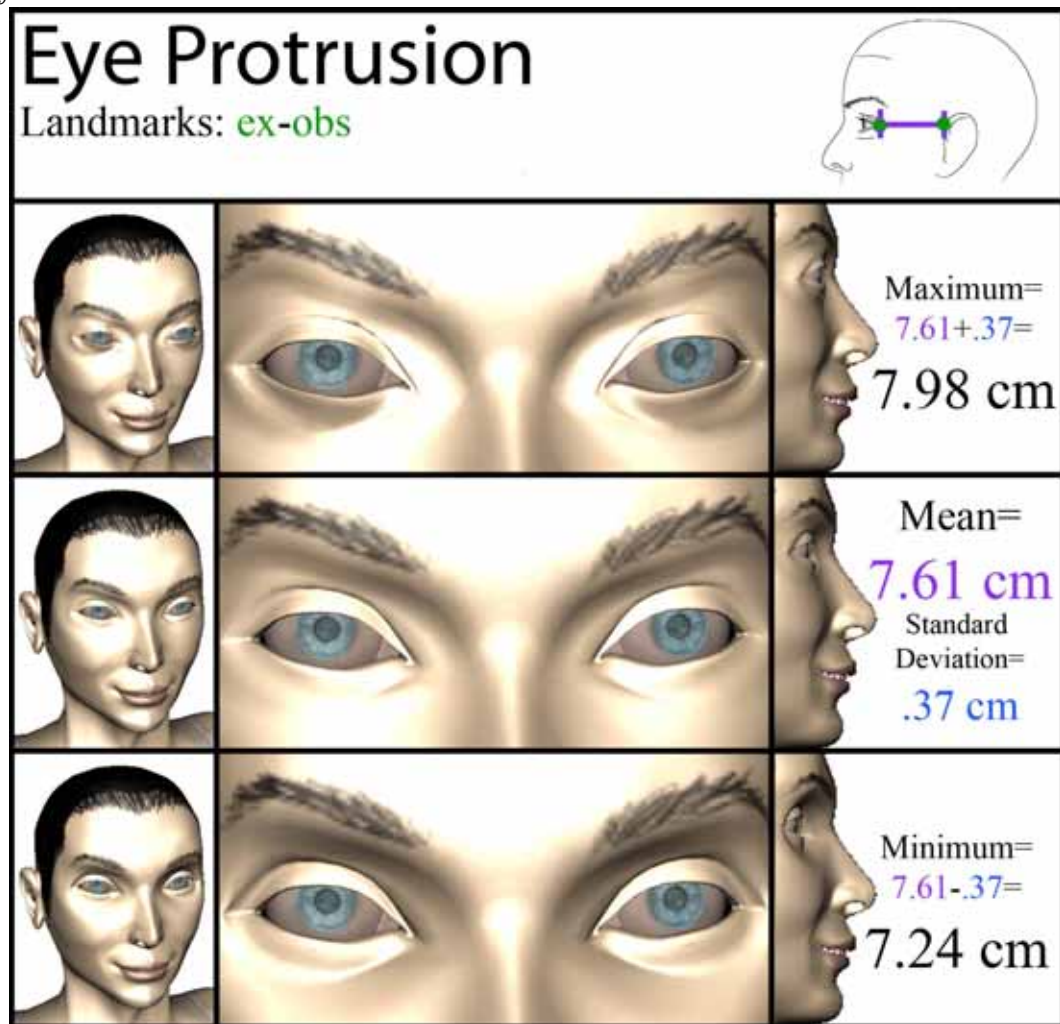

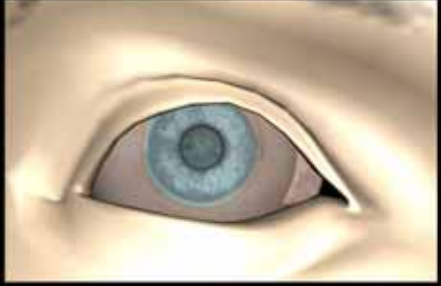

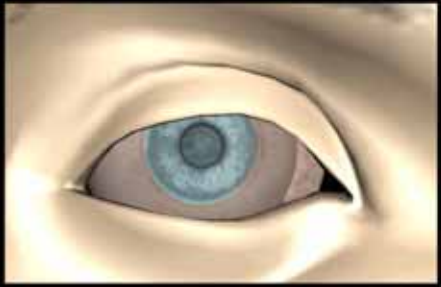

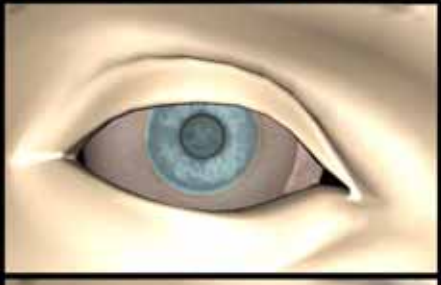

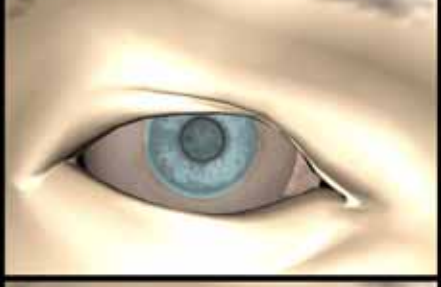

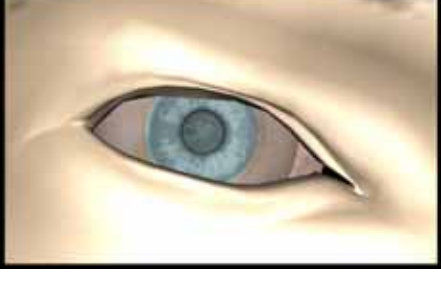


Table 7.4: Frankie's eyes have successfully been molded into the five major eye shape classes.

Eye Shapes		
		Round
		Heavy Eyelids
		Almond
		Hooded
		Single Eyelid

7.3.2 Nose

Nose shapes are often classified by plastic surgeons for rhinoplasty. The components of the nose, the bridge, base, tip, and nostrils, work in conjunction with each other to describe the overall appearance. The nose bridge shape is best understood in the profile view; the width of the bridge and base can be altered in the front view. The tip and nostrils are most easily manipulated in the bottom view. Classified nose profile shapes are illustrated in Table 7.5, and common tip shapes are shown in Table 7.6. Any of the profile shapes can be combined with any of the tip shapes.

7.3.3 Lips

Traits of the lips are generally described in binary terms. For example, they can be full or thin, and have a cupid's bow or lack one. Table 7.7 demonstrates these traits. Here, we have described only *conformation* parameters of lip shape; expression parameters can be changed as well. For animation purposes, we have constrained the corners of the nasolabial crease GSH curve to the Landmarks at the corners of the mouth so that the facial muscles will crease appropriately when the character smiles or frowns. Landmark positions can be keyframed; thus, our setup functions not only as a modeling harness, *but also a built-in facial rig*.

7.3.4 Face

The face shape encompasses the shapes of the jawline, forehead, and cheekbones. Each of these features can be manipulated locally as well. Generally the face shape can be changed with just two Model-Sketch strokes, one for the jawline and one

Table 7.5: The nose profile can generally be molded with a single Model-Sketch operation.

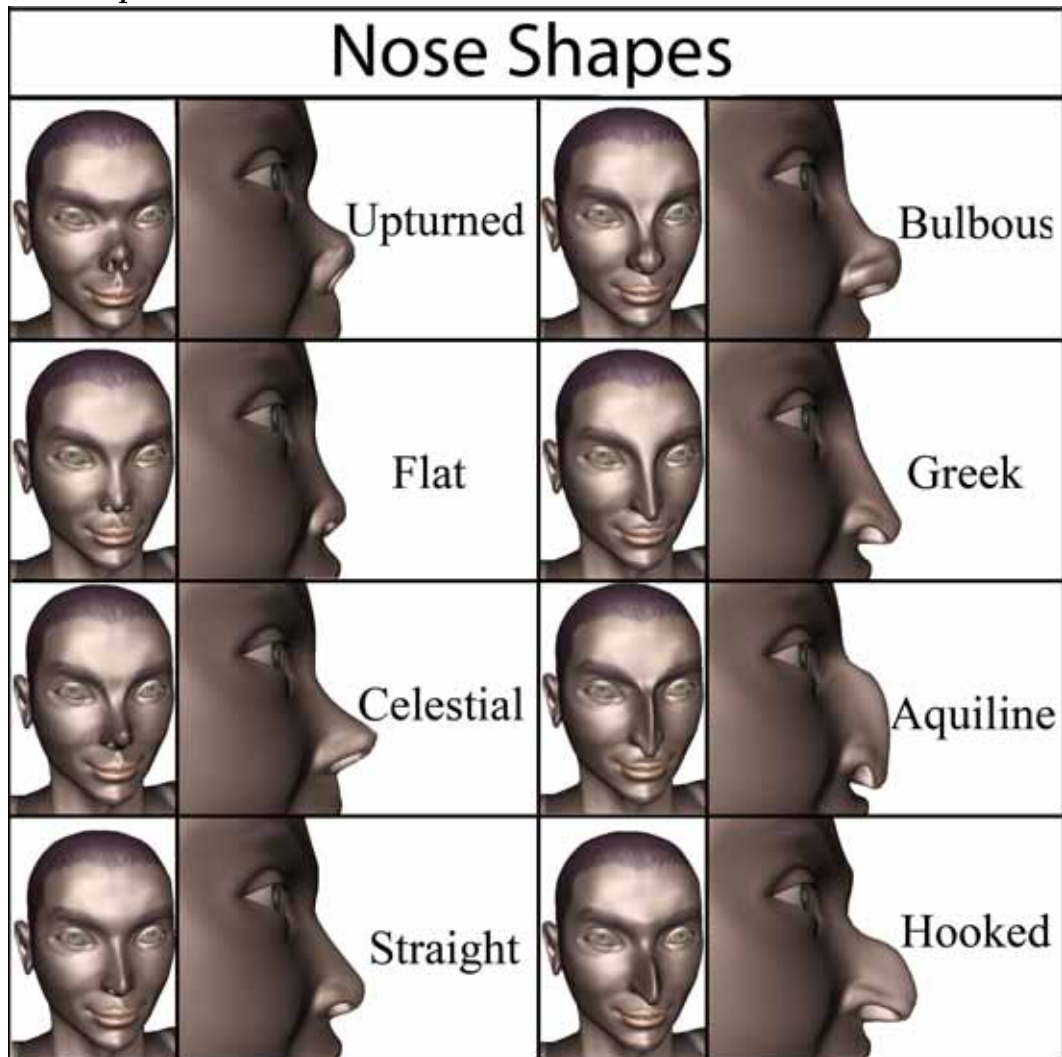


Table 7.6: The effect of reshaping the nasal tip and nostrils.




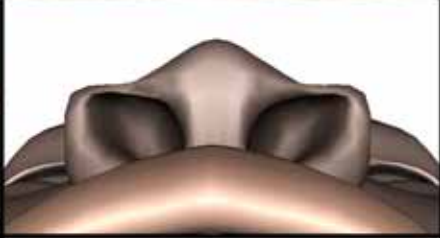



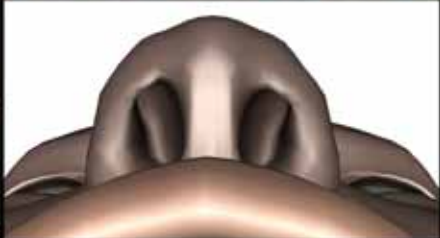



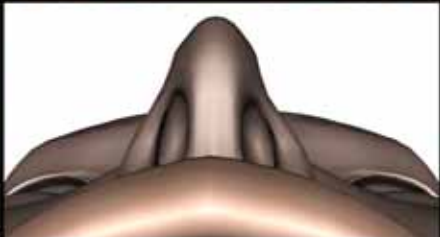









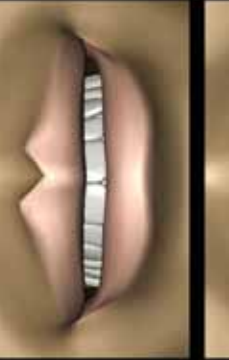














Nasal Tip Shapes		
		Wide
		Divergent Nostrils
		Boxy
		Round
		Triangle
		Narrow

Table 7.7: The parameters describing the shape of the lips.

Lip Shapes							
							
							
							

for the silhouette of the cheekbone that continues along the temples and hairline. There are six major classes of face shapes, which are illustrated in Table 7.8.

7.3.5 Combining Parameters

Using Model-Sketch strokes, the parameter sets describing an individual feature can be independently manipulated, which means that the minimum number of models that can be produced by our system is the product of the number of variations for each feature. The variations we have illustrated in Figures 7.1 through 7.7 include:

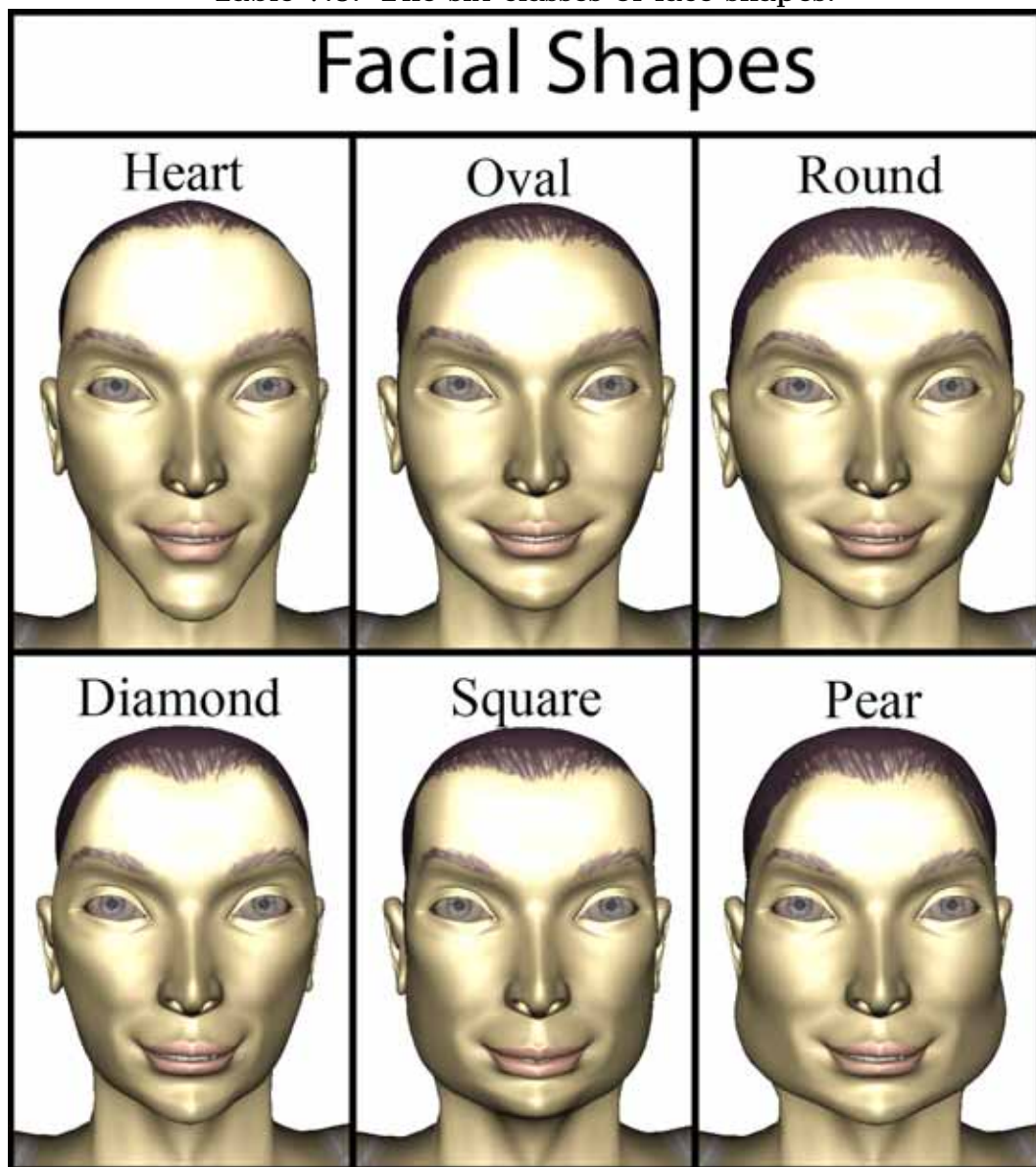
- 2 eye placement parameters for each of the 3 dimensions
- 5 eye shape parameters
- 8 nose profile shapes
- 6 nose tip shapes
- 4 "boolean" parameters for the lips
- 6 face shape parameters

Thus, a lower limit for the number of faces that can be generated from this parameter set is

$$6 \times 2^3 \times 5 \times 8 \times 6 \times 2^4 = 184,320!$$

Of course, this does not account for the range of variation for the eyebrows or ears and assumes that the parameters we described are discrete. In reality, there are many more shapes that can be formed successfully by interpolating between the presented shapes. A creative user may also be able to produce shapes that

Table 7.8: The six classes of face shapes.



cannot be described in terms of these parameters. The freeform nature of sketch-based modeling ensures that no two faces created using the Model-Sketch tool will be exactly alike.

7.4 Body Variation

The body can be modified with in a similar manner as the face, although we have not parameterized the body as exhaustively. A useful feature of the body setup is that the GSH has markers indicating the locations of joints; if Model-Sketch operations change the locations of the markers, the rig can be re-mapped to the new locations. Some variations of the body are shown in Figure 7.5.

7.5 Caricatures

Having visual access to a library of anthropometric measurements that describe the current state of the model is useful for ensuring anatomical coherence. Likewise, the visualization tools can also be used for the inverse task: to exaggerate the "abnormal." Our interface indicates exactly how a given facial model deviates from the "average" face. Boxes are drawn to indicate the range of allowable positions for a given Landmark, with a line from the center of the box through the center of the landmark. The landmark can be moved further into the box along the direction line to make the model more "normal", or moved in the opposite direction to create a caricature (Figure 7.6). Some example caricatures are shown in Figure 7.7.

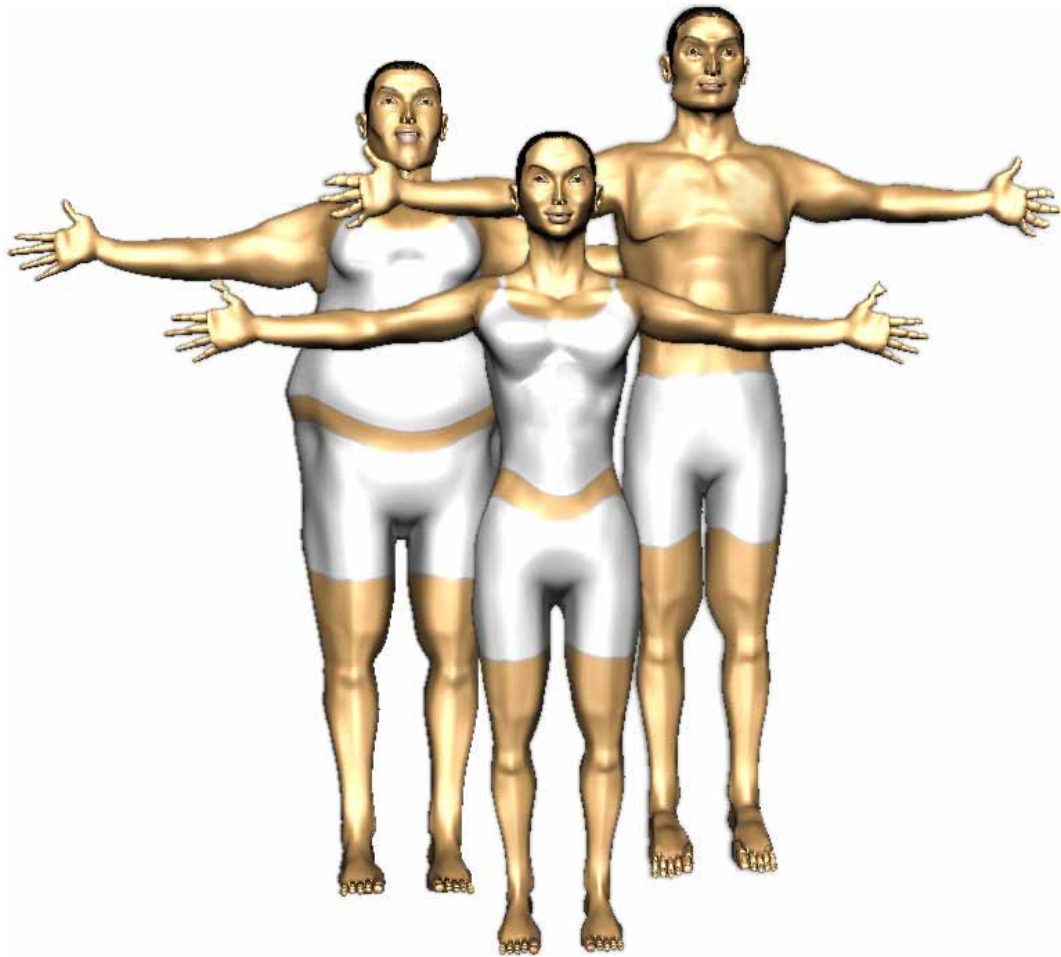


Figure 7.5: A small number of strategic Model-Sketch operations can be used to re-sculpt Frankie's body. Three variations of Frankie's body type are shown: heavy-set, feminine, and masculine.

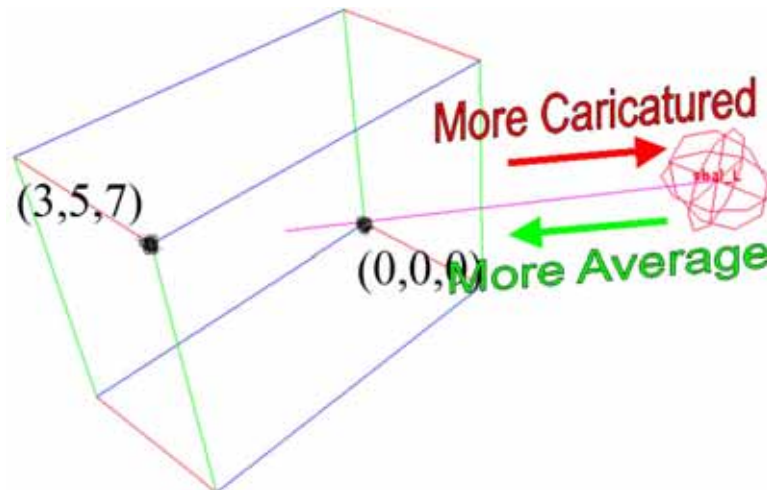


Figure 7.6: Creating 3D caricatures is made simple by the easily visualized anthropometric limits on each Landmark. Moving a Landmark towards the center of its limiting "box" will achieve a more realistic appearance whereas moving it outward will exaggerate the feature represented by the Landmark.

7.6 Results

The versatility of the Model-Sketch system is illustrated in Figures 7.8 through 7.15. As shown, Frankie's ethnicity, gender, weight, expression, and, to a certain extent, age can be varied flexibly and efficiently. The interface also facilitates the creation of caricatures and human-like fantasy characters.

7.7 Performance

The system is fully interactive. The average amount of time spent on each individual face model in Figure 7.15 was approximately five minutes. Most faces required at least one modifying stroke for the nose profile, at least one for the lips, at least two for the eyes (one for placement and one for eyelid shaping), and at least one for altering the facial shape. On some models, the ears, nose tip, nose bridge, and nostrils were also modified. More strokes were applied for further refinement. Not



Figure 7.7: On the left are original photographs of some recognizable actresses [ABC]; on the right the faces in the photographs are substituted with 3D caricatures produced using our interface.



Figure 7.8: Four variations of Frankie's face.

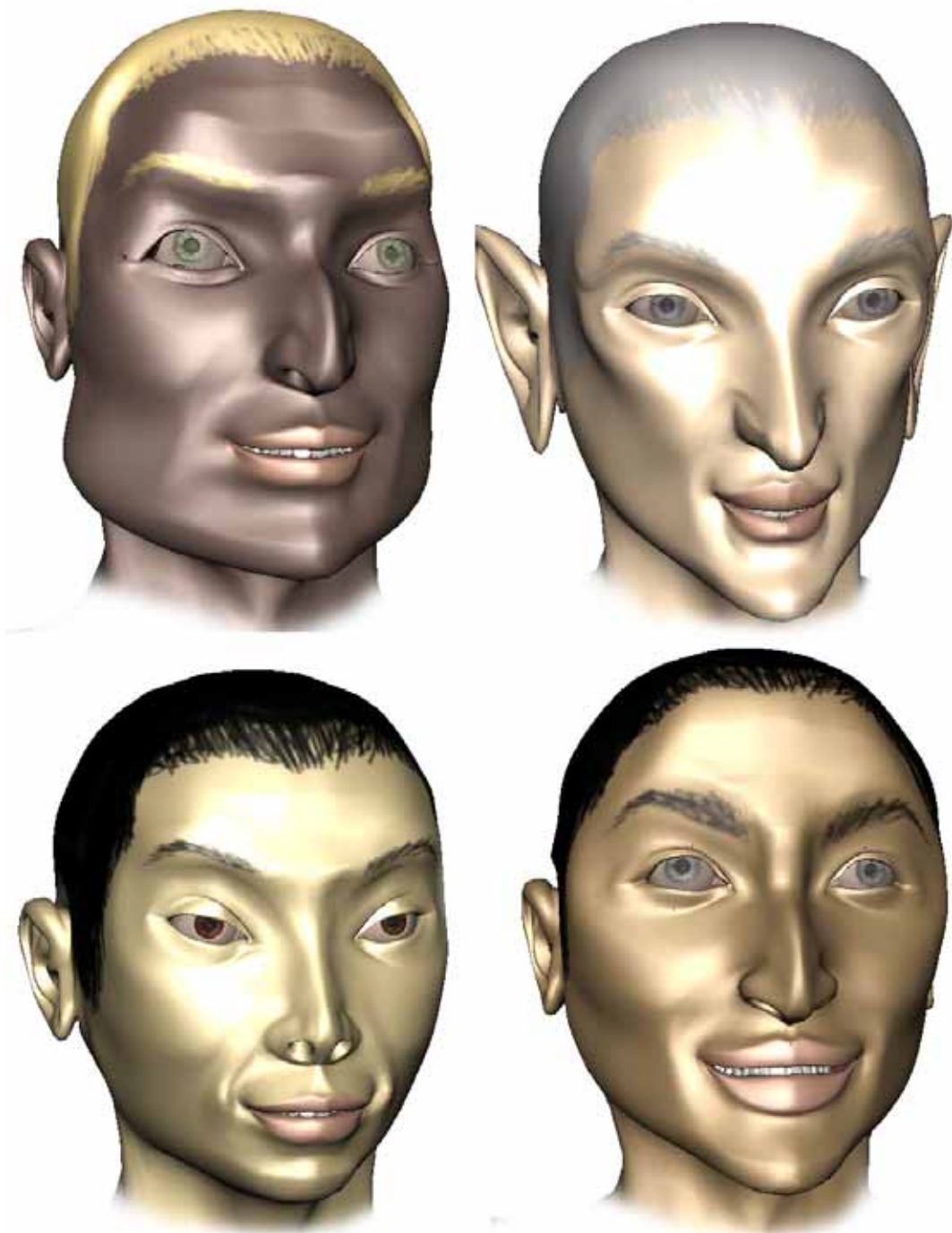


Figure 7.9: Four variations of Frankie's face.

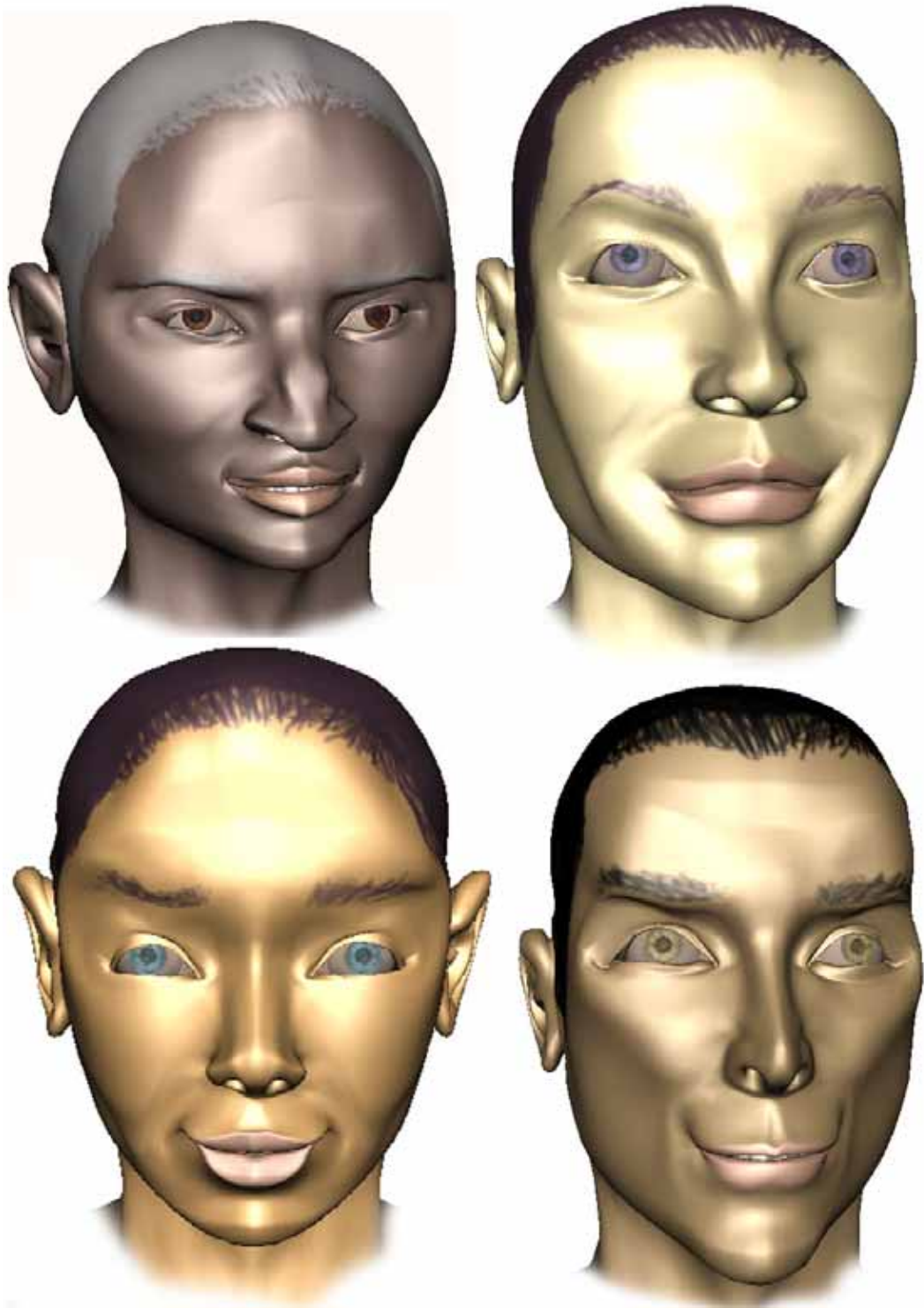


Figure 7.10: Four variations of Frankie's face.

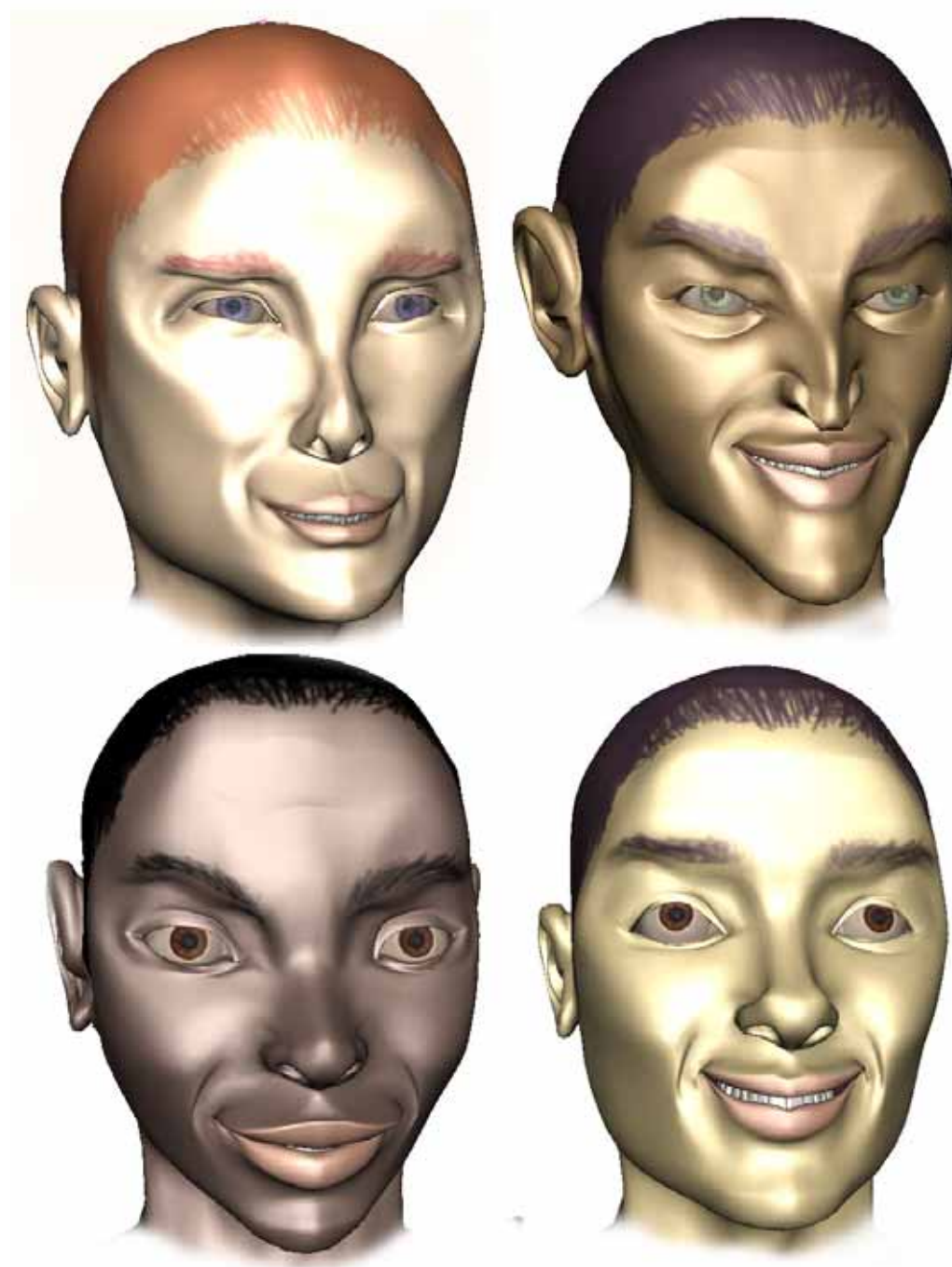


Figure 7.11: Four variations of Frankie's face.

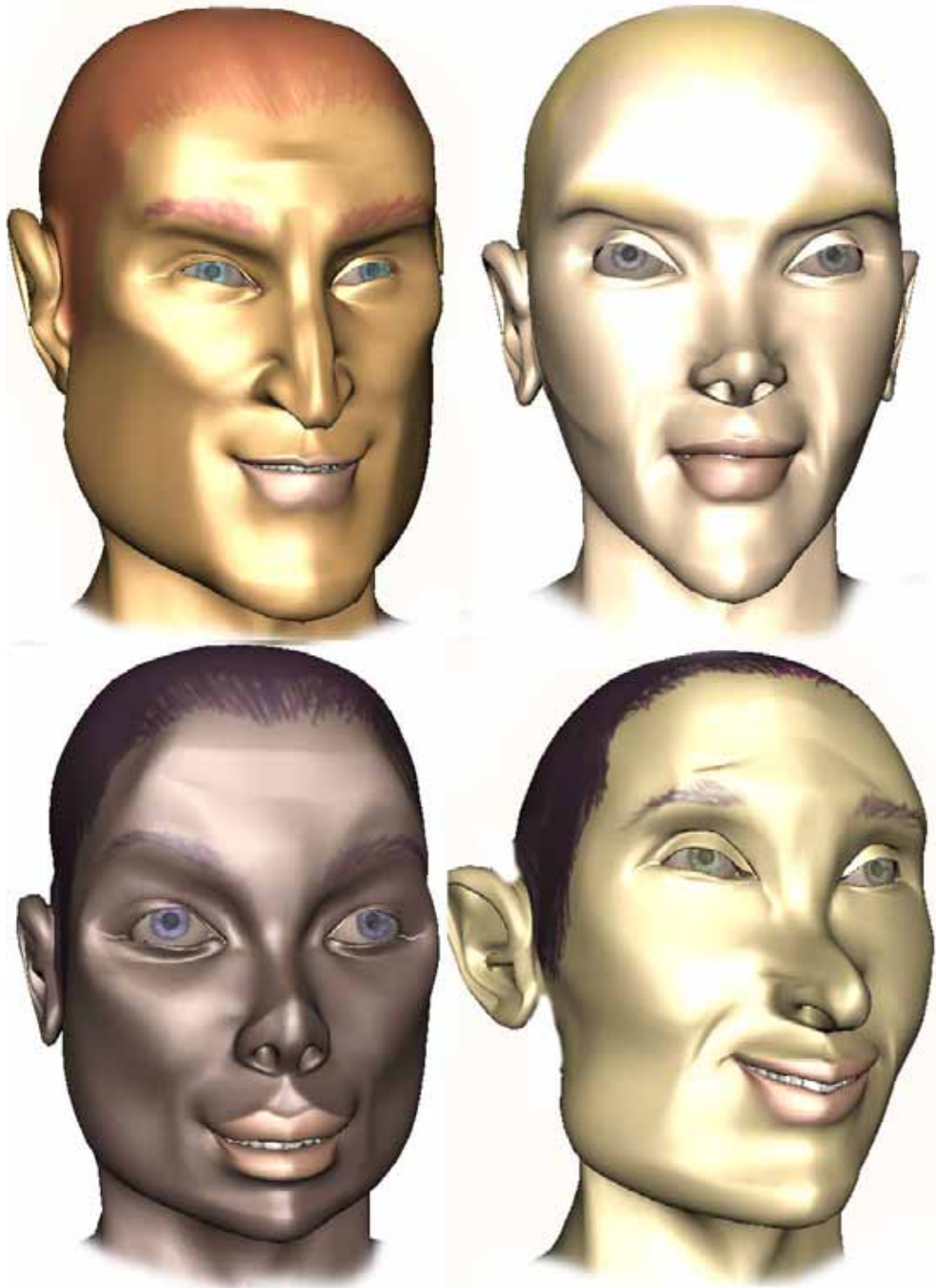


Figure 7.12: Four variations of Frankie's face.

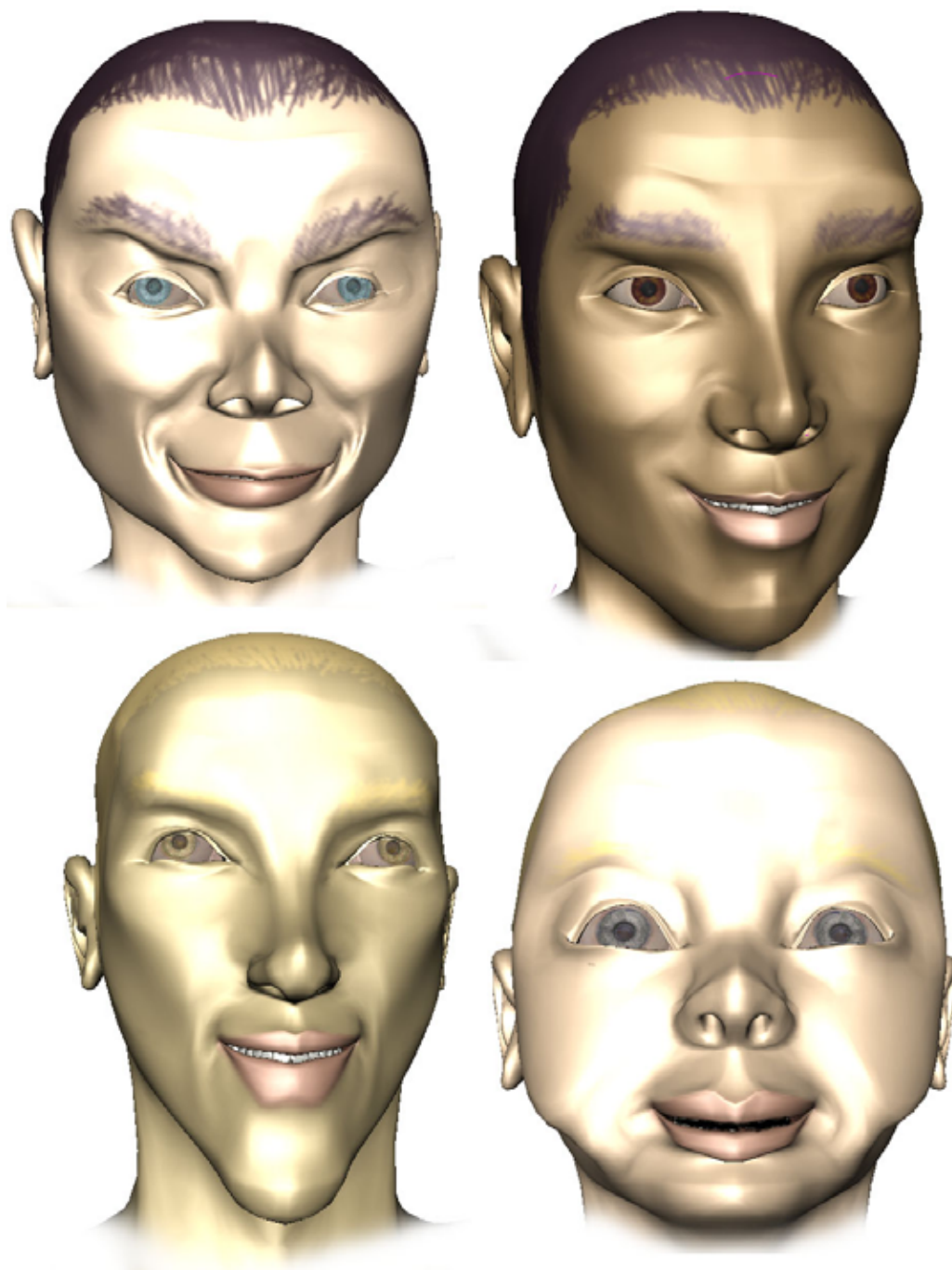


Figure 7.13: Four variations of Frankie's face.

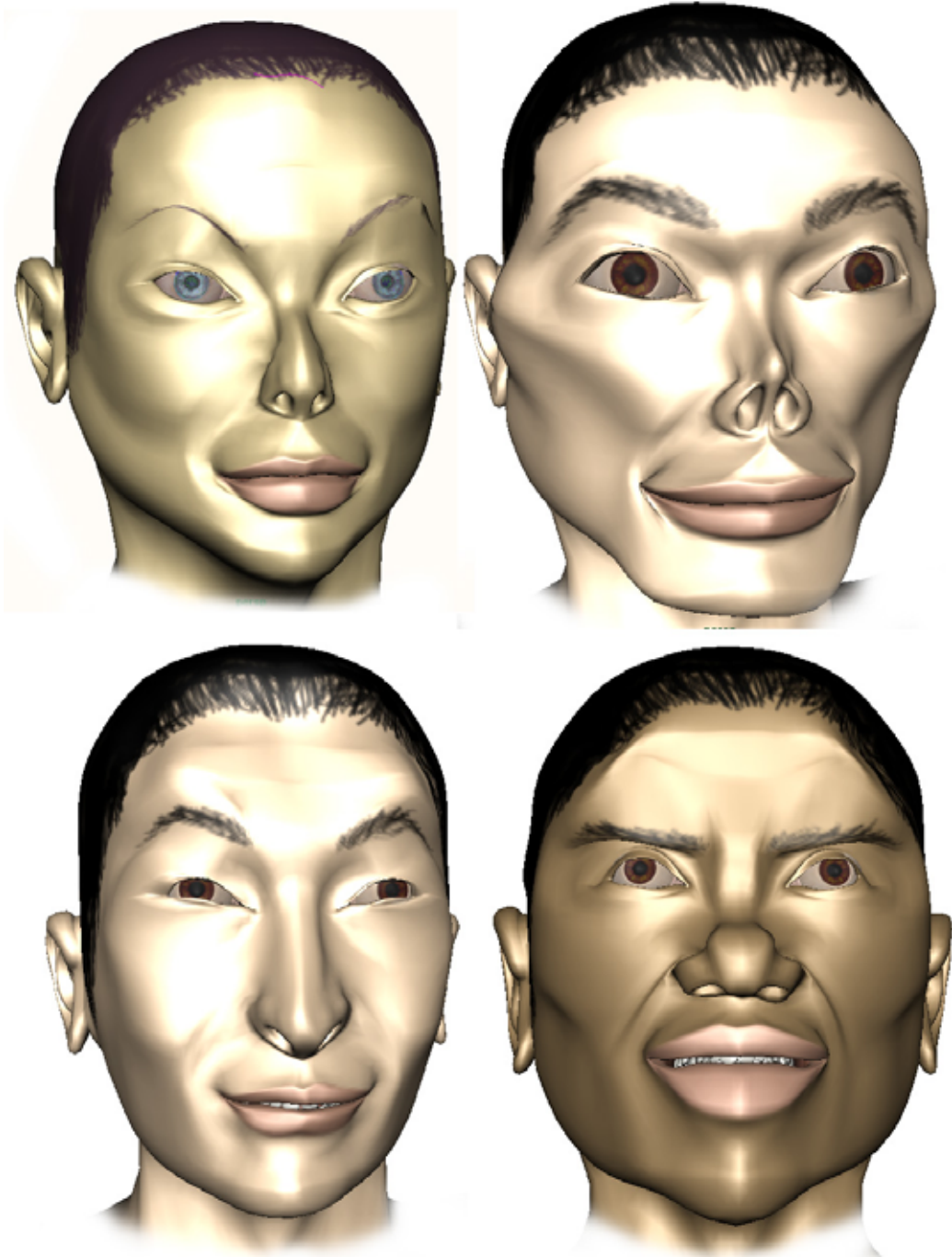


Figure 7.14: Four variations of Frankie's face.

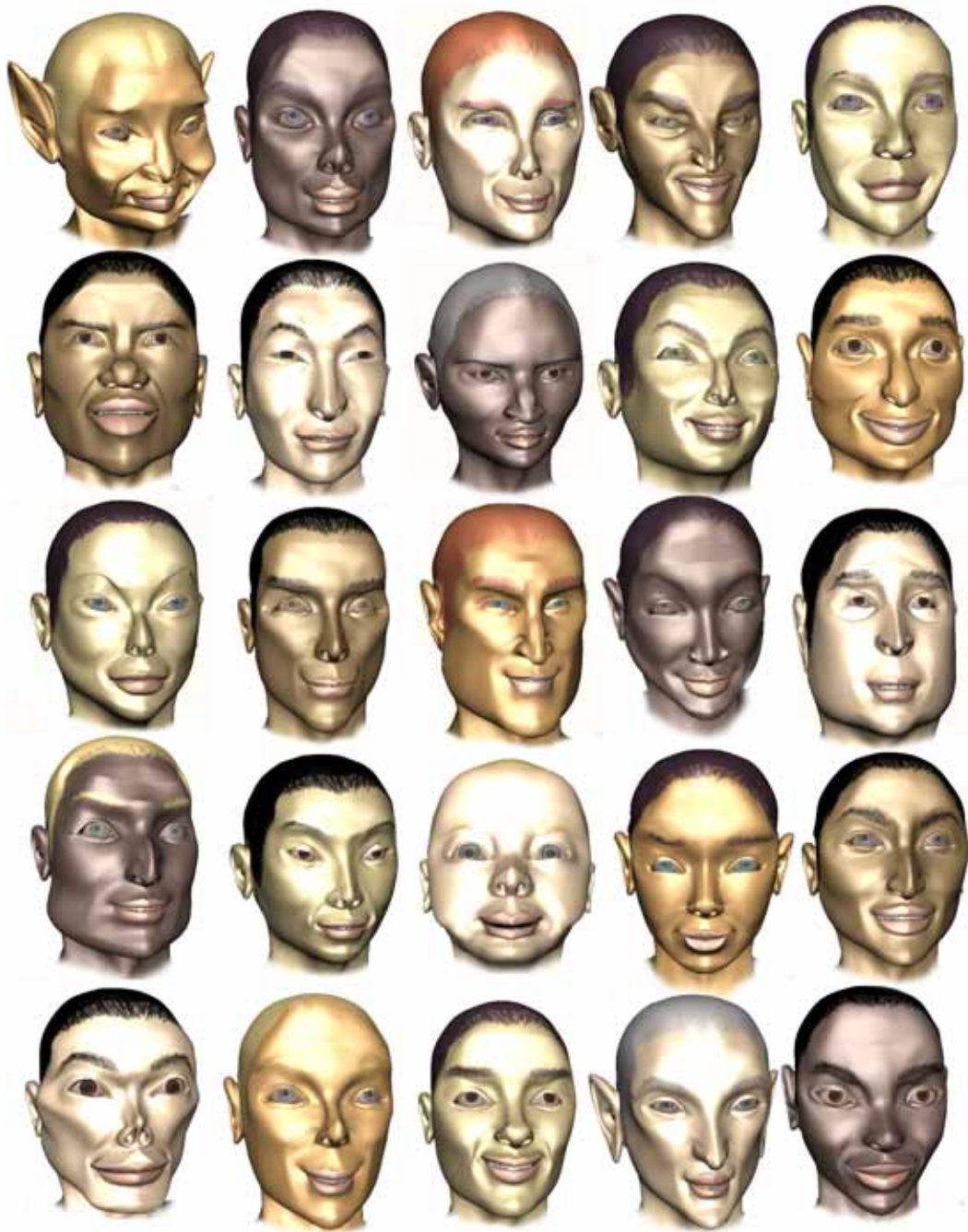


Figure 7.15: Each face was modeled in an average of five minutes.

surprisingly, the eyes, which are the most perceptually important part of the face, generally required the most refinement.

The maximum amount of time spent on any of the models was twenty minutes. Incidentally, this was for the caricatures. Creating a likeness is a challenging task with any media, thus the caricature models underwent several iterations of refinement to capture the personalities of their subjects. As intended, much of the success of such models is dependent on the user's talent. Since the artist does not have to worry about the initial shape creation phase, more time can be devoted to the refinement phase and fully realizing the artistic vision.

7.8 Limitations

Because our system was designed to maximize artistic freedom, we chose to limit the number of constraints on sketch-based operations. Thus, the system assumes that the user will always provide valid input. Unanticipated results can be produced when the input is invalid. To ensure intuitive results, the user must follow some guidelines for both Landmark selection and "sketching".

First, the user must be aware of which Landmarks are selected before drawing a new curve. Our Landmark selection constraints are designed to preserve relative locations, and thus in-between Landmarks that were not selected by the user are selected automatically. If users are not aware of the automatic selection, they may draw curves that do not account for these landmarks.

This leads to the second rule: The drawn curve must have at least as many identifiable "landmarks" as the number of Landmark locators selected. Sketch curve "landmarks" include the endpoints of the curve, as well as local maxima and minima. Thus, if more than two landmarks are selected, a perfectly straight

sketch will result in an erroneous fitting. However, this type of error is extremely rare because the sketch curve is recorded at a high resolution; drawing a perfectly straight line using a stylus is virtually impossible. The more common potential error caused by storing the sketch curve at high resolution is that any unintentional "slips" of the hand could be interpreted as "landmarks", if they are more pronounced than maxima and minima on the rest of the curve. Thus, the most predictable results are delivered when the drawn curve has easily identifiable local maxima and minima.

The third rule involves the viewport in which the operation is performed. Recall that before the curve fitting algorithm is executed, both the GSH curve connected to the selected Landmarks and the sketch curve are projected onto the canonical viewing plane closest to the viewing plane on which the sketch curve was drawn. Thus, the most intuitive results are delivered when the sketch curve is drawn in a canonical view or a view that is unambiguously close to exactly one canonical plane. Also, since most GSH curves lie roughly on canonical planes, the user should not attempt to "re-draw" them on their perpendicular planes. For example, the face profile should not be Model-Sketched in the top view because the projection of the profile curve onto the XZ plane will have overlapping segments.

Because Model-Sketch operations are undoable, occasional erroneous strokes will not halt the workflow. As with any artistic tool, the user may iteratively make many changes to the model before being satisfied with the product. However, unlike conventional 3D modeling tools, each "iteration" of Model-Sketch modification compactly produces a seamless, meaningful change to the model in a single intuitive step.

Chapter 8

Conclusions and Future Work

It is said that a tool is "only as good as the artist who uses it." One may argue that a truly sophisticated tool can help cultivate creative expression by minimizing the time and effort necessary to translate artistic vision into tangible forms. Our primary goal was to promote a harmony between imagination and the tool that expresses it.

We have developed a unique approach to artistic representation of the human form. Our method combines the advantages of 2D and 3D shape creation by implementing an intuitive sketch-based interface with connections to parametric measurement mechanisms that help the artist maintain the model's desired proportions. Our system bridges the fine line between a "constrained" and "freeform" interface by harnessing the model so it can be reshaped in meaningful ways, yet displaying useful parametric constraints which function as non-obtrusive "guides" that will not revoke the user's artistic licence.

To begin the process, we have constructed a detailed, malleable prototype model whose topology is optimized for animation. The model, "Frankie", is equipped with a generic rig, customizable textures for the skin, hair, and eyes,

and a unique deformation system, the Gesture Sketch Harness or "GSH", which serves as both a visualization and a modeling mechanism. Built of NURBS curves that follow the most descriptive and recognizable silhouettes of the body, the GSH, when viewed by itself, is reminiscent of simple gesture sketches drawn by 2D animators. Each curve is applied as a wire deformer to the prototype mesh so that the silhouette of the mesh closest to the given GSH curve will always conform to the outline defined by the curve. Thus, by interactively reshaping the curves, the user can indirectly but smoothly change the outlines of the model.

Our implementation of the anthropometric parameter set applied to the prototype model consists of three unique Maya nodes: Landmarks, Parameters, and Parametric Models. Landmarks are selectable locator objects, displayed as spheres, that we placed on the prototype model to identify anthropometric locations. Each Landmark controls a manually assigned *cluster* of control points in the GSH with weighted influence. Thus by moving the Landmark, the user may deform a part of the GSH, and consequently the mesh, in anthropometrically meaningful ways.

To keep track of anthropometric measurements as the model is deformed, each Landmark is connected to a number of Parameter nodes. A Parameter node encapsulates one of three types of anthropometric measurements: shortest distance, axial distance, or angle of inclination. Each Parameter utilizes two Landmark locations as inputs and uses them to calculate a value for the specified type of measurement. The node also records the current measurement's deviation from a user-specified mean value. Finally, using these values, the Parameter node generates linear limits for the positions of the two landmarks such that if both Landmarks are within their limits, the Parameter measurement will remain within the "normal" range of values.

The Parameter and Landmark nodes interact cyclically to form the "constraints" on our model. The location of a Landmark is sent to a number of Parameters, each of which determines linear limits for the Landmark, which in turn are returned to the Landmark. Thus, the Landmark is passed several sets of limits, and must calculate the intersection of these limits in order to form a spatial domain that, when satisfied, will ensure that the prototype model remains anatomically proportionate. Because our goal is to provide a *flexible* interface for artists, our system does not "force" the Landmark into this calculated domain, but instead displays the limits in the form of a box. This serves as a cue when the model no longer satisfies the "correct" proportions for a human face; the artist may then choose to move the Landmark into the box to restore realism, or simply ignore the box and proceed to create a model with a caricatured or stylized look.

Finally, our system provides a Parametric Model node that functions essentially as a hub for all the Landmark and Parameter nodes associated with a given data set. While our prototype model uses a single Parametric Model node, it is possible to apply several to a model. For example, if a user wishes to use facial data from one source, and body measurements from another, two separate Parametric Models can be used to organize the data. This internal setup will have no effect on the modeling process, but becomes useful if one wishes to export separate sets of data from the model. Future work may focus on novel ways to combine and organize Parametric Models for various applications.

Our interface condenses and optimizes the workflow of 3D modeling. In most 3D modeling packages, the user must physically move the cursor away from the model to the toolbar to switch between tools. In some cases, the artist must scroll through various menus to find the appropriate tool or deformer. While

the tools themselves are sophisticated, these small but frequent interruptions to the creation process become distracting. Thus, encapsulating all the necessary modeling functionality in a single tool streamlines the workflow, allowing the artist to focus on the model rather than the tools.

Given this observation, we have designed a simple and intuitive interface that involves just one tool: the Model-Sketch pencil. It functions as both a highlighter and a sketching pencil. In "highlight" mode, dragging the stylus selects all Landmarks in its path. When the pen is lifted, the tool switches to "sketch" mode. In this mode, dragging creates a NURBs curve to be used as a guide for deformation of the model. When the pen is released, the region of the model defined by the Landmarks selected in the previous step deforms interactively to match the sketched NURBs curve. The tool then switches back to "highlight" mode so that the user can select a new set of Landmarks and repeat the process.

Internally, the fit is achieved in two steps, Landmark mapping and curve matching. In the first step, the the most distinctive points on the drawn NURBs curve, particularly the endpoints and the curvature maxima and minima, are isolated as "potential matches" for the selected Landmarks. Then, an error minimization algorithm is used to map selected Landmarks sequentially to the best matches on the drawn curve. The Landmarks are moved to these locations, giving a rough initial approximation to the curve. To refine the fit, the second step moves the control vertices on the GSH curve regions between the selected Landmarks to the closest corresponding locations on the drawn curve. Consequently, the model is reshaped to fit the silhouette of the newly sketched curve. The algorithm is interactive; all the user sees after the initial sketch is the newly shaped model.

Our method differs from previous approaches to sketch-based model editing in

two significant ways. First, our approach works from a pre-defined set of canonical silhouettes, namely the GSH, rather than silhouettes detected "on the fly." Since the GSH contains a relatively detailed set of curves that encompasses the most commonly recognized and drawn silhouettes, our method efficiently eliminates the step of initial silhouette detection without a significant loss in the amount of detail that can be "sketched" into the model. The GSH also functions as an easily interpreted visualization mode. Second, while some sketch-based interfaces require users to select sets of vertices directly on the mesh, our Landmark-based selection method ensures a cleaner selection and frees the user from having to make sure that undesired parts of the model are not accidentally selected.

Our system also supports local symmetry for Landmarks, which allows the user to maintain symmetry for certain facial features with the option of asymmetry for others. However, the implementation does not support symmetry for GSH curves. Future implementations may consider implementing a symmetry constraint for the GSH; however since faces and bodies are not perfectly symmetrical, the constraint may not be necessary for all applications.

While our interface is tailored to human characters, the framework and concepts we have presented can be generalized to other types of organic models. Our implementation contains a file translator that can read a table of landmark names, anthropometric data, and skeletal hierarchy specifications from a specially formatted text file, and use the information to generate a network of joints, Landmark and Parameter nodes in a Maya file. Using this format, the user can create arbitrary parameter sets. To attach the newly generated Landmarks to an arbitrary model, one must first create a GSH for the model. We have provided two curve drawing tools to expedite the process of drawing curves directly on a mesh. Then,

the curves must be applied to the model as wire deformers, with weights and falloff distances specified by the user. Cluster deformers must then be created for sets of control points on the model, with one cluster corresponding to each Landmark. Finally, the Landmarks should be placed on significant locations of the model, and each should be made parent to the corresponding cluster. With this setup, our Model-Sketch tool will work exactly as it does for our prototype human model.

Our interface aims to bring a new level of artistry to CG modeling and animation. The fully animatable models generated can be used in films, video games, or VR simulations where specific people and situations need to be modeled. The ability to make quick edits to a large number of models can simplify crowd generation; an artist can make each person unique without spending hours on each model. Knowledge of the parametric measurements can be used to generate clothing that fits the character. The constant topology can facilitate automatic placement of hair onto specific polygons. Additionally, the parameter set can be edited to represent the constraints of different human-like characters, such as cartoons, anime, caricatures, and anthropomorphs.

While catered to an artistic environment, our system has exciting possible applications in several different fields. The ability to measure and store a model's distance from the "average" person can be used by tailors or fashion designers to derive measurements for garments, by police detectives to get realistic 3D sketches from witnesses descriptions, and by plastic surgeons to develop plans action and show patients realistic depictions of expected results, while having access to specific measurements that can be used to determine the feasibility of such a change.

The system, built upon anthropometric parameters, can be used simultaneously for the purpose of further anthropometric research. Traditionally, data sets

such as CAESAR [CAE] calculate anthropometric measurements from hand-picked landmarks on the scanned meshes. With large sets of landmarks, the task of manually designating them becomes tedious. Alternatively, a researcher could quickly fit Frankie to the scan by using our Model-Sketch tool to move a large number of Landmarks at a time, and then immediately enjoy access to a large number of interactively calculated measurements, along with mechanisms to visualize them.

Even without 3D scans as reference, our system can be used to gain a better understanding of facial and body proportions. Some facial research is done using 2D "composite" photographs of a population. By having an artist translate these composite photographs into a 3D model using our system and a keen eye for depth cues, a scientist may then read the measurements calculated by the system and derive estimated average anthropometric measurements for the population.

We have successfully created concurrency in the steps of rigging, modeling, and texture mapping of a human character. Future work may attempt to generalize sketch-based concepts to other parts of the animation pipeline, such as skinning, texture painting, and lighting.

Future research may also aim to devise a more robust curve fitting algorithm. While our heuristic works well in most cases, occasionally a Landmark is mapped to an unexpected point on the sketch. This may not negatively affect the shape of the model, but the incorrect placement of the Landmark results in inaccurate anthropometric measurements. Because our system moves Landmarks sequentially, two Landmarks can never be mapped to the same point, but since some of the "potential landmarks" are very close together, two Landmarks may come close to overlapping. More stringent error criteria or a more selective process of determining "potential landmarks" may solve this problem; however this may narrow the list

of "potential landmarks" more than is desirable. The fundamental problem is that it requires human intelligence to accurately identify anthropometric landmark locations on an arbitrary curve; future implementations may attempt to minimize this problem by incorporating sophisticated computer vision techniques.

For centuries, artists have brought characters to life using nothing but a pencil. A skilled artist can capture a model's unique features, expression, and personality on paper in less than five minutes. Expressing this level of artistic intuition in a 3-D model using mesh editing tools is not impossible, but traditionally the process is tedious and unintuitive. The initial steps of shape creation and topology optimization leave little time for refinement in some production environments; our system condenses those steps so that more time can be spent in the refinement stage, stimulating new levels of creativity.

References

- [3DT] 3DTotal: The CG Artist's Homepage. <www.3dtotal.com>.
- [ABC] ABC. Grey's Anatomy. Photos of Ellen Pompeo, Sandra Oh, and Chandra Wilson. <abc.go.com/primetime/greysanatomy>.
- [ACP03] Brett Allen, Brian Curless, and Zoran Popovic. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, 2003.
- [Aut] Maya 6.5. Autodesk, Inc. San Rafael, CA.
- [Beaa] Caricature of Mr. Bean. <blog.doctissimo.fr/php/blog/Passions>.
- [Beab] Photograph of Mr. Bean. <media.scotsman.com>.
- [Bib04] Jacobo Bibliowicz. An automated rigging system for facial animation. Master's thesis, Cornell University, 2004.
- [Bri03] Ken Brilliant. *Building a Digital Human*. Charles River Media Inc, 2003.
- [BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.
- [CAE] CAESAR. Civilian American and European Surface Anthropometry Resource Project. <store.sae.org/caesar>.
- [Cal] CalArts Animation Lab. Photograph of a student using Maya. <www.calarts.edu>.
- [Cin] WACOM, Ltd. Photograph of WACOM Cintiq tablet in use. <www.bubilevar.com/images/wcmc21.gif>.
- [CRE] Commission for Racial Equality. Poster. <www.cre.gov.uk>.

- [Dis92] Disney. *Model Sheet for Princess Jasmine*. Aladdin Central, 1992. <www.aladdincentral.org/images/sketch.html>.
- [DMS98] Douglas DeCarlo, Dimitris Metaxas, and Matthew Stone. An anthropometric face model using variational techniques. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 67–74, New York, NY, USA, 1998. ACM Press.
- [Dun03] J.R. Dunster. *Drawing Portraits: Fundamentals*. J.R. Dunster (USA), 2003. A companion book for <portrait-artist.org>.
- [Edw79] Betty Edwards. *The New Drawing on the Right Side of the Brain*. Tarcher Putnam (USA), 1979.
- [EF82] Paul Ekman and Wallace V. Friesen. Measuring facial movement with the facial action coding system. In Paul Ekman, editor, *Emotion in the Human Face*, chapter 9, pages 178–211. Cambridge University Press, second edition, 1982.
- [EFr] Poser 6. E-Frontier America Inc. Scotts Valley, CA.
- [Fac] Face Research Website. Psychology experiments about preferences for faces and voices. <www.faceresearch.org>.
- [Far94] Leslie G. Farkas. *Anthropometry of the Head and Face*. Raven Press (USA), second edition, 1994.
- [Gog] Vincent Van Gogh. Jozef Blok Portrait. <www.latribunedelart.com>.
- [Gou03] David A. D. Gould. *Complete Maya Programming, An Extensive Guide to MEL and the C++ API*. Elsevier Science (USA), 2003.
- [Gou05] David A. D. Gould. *Complete Maya Programming, Vol. II: An In-Depth Guide to 3D Fundamentals, Geometry, and Modeling*. Elsevier Science (USA), 2005.
- [Idl] Idleworm. Tutorial: Animating a 2D Walk Cycle. <www.idleworm.com/how/anm/02w/walk1.shtml>.
- [IMT99] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [Int] WACOM, Ltd. Photograph of WACOM Intuos tablet in use. <www.wacom.com/pressinfo/photography>.

- [JM91] Mark H. Johnson and John Morton. *Biology and Cognitive Development: The Case of Face Recognition*. Blackwell Publishers (USA), 1991.
- [LTW95] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62. ACM Press, 1995.
- [Mar04] Chris Maraffi. *Maya Character Creation-Modeling and Animation Controls*. New Riders Publishing, first edition, 2004.
- [NAS] NASA. Anthropometry and Biomechanics. <msis.jsc.nasa.gov>.
- [NSACO05] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph.*, 24(3):1142–1147, 2005.
- [Par72] Frederic I. Parke. Computer generated animation of faces. In *Proceedings of the ACM National Conference*, volume 1, pages 451–457, 1972.
- [Par82] Frederic I. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, November 1982.
- [Pix] Pixar Animation Studios. Concept Art and Color Keys from *Finding Nemo*. <www.magicalears.com>.
- [PSU] Penn State University. Animation Pipeline. <viz.aset.psu.edu>.
- [PT97] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer-Verlag, second edition, 1997.
- [PWC] Prince of Wales Collegiate. Claymation at PWC. <www.pwc.k12.nf.ca/projects/claymation>.
- [Ren] Ron Rensick. Clinton Illusion. <www.gerstein.info/fun>.
- [Vin] Leonardo Da Vinci. Vitruvian Man. <www.danbrown.com>.
- [WC05] Chun Wang and Ming Yuan Chuan. *Cubic Tragedy*. National Taiwan University of Science and Technology, 2005. Animated Short.
- [ZHH96] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: An interface for sketching 3d scenes, 1996. <www.cs.brown.edu/bcz/sketch/sig.html>.