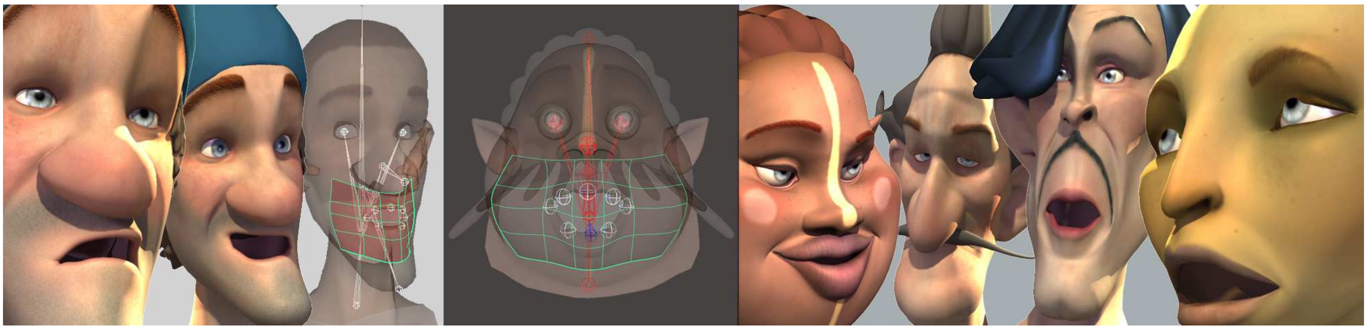
Verónica Costa Teixeira Orvalho

REUSABLE FACIAL RIGGING AND ANIMATION: CREATE ONCE, USE MANY



(Copyright 2005 Dygrafilms)

A DISSERTATION

in

Software (Computer Graphics)

Presented to the Faculties of the *Universitat Politècnica de Catalunya*

in Partial Fulfillment of the Requirements for the *Degree of Doctor of Philosophy*

Advisor: Dr. Antonio Susin
Tutor: Dra. Isabel Navazo

Barcelona, June 2007

Abstract

Facial animation is a serious bottleneck in any computer generated (CG) production. It is the key element to convey emotion to 3D characters. Speeding up the rigging process remains an unsolved problem, specially for film and videogames, which require high quality results. The character rigging is analogous to setting up the strings that control a puppet. Today, skilled artists manually create the facial rig to ensure the best quality in the animations; but, this is a slow, labor-intensive and costly process.

This thesis presents a portable character rigging system that integrates into current animation production pipelines. It automatically transfers the facial rig and animations created for one character to different characters, independent of their shape and appearance. It enables artists to create more lifelike facial models in less time; about 90-99 percent faster than traditional manual rigging. Characters can display complex expressions and behavior, easier and with decreased artistic effort. As a result, we dramatically reduce the time needed to create high-quality facial animations for the entertainment industry.

We studied techniques from the fields of computer graphics and computer vision, to come up with a solution to the rigging problem. Based on a generic facial rig definition and a new deformation method, our system converts 3D face models into digital puppets that experienced artists can control. The system adapts the skeleton, weights and influence objects (NURBS surfaces, lattice, etc.) from a source rig to individual face models to obtain unique expressions, and enables easy reuse of existing animation scripts. Our work differs from previous morphing and retargeting techniques, because that work was oriented towards transferring animations, while ours aims to transfer the complete facial rig, in addition to animations.

The system was validated with a series of experiments. We used models and rigs from major film and videogame companies: Electronic Arts, Radical, Dygrafilms. The results were supervised by Technical and Art Directors, who approved the quality of our rigs and animations to be used in CG productions, replacing the artist generated ones. Our proposal is: generic (the facial rig can have any type of configuration and models can be created by artists), flexible (the rig has no initial constraints), independent of the shape and appearance of the model, and enhances the freedom of the artists (does not force the use of a predefined rig). Now, motionless and inanimate faces can come to life with the help of our technology.

Acknowledgements

This long walk started five years ago, when I took the first step to follow my dreams of working within the entertainment industry. But the journey would have not been possible without the support and friendship of many people. I would like to thank them with all my heart for helping me accomplish this goal.

First, I express my deepest gratitude to my advisor Toni Susin, who rescued me from the bottom of the sea. He was always a great source of encouragement and guidance. Also Isabel Navazo, for providing academic advice.

A very special thanks to Ernesto Zacur, which helped at the early steps of my research by sharing his work on facial reconstruction with me. His invaluable feedback inspired many of the current features of the system.

Juan Nouche and everyone at Dygrafilms, for seeing the potential of the technology at an embryonic stage and believing in me. Thanks for all the time, dedication and material so important to validate the system. I am specially grateful to Xenxo Alvarez, who dedicated many many hours to thoroughly test the application and coordinated the creation of the 3D models and animations; his suggestions have greatly improved the system. I extend my thanks to his team: Rubén Lopez, Juan Ramón Pou, Esteban Ferreño, Maria Lodeiro, Cristina Conde and Cristian Parrado.

Miguel Angel Sepúlveda for helping me narrow down the research focus and for his initial orientation.

Kelly Booth from UBC (University of British Columbia) for his hospitality (Tojo's was delicious!) and dedication in putting me in contact with so many people that later helped my research. Sid Fels for making my stay at UBC's HCT lab not only fruitful but also fun.

Jean-Luc Duprat (then at Electronic Arts), for his friendship (and Belgium beers), for organizing interesting meetings and for giving me very good advice on how to approach the entertainment industry. Fred Fowles and Crystal Wang from EA, for always having time to answer my questions, provide me with high quality test material and their constant words of motivation.

Dave Fracchia (then at Mainframe Entertainment) for sharing his insightful knowledge on the film industry, which helped me define my carrier goals, and his tips on how to improve my demos.

Juan Pablo Buscarini, who gave me the opportunity to get started in the film industry. He inspired my current path.

My friends all over the world, who reminded me from time to time that real life goes on outside the silicon chips. And my godmother Raquelita, for helping me become a better person each day.

My husband’s extended family, for making me part of their lives.

My parents, for always making sure that I received the best education. My sister, because I love her.

Last, but the most, I deeply thank my husband João, for being my source of inspiration and everyday happiness. For his unconditional love and support. For keeping me going in the darkest hours. For all the days he spent fixing my clumsy writing. For giving me constant feedback to improve the research. For reviewing my work, probably as many times as I did. And much much more. Without him, everything would be different...

To João Orvalho for his endless love.

“Where we touch, the fibers merge
and intertangle.

I am no longer
certain where I
end...where
he begins...”
- Alan Moore, Swamp Thing (1990)

“When all I want is you...” - Bono (1988)

Contents

List of Figures

Chapter 1

Introduction

Facial Animation *is the key element to convey emotion and personality to a character. The entertainment industry requires high quality results and drives the research efforts to automate the character setup process for realistic animation of 3D characters. We studied techniques from the fields of computer graphics and computer vision, to come up with a solution that speeds up the rigging process dramatically. We propose a portable character rigging system that integrates into current animation production pipelines, enabling digital artists to create more lifelike characters in less time, when compared to traditional animation techniques. It automatically transfers the rig and animations created in one character to different characters, independent of their shape and appearance. This chapter briefly describes the main problems when preparing a character for animation, gives an overview of our solution and summarizes our research contribution.*

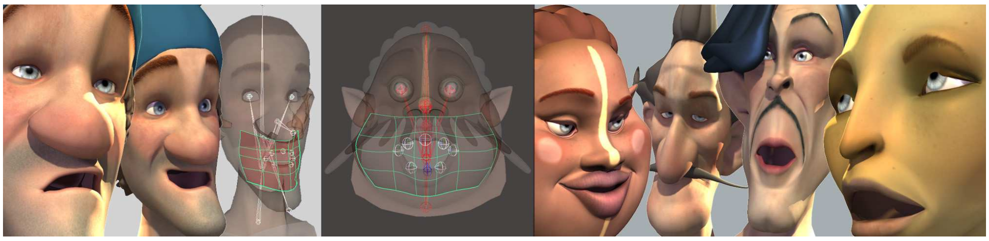


Figure 1.1: Source model rig and expressions (left); target model rig transferred from the source (middle); expressions from different target models extracted from an animation sequence (right). (Copyright 2005 Dygrafilms)

1.1 Motivation

The PhD research started in 2003 after several talks and inquiries with people in the USA digital animation film industry. We assembled a list of problems faced by film productions, and after careful analysis chose one of the most challenging: *automate the setup for realistic facial animation of 3D characters*. Facial animation is currently in great demand, because the entertainment industry requires high quality results ([Borshukov et al. 2006]), and has been an area of intensive research for the last three decades. Facial animation is crucial to convey the feelings of each character, so automating the character setup has the greatest impact on the final production output, comparing it with other unsolved problems. After the initial research stages to determine the viability of the project were completed with success, we focused on creating a prototype of the automation system to demo animation companies and professionals and get feedback and validation. The companies have showed great enthusiasm for the results and, more importantly, for the potential of the technology. From the start, we focused on finding a solution to current rigging difficulties that was simple in design, but still able to solve a hard problem.

To achieve the cinematographic quality we have, we had to consider lots of little details [Richie et al. 2005] to make a versatile system that can be easily integrated into different animation pipelines and work well with very dissimilar characters. Our work differs from previous morphing [Blanz and Vetter 1999] and retargeting techniques [Noh and Neumann 2001], because their work was oriented towards transferring animations, while ours aims to transfer the complete facial rig, in addition to animations. Automatically transferring the rig gives artists complete freedom to manipulate the characters: they can create new animations and not be limited by pregenerated ones.

Facial animation presents many difficulties (time, cost and complexity constraints) that limit its adoption and usefulness in different situations. [Pighin et al. 2006] discuss the research efforts and main challenges faced by some blockbuster films, and emphasize that facial puppetering and the use of nonlinear rigs are still unexplored issues. Generating realistic face movements is hard, because even with current 3D software, animators cannot capture and control every detail of the face. To obtain the desired realism, traditional animation pipelines have each character separately *rigged by hand*, a very labor-intensive and time-consuming task. The character rigging process is analogous to setting up the strings that control a puppet, which in the hands of an experienced digital artist comes to life. Finding a technique that provides accurate and fast rigging remains a challenge.

what if... we want to use the rig created
for one character in other characters?

To create a **portable character rigging system** capable of synthesizing the subtle dynamics of facial expressions expected by human observers, we need to solve several *artistic* and *technological* problems. Artistically, we need to have a clear knowledge of the human familiarity and sensitivity to facial appearance, and the multitude of subtle expressive variations on the face. On the technology side, first, we have to reduce the amount of laborious work needed to create and optimize each facial rig. Second, the weight distribution used on one character will not work for others. Last, any minor artistic modification that causes the model to change in geometry (a smaller nose, a larger eyebrow) or resolution (more resolution around the eyes) leads to the restarting of most of the rigging process from scratch. These issues make it impossible to reuse the same rig in different face models; facial animation becomes a bottleneck in any CG production.

1.2 Our Solution

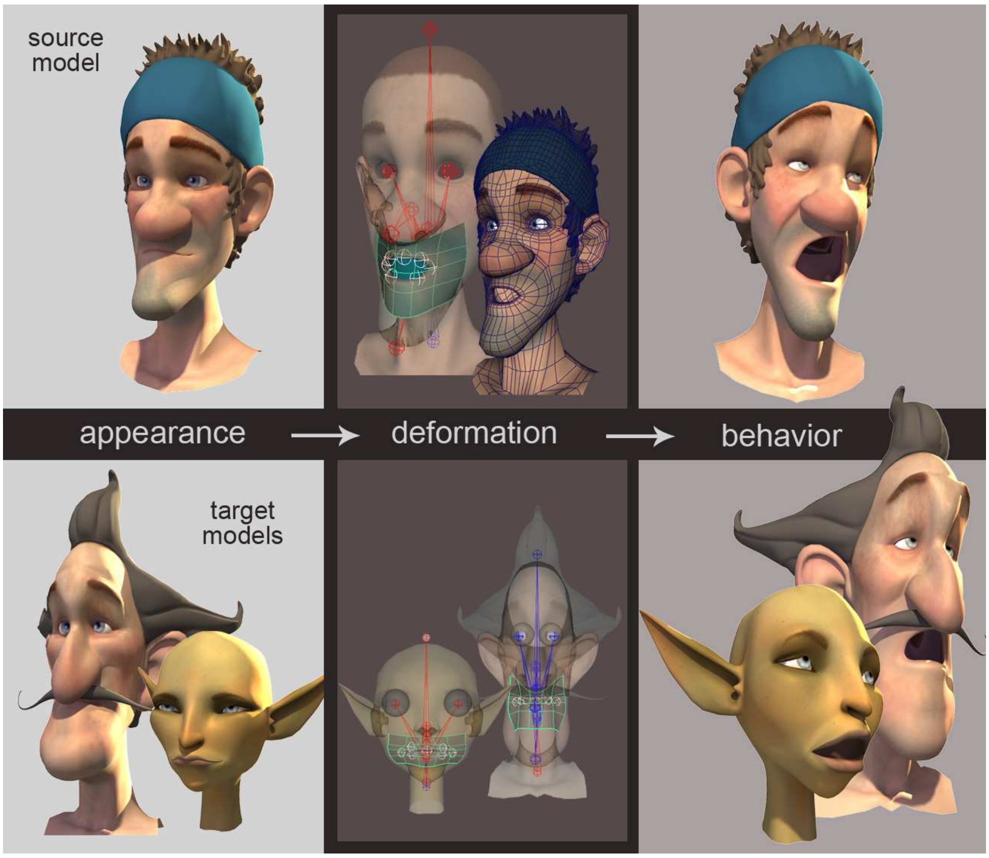


Figure 1.2: Overview: define the source and target models; adapt the source model geometry to fit the target; transfer attributes and shapes; bind the influence objects and skeleton to the target. The result is a model ready to be animated. (Copyright 2005 Dygrafilms)

We narrow our research to the rigging of 3D animation characters for films and videogames, solving the synchronization and realism problems, the reusability of facial components, together with the real time response. However, the results from our project can benefit other industries as well.

We propose a system that *given a face model, analyzes it and creates a rig ready to be animated.* Our solution starts with two 3D face models: the source model, which is rigged and includes a set of attributes (skeleton, influence objects, shapes and animation scripts), and the target model, which doesn't have a character rig associated to it (see figure 1.2). Then, the system automatically transfers the rig and animations from the source to a target model, even if they have different geometries or descriptors (one can be defined as a polygonal mesh and the other as a subdivision surface). The method follows the principals of a non-linear warp transform [Bookstein 1989], and uses facial features landmarks, resulting in an efficient deformation technique. Our solution can build models with underlying anatomical behavior, skin, muscle and skeleton, for human, cartoon or fantastic creature heads. It allows autonomous and user controlled facial features to move naturally, smoothly and realistically.

The system enables artists to create more lifelike characters in less time; it is 90-99 percent faster than traditional manual rigging. The tests showed that artists can complete in one hour, tasks that before took them one or two weeks of work; something like changing the weights, modifying an animation control position or transferring animations between characters can be achieved "instantly".

1.3 Contribution

Current rigging techniques are slow and expensive, because they rely on traditional hand setup. Alternative systems like Eyetronics[1], Universal Capture [Borshukov et al. 2005] and Image Metrics[2], combine different technologies, like image analysis and optical flow, to achieve amazing results, but still require too much time to setup. Also, applying changes and propagating the modifications to different characters is hard and needs cleaning up afterwards, making these approaches unsuitable to be used in films and videogames with lots of characters, at least in a short term.

A major benefit of our solution is that the system is very flexible: the deformation and transfer methods are general and can be used with any type of facial rig setup and 3D model topology. Artists can define their own rig and then quickly apply it to different models, even with disparate proportions and appearance (human, cartoon or fantastic). This allows easy integration in different animation pipelines and reusability in different projects. The key contributions of this dissertation to the field of computer graphics and the entertainment industry are:

- a **generic geometric deformation method**: based on facial features landmarks, deforms and reshapes the source face model surface (polygonal mesh, NURBS, subdivison) into the target surface;
- a **facial rig transfer method**: based on the generic deformation method, relocates and adapts all attributes (texture, weights, joints, influence objects) of the source model rig into the target model. The method allows reusing a rig created for one model in different characters;

[1] http://www.eyetronics.com

[2] http://www.imagemetrics.net

- an **expression transfer method**: based on different key poses of the source model, computes the same poses in the target model, but adjusted to its shape and proportions. These poses are translated into shapes and can be used for blend shape animation;

- an **animation transfer method**: adapts animation curves from the source model to different characters, adjusting the scale, rotate and translate value to the shape and proportions of the target model.

As a result, film and videogames studios will be able to animate characters that before were devoid of expression, because of the time it takes to manually create a rig. Suddenly, all those secondary characters can come to life with the help of our technology.

1.4 Outline

The remaining chapters of the dissertation are organized as follows:

Chapter 2 describes the use of facial animation in the context of visual communication, focused on the entertainment industry. It details the character rigging process and the different components that constitute a facial rig. The chapter ends with a description of the role of the character Technical Director.

Chapter 3 discusses different methods related to facial synthesis: physically-based, geometric deformation, keyframe animation, performance-driven and retargeting. It briefly describes facial parametrization, MPEG-4 standard and Facial Action Coding System (FACS). Last, the chapter lists existing animation applications.

Chapter 4 describes our solution for reusing the same facial rig in different face models. It describes the algorithms we developed and the system architecture.

Chapter 5 shows the results and data obtained from tests with several film and videogame companies. It discusses the visual and timing validations performed, comparing the automatic output of our system with the results manually created by an artist. It also summarizes our conversations with people from the entertainment industry.

Chapter 6 discusses our work, its implications and future trends.

Chapter 2

Application Domain

The face plays an important role in verbal and non-verbal communication. Everyday, we see many faces and interact with them in a variety of ways: talking, listening, looking, making expressions. Currently, virtual characters are used in many different situations: medicine, broadcasting, films, videogames, criminology, virtual worlds and others. This chapter describes the application of facial animation related to visual communication, focusing on films and videogames. It discusses the current needs of the entertainment industry, including the main issues associated with the setup of 3D characters. Then, it describes the rigging process and a typical character animation pipeline, to present a complete view of a facial animation production process. Finally, it outlines the roles of the artists, software engineers and the character Technical Director, and the relationships between them. After reading this chapter, you should have an understanding of the complexity involved in the rigging process and the reasons why facial animation remains a challenge.

2.1 Entertainment Industry

Facial animation is still an immature and unconsolidated field, with no defined methodology or standard for character rigging. The increasing number of completely computer generated (CG) films (*Toy Story (1995, 1999), Monsters Inc. (2001), Shrek (2001, 2004), Finding Nemo (2003), The Incredibles (2004), Cars (2006), Over the Hedge (2006)*), coupled with the requirement of more complex characters, has generated a growing demand for new and sophisticated 3D tools. Artists need intuitive and efficient animation tools that help them automate the character setup process and allow rig portability between characters.

Within the entertainment industry, the applications can be divided into *interactive systems* and *off-line systems*. Interactive systems, like dialog-based interfaces, virtual worlds, videogames and conversational agent systems, require real-time animations and limited realism. Off-line systems, mainly used for feature films, require high realism and accuracy to reinforce the spectators “suspension of disbelief”. Next, we describe different fields where the face plays a key role to convey information and emotion, or to interact with a user.

2.1.1 Films

In **Films**, the face of the character plays the most important role in communicating with the audience and getting it involved in the story. The audience quality and realism expectations drive the development of technology for films. Facial expressions enhance the immersive experience of the spectator, so any inconsistency in appearance or non-realistic expression can ruin the atmosphere. To create believable characters, it is imperative that digital artists have artistic freedom in modeling and animating the face models, independent of the film style: animation films with human (*The Polar Express (2004)*) or cartoon look (*Monster House (2006)*), fantasy films (*The Lord of the Rings: The Return of the King (2003)*) or live-action films (*The Matrix Revolutions (2003)*). Characters can be animated using two techniques: performance-driven or keyframe animation. The film industry needs tools that help the artists create believable animations with reduced effort and improve the work-flow between modeling and animation.

2.1.2 Videogames

Facial animation plays an ever increasing role in **Videogames**. At first, it was applied only in cut-scenes using the same techniques as the cartoon animation films (*The Curse of Monkey Island (1997)*). But, the biggest priority in videogames is interactivity, so it is a requirement in current releases that the animations run in real-time. Today, it is possible to achieve cinematic quality because of the improvements in hardware (more CPU power) and software (development of specialized algorithms), leading to the replacement of cut-scenes by scripted animations (*Need for Speed Carbon (2006)*, *FIFA 07 (2007)*). Also, high quality definition of the facial expression details can be obtained with the help of motion capture (*Tiger Woods PGA Tour ’07 (2007)*). The videogame industry wants technology that speeds the rigging process and generates results that can be easily integrated in the game engine. A possible solution can be a generic facial rig that is adapted in real-time to different characters. The facial animations created for the generic base character will suite all the other game characters, reducing the artistic effort dramatically.

2.1.3 Other Interactive Systems

While we focus our work on solving the needs of the film and videogame industries, there are other fields where our solution can be beneficial.

In **Broadcasting**, facial animation can be seen in programs that involve virtual characters: news anchors, weather people or even comedians. These are classical examples of talking heads, where the main problem is speech synchronization [Waters and Levergood 1994; Kohnert and Semwal 2006]. Prerendered facial animations, including predefined phonemes and expressions, are stored in a database and played back based on automatic text-to-speech or prerecorded audio synthesis. The animations are usually of lower quality than the ones available in films and videogames, so any application that simplifies the production and reduces work can have a direct impact in the final results.

Dialogue-Based Systems allow users to interact with question-answer applications, such as database or expert systems, through natural spoken language [Fraser 1997; McTear 2002]. By using a face as the interface, the system can enable casual users to interact in a natural way with complex applications. Animation should adequately respond the users' input and create the appropriate facial movements, or else they can become distracted or repulsed, rejecting the conversation. But current research is more concerned with solving the natural spoken language communication problems, than obtaining believable character movements. Today, conversational virtual characters are used in diverse systems: online shopping, where they assist users find products based on their personal patterns and constraints; customer service or help desk; and others [Chai et al. 2001; Lester et al. 2004].

In **Virtual Worlds**, the user inhabits a computer simulated environment and interacts, via avatars, with other avatars or non-playing characters [1]. An avatar is a virtual character (2D or 3D depending on the environment) or a text construct (like the ones in early MUDs - Multi-User Dungeon, Domain or Dimension) that represents the user in a virtual world. In the past few years many virtual worlds have been developed, either as social spaces or videogames: *Ultima Online* (1997), *EverQuest* (1999), *The Sims Online* (2002), *World of Warcraft* (2004), *Guild Wars* (2005). The biggest success is *Second Life* (2003), which gained huge international attention by late 2006; sometimes it is called a videogame, but as it doesn't have points, scores, winners or losers, it can be defined as a huge virtual terrain where characters coexist and avatars can mature with time [Pro 2006]. As avatars represent real persons, the visual appearance plays an important identity role, which increases the need for a flexible system capable of adapting the facial animations to a wide variety of model geometries. A major problem with these distributed systems is how well they can support real-time facial animations. In contrast to virtual characters for broadcasting where animations are prerendered, stored and reproduced from a database, faces in virtual worlds have to respond in real-time, to avoid interaction delays or unnatural communication between characters.

[1] Non-playing character is a character in a role-playing or computer game, whose actions are not controlled by a human player.

2.2 Character Rigging

"*Rigging* is the process of taking a static, inanimate computer model and transforming it into a character that an animator can edit frame-by-frame to create motion" [Falk et al. 2004]. The result is a *rig* that can be manipulated by a set of controls like a virtual puppet [Schleifer 2002] or by motion capture data. Creating the character rig is a very complex, time consuming and labor intensive task. Today, there in no defined standard methodology for rigging a face. Studios continue to redefine the techniques, processes, technologies and production pipelines to efficiently create films and videogames.

Rigging links the modeling and animation processes within a traditional animation pipeline (figure 2.1). It is very common to remodel some parts of the character after the rigging has started, to achieve better deformations or because the topology needs more resolution. This causes the restart of the rigging process, which delays the production. Sometimes it is necessary to rework the rig and the controls after animators have already started the animation process, because the rig doesn't perform the desired behavior.

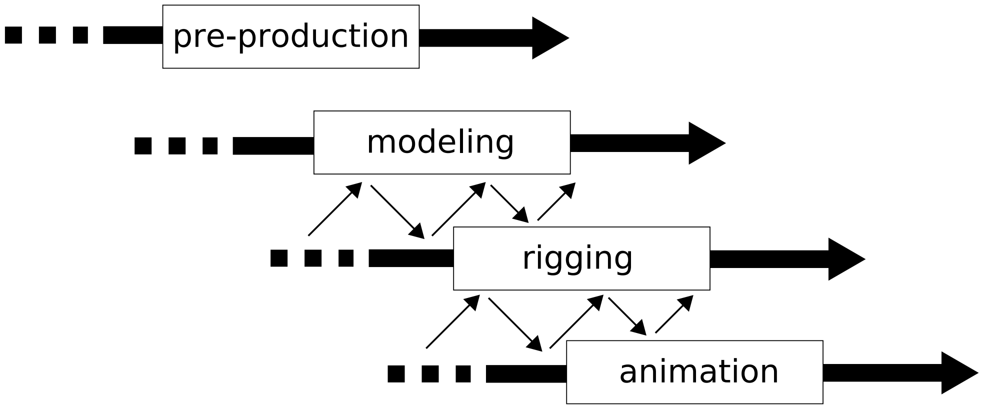


Figure 2.1: A traditional animation pipeline where the modeling, rigging and animation processes work in parallel. Any change in the character model topology will cause the rigging process to restart. Any undesired behavior of an animation control will lead to a readjustment of the rig.

Before starting the rigging process, the rigger (person responsible for the character rigging in the production pipeline) discusses with the animators the main features a character should have. Usually, many of the features requested by the animators are not included in the existing 3D software packages, so custom based tools are created to make the rigging process more robust and production friendly. Every rig must follow these rules:

- animation controls should be easy to use and understand;
- animation controls should work the way animators expect (principle of least surprise);
- the rig should allow quick updates, if an undesired behavior is found;

- the rig should carefully define constraints to avoid creating "impossible" or strange movements.

After the definition of the rig features and before start rigging, it is important to check that the model's mesh is stable, clean and properly connected. The first step of the traditional rigging process consists of defining the skeletal structure and placing the joints. Second, the rigger aligns the joints to guarantee correct rotation during motion. Third, the rigger skins the character using a smooth algorithm and paints the skin weights. Adjusting the weights on the skin is normally an iterative process, which might involve editing the skeleton and refining the model's deformable objects to get the right skin deformation effect. The final step involves testing the response of the rig to the manipulation of the animation controls [Richie et al. 2005].

An experienced artist can take from 1 to 4 weeks to rig a character, depending on the complexity of the rig. The slowest task is painting the weights. To speed up the process it is critical to automate common tasks. Chapter 4 describes our solution that dramatically decreases the time needed to go from modeling to animation. It allows transferring a complete facial rig between different characters and avoids the need for painting the weights.

2.2.1 Rigging a Face Model

Finding the optimal solution to create a facial rig depends on several constraints: time to develop the rig, budget, artists' experience, expected rig performance and actions, and others. The three most common approaches to create a rig are based on: *blend shapes*, *bones* or a combination of both.

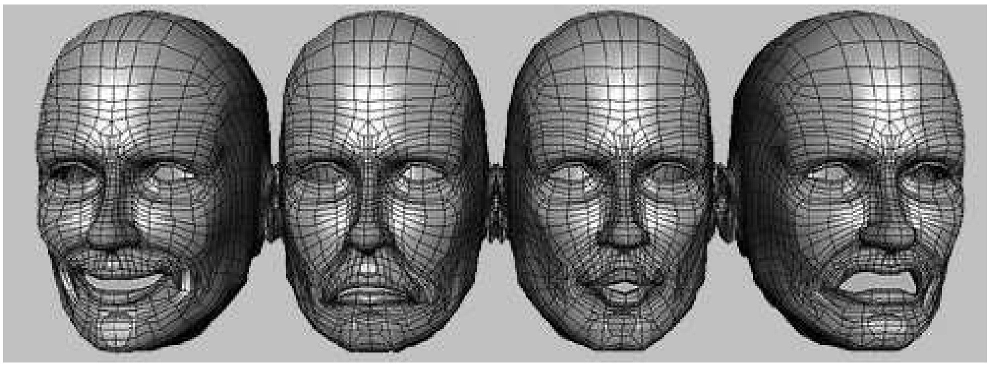


Figure 2.2: Blend shapes that represent the basic expressions: happy, sad, angry and surprise. (Copyright 2004 New Riders Publishing)

A rig based on *blend shapes* [Maraffi 2003] consists on sculpting facial poses into several meshes of the same topology, where each new mesh is called a shape. Animation of the character is generated by morphing several shapes: interpolating between the open mouth shape and the neutral position, makes the character open or close the mouth (figure 2.2). Each region of the face can define many localized shapes, which enable independent control of that region and allow mixing multiple shapes to create a wide variety of poses during animation [Sloan et al. 2001].

A face model can be completely rigged using only blend shapes, but it is necessary to generate a large number of shapes to provide control over every region of the face. In the film *The Lord of the Rings: The Two Towers* (2002), Gollum's facial animations required 675 blend shapes [Fordham 2003]. However it is possible to create complex behaviors and realistic facial expressions using a smaller number of shapes [Ekman and Friesen 1975], or by combining them with other animation techniques. A major problem of this approach is to correctly mix between shapes and still obtain accurate animations. Blending small sections of the face works well, but it is harder for animators to obtain appealing results for bigger regions. It takes time and considerable skill to manually define all the shapes for just one model; this process needs to be repeated for every character that is going to be animated. To overcome this, we propose an automatic method that transfers all the shapes to each character in less than a second, while maintaining the same level of deformation and expressiveness.

A *bone-driven* rig is based on a highly articulated facial skeleton structure. This approach generates smoother movements compared to blend shapes, but needs more preparation to get the desired results. As each vertex is only animated by the bones around it, much more planning must go into rigging process of each model [Ward 2004]. Bone-driven rigs are the usual choice of videogame productions. A typical setup contains approximately 50 bones, divided into: 12 bones at the forehead, 4 per eye, 4 controlling the nose, 4 at the cheek, 8 in the upper lip, 11 in the lower lip and chin and 4 bones for the tongue (figure 2.3). This skeletal structure allows the representation of basic expressions like blinking, smiling, eye tracking and phonemes. When using motion capture to drive animation based on the performance of an actor, each bone of the rig can represent a motion sensor placed on the face.

Combining blend shapes and bones: best of both worlds?

In film and videogame productions the main concerns are budget and time. It is crucial to pick the rigging technique that best suits the projects: a rig with too few joints or blend shapes makes the face look stiff and hard to control. A blend shapes rig has a simpler initial setup and is always limited by the number of available shapes: if the face topology is changed, all existing shapes have to be redone. On the other hand, a bone-driven rig has a tricky initial setup, which needs to guarantee that all desired expressions are possible, but needs no further work when the topology of the character is modified. Blend shapes allow artists to define each pose perfectly, because the shapes are created by hand sculpting each vertex of the mesh. This is not possible with a bone-driven rig, as each vertex position is influenced by the attached bones.

A typical choice is to combine blend shapes with a skeletal approach, which provides a rig with the flexibility and smoothness of a bone-driven system and the expressiveness of blend shapes [Lewis et al. 2000].

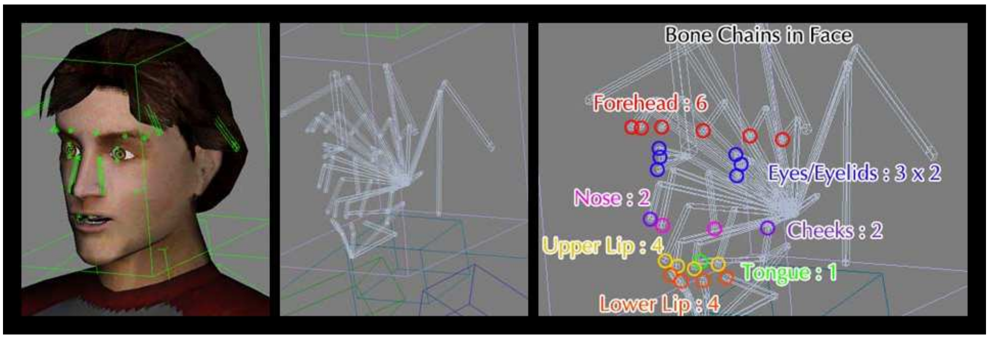


Figure 2.3: A highly articulated facial skeleton model prepared to be used with the Unreal game engine. The skeleton is composed of approximately 50 bones: textured 3D models with joints and a bounding box for collision (left); skeletal structure and collision bounding box (middle); detail of the terminating ends of the joint chains and the number of joints of each chain (right). (Copyright 2001-2007 Epic Games)

Additional layers of deformation can also be added to the rig, independent of the approach used to create it. These deformers are commonly denominated as influence objects or deformable objects, and can emphasize different facial features, like wrinkles, in areas of the face where neither bones nor shapes produce the desired results. For example, NURBS curves can be added in the cheek to simulate a fleshy and elastic skin.

Starting with a character face model represented by a polygonal mesh, a NURBS surface or a subdivision surface, a typical rig includes:

1. **Skeleton**: represented by the underlining joint and bone hierarchies. Usually, every skeleton has several parent and child joints and one root joint. The skeleton enables the creation of hierarchical and articulated deformation effects.

2. **Deformable objects**: any object that can be deformed, like NURBS surfaces, NURBS curves, polygonal meshes, lattice deformers, and others. They are bound to the skeleton and are organized in hierarchical groups based on how the face model will be animated. They help change the shape of the character. The deformable objects add extra control and realism to the animations: they can represent the geometry of a muscle and simulate its behavior.

3. **Skinning:** the process of binding deformable objects to a skeleton, so they can later be deformed by the skeleton. There are different skinning techniques, like smooth or rigid skinning [Larboulette et al. 2005; Yang and Zhang 2006]. The most important task during this process is the weight definition [Wang and Phillips 2002]. The weight is the degree of influence of a vertex during deformation. Each joint and deformable object has its own weight distribution map, which defines the amount of influence they will exert on the model during animation [Mohr and Gleicher 2003].

4. **Constraints**: restrictions of position, orientation or scale of an object. They can be defined as absolute values or relative to other objects.
5. **Shapes**: deformation fields defined on each vertex of the character geometry. Shapes help define a set of poses or facial expressions. The interpolation of different shapes results in the animation of a face model.

2.3 Artists, Software Engineers and the Technical Director

Artists and Software Engineers have very distinctive activities within a facial animation domain. *Artists* focus in producing high quality content, using 3D software available in the market and in-house tools. Many times, when a special effect needs to be accomplished and a deadline approaches, artists apply *ad hoc* solutions that are forgotten once the project is over. *Software Engineers* are more concerned with the technical aspects, working to develop solutions that can solve a wide variety of problems, based on the artists requirements. The role of the *character Technical Director (TD)* becomes crucial in a production, because it creates the communication channel between Artists and Software Engineers [Falk et al. 2004]. The TD's job demands good artistic and technological understanding, including mathematical and programming background, to be able to develop an architecture that links all the departments together. The responsibilities of a TD are:

Design: design the technical solutions required to fulfill the Director's creative vision of the project. It includes choosing the rigging technology and defining the rigging standard.

Development: integrate the rigging system in the animation pipeline and implement all the application interfaces that allow artists to control the characters.

Support: help the animators understand the rig controls and the character behavior.

Chapter 3

State of the Art

Facial animation remains a challenge since the 1970s. The most ambitious research goal is to perform facial synthesis in real time with cinematographic quality. Several approaches emerged, all dealing with the complexity of the human face, to obtain the proper geometry to generate motion: physically-based, geometric deformation, performance-driven, keyframe interpolation and retargeting. The most important feature of each face, and also the main difficulty during animation, is its uniqueness. This chapter describes the major problems related to facial rigging and animation encountered by the entertainment industry, and their current partial solutions. Next, comes an exhaustive analysis of the published literature and previous work. Last, it investigates the different applications and solutions available in the market. After reading this chapter, you should have an understanding of the underlying work that goes into building 3D facial animation tools.

3.1 Body Animation and Facial Animation

Facial animation is greatly influenced by all the research done for body animation. Facial and body animation share a number of similar research problems like muscle deformation, soft tissue deformation and animation retargeting. However, they differ in many significative ways. First, the number of muscles required to create a facial expression is greater than the muscles required to create a pose for a specific part of the body. Second, the head cannot be animated with a single joint like most parts of the body. Third, the soft tissue simulation needs to be more realistic to capture all the subtleties of the facial expressions. Fourth, it is near impossible for an animator to achieve realistic results, unable to overcome the expectations of human observers, who are experts at watching faces. It is easy for results to fall into the creepy appearance category known as the Uncanny Valley (figure 3.1), first introduced by [Mori 1982; Hanson et al. 2005]. Facial animation requires more work to obtain lifelike response than body animation: it needs to use more variety of deformers to simulate the soft tissue and muscles, assign a greater number of joints to influence each region of the face and implement a higher number of controls to manipulate the hole head structure and secondary face regions. ‘Every face is unique - in its looks, its shape and proportions, as well as in the specific ways of articulation and expression that are so characteristic for an individual’ [Kahler 2003]. The ultimate goal is to give artist a very sophisticated *facial rig* capable of adapting the face model to the uniqueness of each character.

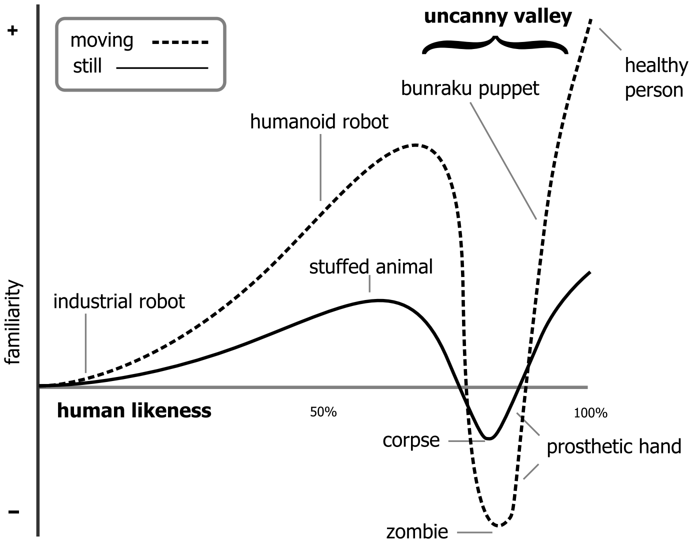


Figure 3.1: The Uncanny Valley hypothesis was introduced in a study of the emotional response of humans to robots and other non-human entities: as the appearance becomes more humanlike the response of a human is increasingly positive and empathic, until a certain point where the the response quickly becomes strongly repulsive; then, as the appearance and motion of the entity evolves to be (almost) undistinguishable from a human being, the emotional response and familiarity approaches human to human empathy levels. This gap of repulsive response provoked by an “almost human” entity is called the Uncanny Valley. (Original graph by Dr. Mori 1982)

3.2 Background

There are many approaches and a even greater number of problems related to modeling and animating faces. To keep the size of this section manageable and due to the extensive available bibliography, the material is restricted to topics that present an overview and the historical background of this field. Research that is relevant to specific sub-problems related to this dissertation, like rigging, is discussed in the corresponding sections.

Facial Animation is based on ideas pioneered by [Parke 1972] (for a detailed review see [Parke and Waters 1996; Haber et al. 2004]). Early work includes [Platt and Badler 1981], who built a face model capable of simulating and animating nonrigid objects using masses and springs, and classified the units of motion using Facial Action Coding System (FACS) [Ekman and Friesen 1978]. Since Parke's first forays into the field, many approaches have emerged that can be classified into 3D geometric manipulation or 2D image manipulation. Beyond this classification, it is difficult to identify which group each method belongs to, because the boundaries between technologies are not clearly defined. Geometric manipulation includes techniques like keyframe interpolation, parameterization, physically-based and others; image manipulation includes morphing and blend shaping. An excellent survey of these efforts can be found in [Noh and Neumann 1998; Pighin and Lewis 2006]. Face analysis and understanding is another area that influences current trends of facial synthesis. [Zhao et al. 2003] presents a detailed survey on facial recognition that can provide a different perspective and complement current research.

Recent approaches, like performance-driven, often combine several methods to produce better results. For instance, geometric methods can be used for modeling purposes, while image-based techniques can be used for animation transfer. The following sections describe the most common approaches used for facial modeling and animation.

3.2.1 Keyframe Interpolation

Keyframe interpolation is the easiest and also the oldest completely geometric approach. It consists on specifying complete face models for a given set of points in time, called keyframes, key poses or key expressions. Usually, these keyframes are essential poses required to define a movement, like in traditional hand-drawing pose-to-pose animations. The face models for in-between frames are generated by interpolation. If not enough keyframes are defined, the resulting animation is not very smooth, specially for complex models, where creating the in-betweens poses is too unpredictable. The path of action will usually be incorrect and objects will intersect each other [Lasseter 1987]. This method has several drawbacks: requires the complete face model for each key frame; generates large amount of data; requires maintaining an identical topology for all keyframes; and modeling is laborious.

Convincingly combining separate expressions into one, like crying while sleeping, is very complicated; doing it automatically remains an unsolved problem. Different interpolation methods try to solve it: linear interpolation, used for blending and morphing; non-linear interpolation, useful for displaying dynamics, like acceleration or speed; and segmental interpolation, where different interpolation values and treatment are defined for independent regions of the face, like the eyes, mouth or nose [Möller and Haines 1999]. Keyframe interpolation was the technique of choice in the 1970s, but current demands for realism make it unsuitable. Now, keyframe is used in combination with other animation methods, like performance-driven techniques

[Igarashi et al. 2005]: instead of having to position every vertex of the model for each keyframe, the animator only needs to set the values of the animation controls.

3.2.2 Geometric Deformation

Geometric deformation methods consist on using an object to modify another more complex object, by presenting a easier or simpler control interface. They are efficient for modeling and animating deformable objects, because they provide a high level of geometric control over the deformation.

A typical geometric approach is free-form deformation (FFD). It was first introduced by [Sederberg and Parry 1986] and uses a lattice to control the 3D model deformation [Cohen et al. 2001; Singh and Fiume 1998]. [Chadwick et al. 1989] used FFD for layered construction of flexible animated characters, which didn't require setting the corresponding features on the geometries. [Coquillart 1990] and [MacCracken and Joy 1996] extended FFD to support more general lattices, but lost some of the flexibility, stability and control [Sederberg and Parry 1986], while [Hsu et al. 1992] developed a version of FFD that allowed direct manipulation of an object.

Space warping methods are independent of the object representation, making them well suited for editing meshes, voxel volumes, scattered point data, and others. [Lazarus et al. 1994] proposed an intuitive method that used axis instead of a lattice, which was independent of the object and valid for any axis. [Singh and Fiume 1998] took curve based deformations further, with a method that allowed deforming a 3D model by manipulating parametric NURBS curves. This method can be used for high level control to simulate wrinkles or to drive animations.

Other geometric deformation methods related to character animation where introduced: [Turner and Thalmann 1993] defined an elastic skin model; [Singh et al. 1995] used implicit functions to simulate skin behavior; and [Wu et al. 1996] studied skin wrinkling. [Lewis et al. 2000] used radial basis functions to develop a pose space deformation technique for facial skin and skeleton-driven animation. [Joshi et al. 2003] proposed an automatic physically motivated segmentation that learned the controls and parameters directly from the set of blend shapes to create facial animation. [Chang and Jenkins 2006] presented a method for articulating and posing meshes, to assist the users control a specific rig based on 2D sketching as an interface for 3D mesh animation. These methods provide artists with easy controls to generate animations, but automating these procedures still requires considerable effort.

3.2.3 Physically-based

Physically-based methods simulate the visco-elastic properties of facial skin and muscles to generate expressions and animations, and to build facial models. The dominant technologies used in physics-based models are mass-springs and finite element algorithms, which can be used separately or combined, depending on the intended simulation.

[Platt and Badler 1981] developed the earliest work towards 3D facial animation using a muscle-based model. They represented muscle fibres with a mass-spring model that simulated the forces generated by muscles, and used FACS encoding. [Waters 1987] defined three different muscles types based on the nature of their actions: linear, sheet and sphincter. The muscles were embedded in the surface and were independent of the bone structure, so it was possible to transfer them to face models with different topologies. His model represented the skin as a geometric surface with no underlying structure. Facial expressions were obtained by simple geometric deformations controllable by a limited number of parameters, but failed to reproduce subtle skin movements. [Terzopoulos and Waters 1990] extended previous work by creating a model with physically-based tissue and muscles based on the face anatomy, which allowed for more realistic surface deformations. Today, many physics-based models still follow Water's basic principles. Other early work includes [Magnenat-Thalmann et al. 1988; Kalra et al. 1992].

[Lee et al. 1995] constructed an anatomically motivated facial model based on scanned data, and used a multiple-layer dynamic skin and muscle model, together with a spring system, to deform the face surface; the model was later driven by muscle contraction. Their approach made it hard to define accurate muscle parameters, due to the complexity of human muscles. [Waters and Frisbie 1995] created a muscle model for speech animation, emphasizing that it is more productive to focus on modeling muscles instead of surfaces, as muscles are the ones that drive facial animation. [Essa and Basu 1996; Essa and Pentland 1997] described a system for observing facial motion by using an optical flow method together with geometric, physical and motion-based dynamic models, to describe the facial structure. As a result, they obtained a parametric representation of each muscle action group and an accurate estimation of the facial motion. The tool extracted facial modeling and animation parameters, head tracking, and real-time interactive facial animation, by measuring information from a video. The limitation of this approach is that it considers a two dimensional finite element model for the skin, along with a simple muscle model for actuation.

[Basu et al. 1998] described a method for tracking and reconstructing 3D human lip motion from a video stream. They built a physically-based 3D model of the lips and trained it to understand the motion of the lips. It used PCA (Principal Component Analysis) to reduce the degrees of freedom to ten, because they only had a small number of training observations. Consequentially, the method was able to track lip motion automatically by matching the parameters of the model with the motion data from a video. [Choe and Ko 2001; Choe et al. 2001] presented a system to synthesize facial expressions based on performer captured data. They used a 2D linear quasi-static finite element model of the skin surface to simulate the deformation caused by the muscles, and the actuation of expression muscles that applied forces to the skin surface, to control facial expressions. Artists sculpted the initial state of the muscle actuation values. This lead to a lack of anatomical accuracy that could produce unnatural artifacts, requiring repeated resculpting of the base elements to produce decent results.

A promising anatomical model was described by [Kahler et al. 2001; Kahler et al. 2002] that included different types of muscles, and the effects of bulging and intertwining muscle fibres. The skin deformation was simulated by the influence of the muscle contraction, which used a mass-spring system connected to the skull, muscle and skin layers. The result was an head model suitable for real-time physics-based facial animation. [sen Tang et al. 2004] described a NURBS muscle-based system to simulate 3D facial expressions and talking animations. The NURBS curves represented the different face muscles that were positioned based on anatomical knowledge. Muscle deformation was achieved by manipulating the different control points of the curve and changing the weight distribution.

Recently, [Sifakis et al. 2005] developed one of the latest and more advanced muscle based models, which used a non-linear finite element implementation to determine accurate muscle action, captured from motion of sparse facial markers. An interesting feature was that the muscle actions can interact with the environment, so the muscle forces can be combined with external forces, like the impact of a colliding object, and modify the final appearance of the face. The method showed the success of performance-driven animation, but it is not clear if it can handle anatomically inaccurate models. Also, some improvements can be done to increase realism, like including a skeleton structure and more accurate lip deformation data.

3.2.4 Performance-driven

Facial *performance-driven* (or motion capture) technology allows capturing the complex movements of a human face. These movements are recorded as animation data, which are then mapped to a 3D model and reproduced to animate synthetic characters. This concept is similar to an older animation technique called rotoscopy, where animators trace, frame by frame, live action film movements and use it as a guide for hand drawing animation [Laybourne 1998].

Performance-driven methods use both image and geometry manipulation techniques. Early attempts go back to [Waters 1987; Lee et al. 1993; Lee et al. 1995], where it was possible to digitize facial geometry through the use of scanning range sensors, create a structure facial mesh and animate it through the dynamic simulation of facial tissues and muscles. These advancements led to further research related to motion estimation from video. The first approach was introduced by [Williams 1990], who presented a method for tracking and acquiring facial expressions of human faces from a video stream and applied the data extracted to computer generated faces. [Guenter et al. 1998] extended previous work to recover data from multiple video streams and for capturing 3D geometry, color and shading information of human facial expressions. Most methods track facial markers placed on an actor, recover the 2D or 3D position of the markers, and animate a 3D mesh using the captured data. Usually, these methods need that the shape of the actor closely resembles the target face, otherwise retargeting [Fidaleo et al. 2000; Chai et al. 2003] must be used to correctly map the movements between source and target data. Marker-based motion capture systems support between 30-160 markers on the face, resulting in a very sparse representation of the movements. While this sparse information works well for capturing the motion of rigid objects, it is not very effective for capturing the subtleties of expressive deformable surfaces, like the face. The limitations of marker-based systems have encouraged the development of a variety of markless motion capture systems [Blanz et al. 2003] and facial feature tracking from video using complex models [Reveret and Essa 2001; Hyneman et al. 2005].

Many performance-driven techniques have emerged, leading to the development of specialized applications, including: optical flow, which uses two or more cameras to triangulate between natural surface features and reconstruct surface geometry, but results might need extensive manual cleaning, because it is difficult to capture tiny surfaces features [Borshukov et al. 2003]; and image analysis, which uses one ore more camera views to analyze a scene and identify the character poses previously stored in the system. The derived poses are used to control the synthetic character, but these poses are limited to predefined information as the system doesn't capture the performer's actions [Pavey 2007]. Motion capture technology is changing dramatically and new methods continue to appear [Zhang et al. 2004; Borshukov et al. 2006]. Recent advances show that performance-driven techniques are being used in combination with other animation approaches, like blend shape [Deng et al. 2006], providing artists with the possibility to locally control the animations and generate more convincing results.

Today, nothing is more natural than the actual expressions created by real people. If such performance is accurately captured and reproduced the results can be very lifelike. Thus, methods based on performance-driven data can generate realistic facial motion, but continue to be expensive to use and more suited for human beings than imaginary characters. Some other limitations remain unsolved, like how to accurately capture the inside of the lips.

Last, comparing the films *The Polar Express (2004)*, completely generated with motion capture, and *The Incredibles (2004)*, which combined motion capture with traditional animation techniques, we can appreciate that the combination of keyframe animation and motion capture produces better results, because the creative freedom to manipulate the character remains in control of the artists. Performance-driven methods will continue to improve using machine learning or interpolation techniques, and will complement current animation and rigging techniques [Pighin and Lewis 2006].

3.2.5 Retargeting

The concept of retargeting is related to the synthesis of facial motion by reusing existing data. Consists on directly mapping motion from a source to a target model of different proportions, where the source data has to be adapted to the target model shape, making the target animatable. The name was first introduced by [Gleicher 1998], who presented a method for adapting motion capture data from one character to another, which share the same structure but might have different sizes. The method was well suited for human body structures, but wasn't prepared to capture the subtleties of facial motion because it lacked facial structure. However, this concept was earlier presented by [Lee et al. 1995], who developed a method that mapped videorecorded performance of one individual to another, generating a detailed 3D texture face mesh for the target identity. [DeCarlo et al. 1998] constructed smooth face models based on facial anthropometric techniques. Retargeting takes one step further the use of performance-driven methods, as it allows mapping the motion of a performer in characters with dissimilar shape, proportions, resolution and appearance. A wide variety of facial retargeting methods appeared based on motion data analysis or facial parametrization. These methods are classified into explicit and implicit parameterization.

Explicit parameterization relies on direct correspondence of a specific set of parameters defined in the source and target models, in order to transfer information like motion capture data between them. Usually, the method starts by defining key facial features and establishing the correspondence between source and target models. Then, the displacements vectors from the source data (i.e., position of the motion capture markers) are normalized in orientation and magnitude to match the target model's shape and proportions. Last, the remaining vertices of the target model (i.e., vertices that don't have a marker associated to it) are calculated using a scatter data interpolation method [Amidror 2002]. Different warping methods have been used as kernel functions to produce the smoothest and best approximation of the intermediate vertices [Litwinowicz and Williams 1994]. [Noh and Neumann 2001] proposed a method for cloning facial expression of an existing model on a new model. The movements of the source face were expressed as motion vectors applied to the mesh vertices and morphed using a radial basis function (RBF). The motion vectors were transferred from the source model vertices to the corresponding target model vertices. The initial correspondence was defined by manually positioning key points in both source and target models. To obtain an accurate animation in the target model, the magnitude and direction of the transferred motion vectors were properly adjusted to fit the target shape. Another approach similar in concept was introduced by [Pandzic 2003] that consisted in copying facial motion from one 3D face model to another, while preserving motion with the MPEG-4 FBA (Facial and Body Animation) encoding standard [Pandzic and Forchheimer 2002]. Facial motion was obtained by calculating the difference of the 3D vertex positions between the animated source face and the neutral source face. The facial motion was then added to the vertex positions of the target, resulting in the animated target face. Many other applications follow these approaches [Pyun et al. 2006; Fratarcangeli et al. 2007], but most of them make no attempt to analyze the appearance of the data, focusing only on transferring the motion information, which is not enough to reproduce every expression a face can accomplish. To overcome this limitation, some retargeting methods decided to include in the transfer values associated to the physical appearance of the face, to produce a more accurate deformation result. These additional parameters were defined on the mesh and on the underlying structure of the face, creating an additional layer of control: like the one described by FACS [Ekman and Friesen 1978] that related the muscle surfaces to the movements of the skin surface, based on muscle parameters; or like [Ishikawa et al. 1998], that built a complex physically-based model of the performer and estimated the muscle contractions to accurately deform the skin surface, to be able to match the data in the correspondent inner structure of the target model. [Kahler et al. 2002] differed from the previous approaches because they presented a construction and deformation method for transferring the human head anatomical structure, instead of just the motion data. Their head model included landmarks in the skin and skull and was prepared for physically-based animations. With their technique, it is possible to adapt a predefined muscle and bone structure into different models. As a result, all head models share the same set of parameters, enabling reuse of existing animation scripts.

On the other hand, *implicit parameterization* focus on analyzing the motion data by studying the relation between the different shapes of a model. A shape is a pose of a model and interpolating different shapes results in animation. This is called blend shape and is a very common approach among artists. To successfully map the animations between source and target models, it is crucial that both models have the same corresponding shapes. Then, it is a matter of transferring the weights associated to a shape along a period of time, to generate the appropriate motion in the target model. For example, the weights from a specific shape from the source model are directly applied to the corresponding shape on the target model, producing an equivalent facial pose; the weight is the amount of deformation that is applied to a specific shape. [Kouadio et al. 1998] proposed a method that captured live facial expressions from the performance of a an actor and used them to animate a synthetic character in real-time. The mapping of the motion was done using a bank of expressions that allowed generating subtle movements, which were more dependent on the characteristics of the model itself. The deformations of the model were obtained without any additional information on how the model should move. They used explicit parameterization to map the markers of the motion capture data to the target model. Instead of using scatter data interpolation to determine the position of the remaining vertices, they used a bank of 3D models of basic facial expressions of the target model. As the sparse and dense version of the model were the same, correspondence was maintained and animations retargeted. [Pighin et al. 1999; Pighin et al. 2006] presented a technique that consisted on interactively marking corresponding facial features in several photographs of a person, deforming a generic face model using radial basis function and interpolating the different poses. [Chuang and Bregler 2002] presented a method that used a combination of motion capture data and blend shape interpolation. The artist created the blend shape models, guaranteing that the shapes will nicely mix during animation, while motion capture data is used to drive the facial movements, rather than animated by hand. The drawback of the method is that modeling each shape by hand is crucial for the success of this technique, which is a very time-consuming task.

Usually, explicit parameterization is by far the most common approach used for retargeting. Choosing the appropriate parameterization is what distinguishes the different techniques and defines how useful each one is. Implicit parameterization is more intuitive for artists, because it follows the traditional way they animate.

3.3 Facial Parameterization

The parameterization process consists on defining an optimal set of parameters that can be used to control facial movements. It is a way of abstracting the shape or movements of the face model, so an artist can easily create motion: parameters become *puppeteer handles* that help animate the face. Parameters can also be used to obtain information from video analysis, to reconstruct a head or to transfer animations between different models. Using parameters instead of the complete geometry or image information, leads to number of benefits: a facial expression can be determined by a small number of parameters; artists can work on a higher level of control, instead of manipulating the geometry directly; expressions can be transferred between characters because they share the same configuration; and others. Research has shown that an ideal parameterization does not exist: it is difficult to satisfy all user demands for a broad range of facial applications. Considering the following issues helps define an efficient parameterization:

Environment: determine the purpose of the parameterization. It can be for generating animations for films, capturing data to be used in videogames, physical simulation of the head, controlling a character in real-time, tracking facial features from a video sequence, etc.;

Appearance and behavior: define the range of possible faces and expressions. Determine if the parameterization is flexible enough to deal with all the different looks: from realistic human faces to cartoon characters with exaggerated features and motion. Parameterization becomes more complex when the range of acceptable faces is wider;

Usability: define an intuitive way of controlling the parameters, making sure that it can represent all required facial configurations.

Developing an optimal parametrization is a difficult task. Once we choose the purpose of the parameterization and the key features, it is critical to evaluate if the set of parameters can perform the complete range of required facial expressions. [Parke 1974] created the first facial parametric model that allowed direct creation of facial deformation by defining *ad hoc* parameters or by deriving parameters from the structure and anatomy of the face. [Ekman and Friesen 1978] defined the Facial Action Coding System (FACS)[1] to describe and measure facial behaviors. FACS was not created for animation; it is a system for evaluating the human facial expressions that are later coded in basic facial movements, called Action Units (AUs). FACS became a popular standard used to systematically categorize the physical expressions of emotions and it is currently used by animators and psychologists. There are 66 AUs that represent contractions or relaxation of one or more muscles. For example, AU 43 represents eyes closed, which is a relaxation of the muscles Levator palpebrae superioris and Orbicularis oculi.

Today, facial animation parameterization varies greatly between applications. Isolating the appropriate parameters to control and animate the face is complex but fundamental. There are two main parameterization approaches: feature based and muscle based.

3.3.1 Feature Based Parametrization

Feature based parameterization uses the empirical study of the face to determine the set of parameters. When a new expression is required, a new parameter is added. The first approach was by [Parke 1974], who in addition to empirical studies of human faces, used traditional hand drawing animation to select the parameters. He defined two groups of facial parameters: expression parameters, which control expressions, and conformation parameters, which define the basic appearance [Parke 1982]. Motion was generated by interpolating the vertex position of the parameters over time, where each parameter specified an interpolation value between two extreme positions of the face model. Later, [Guenter et al. 1998; Guenter et al. 2005] presented a method that used a large set of feature points on the face to accurately track the three dimensional deformations of the face. The feature point was linked to the closest control point of the face geometry, so changing the feature point position caused deformation on the face model. There is also feature point implementations in the MPEG-4 standard [Koenen 2002], although it makes a distinction between facial definition parameters (FDP) and facial animation parameters (FAP), where FDP define the shape of the face and FAP control the animation. MPEG-4 does not define the mapping between the feature points and the skin geometry deformation that generates motion. This is left to be implemented by the applications.

[1] http://wwww.face-and-emotion.com

3.3.2 Muscle Based Parameterization

Muscle based parameterization became a popular method to define face deformation, because of its ability to manipulate facial geometry based on simulating the characteristics of the muscles. It uses anatomical knowledge of the face to define shape and motion. But this approach is hard to animate, because the face has many muscles that need to be controlled to obtain a realistic movement. So, it needs a higher level layer linked to the muscles, which simplifies the animation control interface. FACS is the basis for further muscle based parameterization research. [Platt and Badler 1981] introduced a method for internal representation and simulation of the face, where the muscle and skin layer were conceptually separated and the muscle didn't have an actual geometry. The skin deformation was not directly specified; it emerged from forces acting on the skin layer, applied by attached virtual muscles. For instance, wrinkles appeared automatically without explicit modeling. [Waters 1987; Waters 1989] developed a facial tissue model articulated by synthetic muscles, which provided a tool for observing, analyzing and predicting soft tissue mobility on the face. This lead to different facial animation applications, which didn't require predefining the performable actions [Waters 1992]. [Magnenat-Thalmann et al. 1988] presented a system that used Abstract Muscle Action (AMA) procedures, which defined different muscle actions to create expressions. Later, [Terzopoulos and Waters 1990] presented a method that incorporated approximation to facial tissue and a set of anatomically-motivated facial muscle actuators. It also estimated muscle contraction from video sequences of human faces performing expressive articulations. [Lee et al. 1995] extended this method by specifying the width of the muscle, allowing the muscle to follow the shape of the skull when a movement is detected. Their deformation process was able to match a generic head model to individual head geometry.

3.3.3 The MPEG-4

The MPEG-4 Facial Animation[2] specification extended previous approaches and was the first facial control parameterization to be standardized into MPEG-4 FAB (Face and Body Animation) [Pandzic and Forchheimer 2002; Koenen 2002]. The standard is used for academic research and for commercial applications. Mark Sagar pioneered the use of FACS for facial animation in the entertainment insdustry. He developed a motion capture system that conveyed and translated subtle expressions from an actor's performance to the face of the synthetic character, which was used on *King Kong* (2005), *Monster House* (2006) and other films [Sagar 2006]. The Fraunhofer HHI [3] developed an advanced video conference system using MPEG-4 that is able to encode head and shoulder image sequences at bitrates of 1 kbits/s. The 3D models are used to represent the person in the scene and the MPEG-4 specification animation

[2] http://www.cselt.it/mpeg

[3] http://www.hhi.fraunhofer.de

parameters define temporal changes of facial expressions. The range of applications that use some kind of parameterization is constantly growing and novel techniques continue to emerge. Xface [4] is a set of open source tools for creating and editing MPEG-4 and keyframe based 3D talking heads [Balci 2004].

3.4 Entertainment Industry: in-house tools

Through our contacts with major film and videogame companies, we discovered that currently there is no automated solution for rigging a facial character. Most companies develop their own in-house tools to speed up the rigging process, but still, at least 50% of the work has to be done manually. Usually, for main characters, such as *Shrek* (2001) or *King Kong* (2005), that need to perform very detailed deformation to convey subtle emotions, the rig is completely customized, and specific techniques are implemented to fulfill the Technical Director's requirements. For secondary characters, the rigging process combines scripting and hand tuning to speed up some tasks like animation transfer, but if the characters are very dissimilar the scripts cannot be reused.

An average 90 minute animated film, such as *Over the Edge* (2006), has at least 50% of animations that render the face, with an average of 40 characters, where 5 are main characters and the rest are divided in two levels of secondary characters. To create the character setup of these 3D models, large animation studios employ about 10 professionals, with smaller studios employing between 2 to 3 professionals. It takes from 1 to 4 weeks to create a sophisticated character rig for one model starting from scratch.

3.4.1 Film Industry

In *The Incredibles* (2004), Pixar created a muscle-skin system that supported wild cartoony distortion of the face and squash-and-stretch movements (see figure 3.2). They also developed a statistical modeling technique, because the simulations would not run fast enough for animators to see the results in realtime. Pixar created a variety of rig templates to support the different behaviors: characters with super powers had a rig that allowed squash-and-stretch movements, while characters in the real world had a different rig limited to normal movements that obeyed the laws of physics. Some characters ended up having multiple rigs to accomplish all the behaviors they had to perform. Every rig template was based on a multitude of deformers tied to macro-controllers, which allowed animators to manipulate all parts of the face. They also developed a variety of deformers to satisfy specific situations, like the Elastigirl parachute rig. The animation process started by defining the type of rig the character was going to use. Then, they built the musculature on top

[4] http://xface.itc.it

Figure 3.2: Different facial rigs were created to control the characters and produce different effects, such as squash-and-stretch. Images show the main characters: Mr. Incredible, Violet, Dash, Elastigirl and Jack-Jack. (Copyright 2004 Walt Disney Pictures / Pixar Animation Studios)

of the rig with skin that slid over the muscles, following traditional muscle-skin animation techniques [Kahler et al. 2001]. Last, artists manipulated the animation controls to create the desired motion [Russ 2004; Kerlow 2004].

In *King Kong* (2005), facial animation was achieved by combining motion editing, using the MoCap system, and keyframe animation. Weta Digital implemented motion capture analysis and mapping techniques to successfully translate from the most ferocious to the most subtle of Andy Serki's facial expressions to the face of King Kong (see figure 3.3). Standard motion capture techniques were very unstable, due to the complexity of King Kong's facial geometry and the subtlety of the markers displacement error. The MoCap motion capture system is based on Mark Sagar's technology [Sagar 2006]. To achieve the extreme realistic expressions of a giant gorilla that the film re-

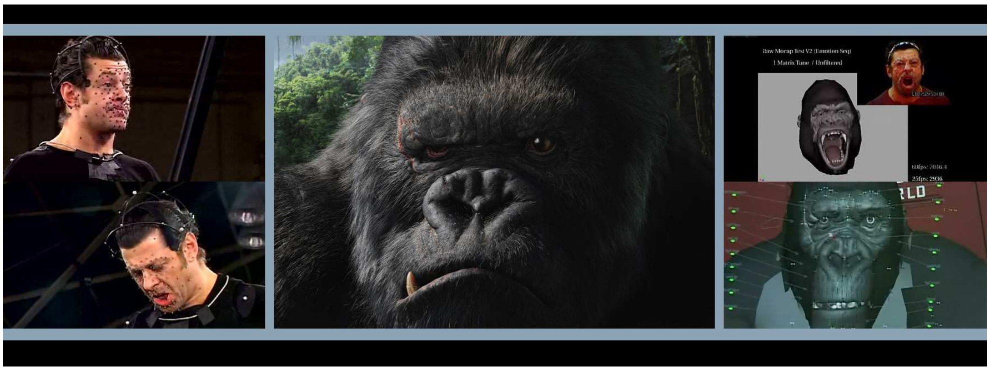


Figure 3.3: Actor Andy Serki performing King Kong facial expressions wearing 132 facial markers (left); after polishing the motion capture data the animations were mapped to King Kong (middle); mapping the actor's facial expression to King Kong (right top); King Kong rig controls for keyframe animation (right bottom). (Copyright 2005 Weta Digital)

quired, the rig of King Kong included: a great number of animation controls that were preset on the creature, which simulated muscle movements and facial expressions; and a sub-system that allowed manually adjusting the animation after mapping the motion capture data. Animating the big ape started by attaching by hand 132 reflective markers to Andy Serki's face. Then, the motion capture data was translated to an analytical model of King Kong's face, to create facial movements. Next, the animation system compared equivalent expressions between the actor and King Kong, decomposed the motion capture data into muscle behavior and mapped it to King Kong's face. This allowed animators to manipulate parts of King Kong's face while retaining the integrity of the live performance. Last, the final touch that gave King Kong that supreme realism was achieved by keyframe animation, because the MoCap system was unable to capture the subtleties of the facial movements, such as the special behavior of the eyes and jaw that were key for conveying King Kong's emotions. [Fordham 2006b]

In *Pirates of the Caribbean: Dead Man's Chest* (2006), ILM used their iMoCap motion capture system and developed a tool called Blockparty to animate the facial models (see figure 3.4). The goal of Blockparty was to speed up the rigging process for all the secondary characters (specially the crew from the Flying Dutchman) and to animate Davy Jones. Blockparty automated 60% of the rigging job by automatically placing the character's skeleton and muscles in the creatures models. However, artists had to rig by hand the additional elements needed to control the subtleties of the facial model. Transferring animations from Bill Nighy (actor) to the 3D characters (Davy Jones and secondary characters) involved several steps: creating the rigs for each model; tracking the iMoCap performance; translating the actor's performance to the 3D models; and finally, extra cleaning and tuning to adjust the motion capture data to the different characters. As the actor and the models where of very different proportions and even shapes, the system used an interpretative process: it had to understand the intent of the action and what

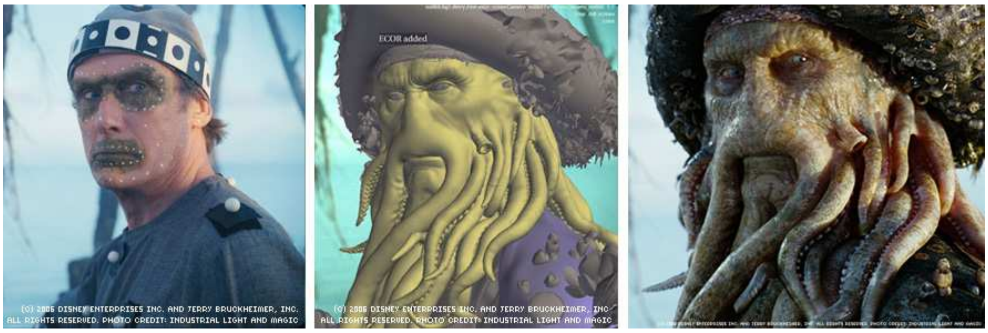

(c) 2006 DISNEY ENTERPRISES INC. AND JERRY BRUCKHEIMER, INC. ALL RIGHTS RESERVED. PHOTO CREDIT: INDUSTRIAL LIGHT AND MAGIC
ECOR added
(c) 2006 DISNEY ENTERPRISES INC. AND JERRY BRUCKHEIMER, INC. ALL RIGHTS RESERVED. PHOTO CREDIT: INDUSTRIAL LIGHT AND MAGIC

Figure 3.4: Actor Bill Nighy enacts his role in the ILM iMoCap suit, with facial makeup, horizontal bands and tracking markers (left); character Davy Jones reproducing the same facial expression (middle); render of Davy Jones including: viewpaint, lighting and compositing layers with cephalopod skin textures (right). (Copyright 2006 Disney Enterprises Inc. and Jerry Bruckheimer Inc. Photo credit: Industrial Light and Magic)

feeling the actor was transmitting, in order to correctly reproduce the facial movements in the 3D model [Fordham 2006a; Jason Smith 2006].

3.4.2 Videogame Industry

In the videogame *Tiger Woods PGA Tour ’07*, Electronic Arts Ucap (Universal Capture) technology creates the most lifelike character model in gaming. It combines head scanning, motion capture and video capture to add high-fidelity animations to the real-time interactive videogame (see figure 3.5). Standard motion capture technology couldn’t be used, because it misses out all the wrinkling and creasing a face does when it moves, and it wouldn’t capture the movement of the eyes to add that extra realism and emotion. The animating process starts with a 3D scan of Tiger Woods head, to keep the in-game 3D model with the exact physical proportions of the real life Tiger Woods. Then, 80 markers are carefully set on the golfer’s face to place him in the Ucap stage. The stage consists of 20 motion capture cameras, 3 video cameras, 2 microphones and a bunch halogen lights. Last, the captured data is mapped to Tiger Woods 3D model facial rig created in Maya. It uses a bone-driven rig: one bone per marker and additional bones for mouth/tongue tracking. Each bone is weighted by an artist using a standard skinning process. This method is very accurate in all areas of the face, except for the interior contour of the lips that has to be adjusted manually by an animator. The addition of animated facial textures to the motion capture data is the key element to reproduce the subtle changes of the face. Tiger Woods facial animation required a very specialized team to be able to process the motion capture data, clean up the video data, process the audio and then put everything together in the rig to integrate it into the game. Using Ucap in videogames with a large number or characters, such as FIFA or NBA Live, wouldn’t be feasible in the short term, because it is still a slow and laborious process. [Borshukov et al. 2006; Miller 2006]

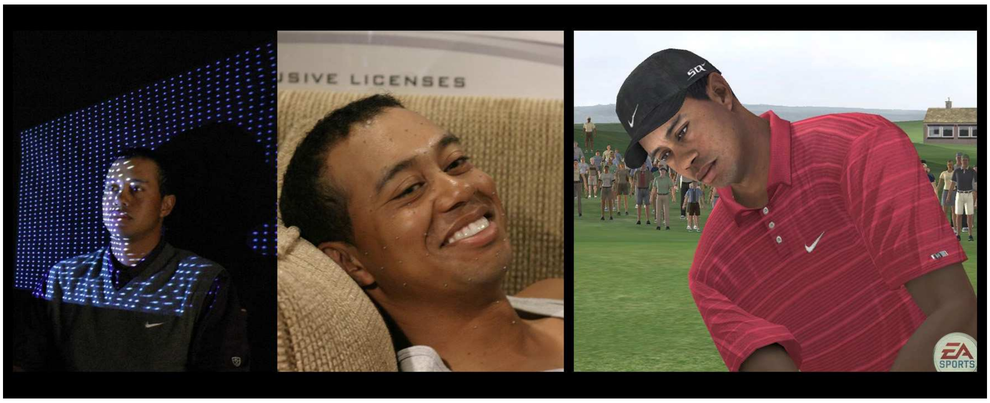


Figure 3.5: Head scan of Tiger Woods using Ucap technology (left); 80 tiny markers carefully set on real-life Tiger Woods (middle); screenshot of Tiger Woods PGA Tour ’07 (right). (Copyright 2007 Electronic Arts)

3.5 Free and Proprietary Applications

We found a number of components already in the market, free or proprietary, developed from academic or industry background, which provide some of the tools needed to create the data workflow required to develop facial animation of a virtual character. Most of the software applications that perform *automated facial analysis and animation* are of extremely low quality and are only used in videogames or web applications. Solutions based on motion capture show outstanding results to be used in films, but are yet far from being automated into an animation pipeline and far from being free of tuning after capturing the performer's actions. None of these applications completely solve the problem of automatic facial rigging of arbitrary characters.

3.5.1 Facial Rigging

Face Robot[5] (2006) by Softimage, is the first and only commercial facial animation software with lifelike quality that assists the rigging of a character. It reduces the preparation time of 3D models and allows animators to concentrate on sculpting the details of every expression. The technology behind the toolset is a soft tissue solver, which creates organic skin movements when the animator manipulates the animation controls. This application has a major drawback: it forces production studios to adapt their pipeline to the workflow defined by Face Robot. It is very rigid and requires setting up by hand some software components. Face Robot is able to produce astonishing results, only when using its predefined technology: artists cannot define their own facial rig, so it is impossible to reuse custom made rigs in different characters. Also, it is impossible to automatically transfer a custom facial rig, severely limiting the usability of the application.

3.5.2 Facial Synthesis

Some applications related to facial animation, but not aimed at automating the facial rigging process include:

ProFACE [6] (Famous 3D): is a standalone application to capture facial expressions and lip-sync that allows animating faces using either voice, video and motion capture. To obtain appealing results, users have to manually define the areas of the face that may be moved independently and blended to create complex expressions.

FaceGen [7] (Singular Inversions): is a standalone parametric face modelling software that allows creating faces from one or more photographs, or at

[5] http://www.softimage.com
[6] http://www.famous3d.com
[7] http://www.facegen.com

random. It allows customizing the face model by race, age and gender based on statistical technology. It includes a set of facial expressions, phonemes and modifiers to control the face model, which can be imported into a 3D animation software to create animations.

LIFESTUDIO:HEAD [8] (Lifemode Interactive): is a standalone application or plug-in for Maya and 3D Studio Max that allows real-time lip-synching based on manipulative phonic structures. It can be applied to human or fantasy head models.

Facial Studio [9] (Di-O-Matic): is a standalone application or 3D Studio Max plug-in that helps create 3D face models. It has over 500 predefined controls that assist the creation process. It allows deforming anything from the eyes, nose, mouth, jaw, chin, ears, cheeks, forehead, eyebrows to the overall head shape, including teeth and tongue, as well as controlling the shading, the textures, the wrinkles deformations and the muscles animation.

Visage Interactive [10] (Visage Technology): is a set of interactive tools that produce character animation. The application prepares the face models for animation using a Facial Motion Cloning (FMC) technology, based on the MPEG-4 standard. It automatically produces 86 standard shapes that are used for further animation. This technology comes from academic research.

Face FX [11] (oc3 Entertainment): is a system for creating realistic facial animation from audio files. Allows artists to model facial expressions and create different character setups. It is used mainly for videogames.

3.6 Brief History of Facial Animation

We look back over the last 35 years of facial animation research, and see how and when it was embraced by the entertainment industry.

1970s: The beginning of facial animation research

- First parameterized facial model [Parke 1972].
- Facial Action Coding System (FACS) [Ekman and Friesen 1978].
- Muscle-based systems use a subset of FACS.

[8] http://www.lifemi.com

[9] http://www.di-o-matic.com

[10] http://www.visagetechnologies.com

[11] http://www.oc3ent.com

1980s: Birth of production studios

- Early 1980s, SIGGRAPH starts to really take off.
- New production studios start to appear, but only a few will survive.
- Bifurcation of development effort can be detected between academic research, who focus on developing new methods, publish and understand current problems; and production studio development, whose main goal is to get the job done, survive and if possible make some money.
- Graphics software comes to the market. Wavefront produces the very first commercially available 3D animation system to run on off-the-shelf hardware, followed by Autodesk Animator.
- New muscle-based models [Badler and Platt 1981; Waters 1987].
- *Tony de Peltrie* (1985) is the first short animated film to use a parameterized facial model.
- *Tin Toy Boy* (1988) by Pixar wins an Oscar for best animated short film.
- Development of the abstract muscle action model (AMA) by [Magnenat-Thalmann et al. 1988].
- *The Abyss* (1989) is released; a film based on an underwater creature that interacts with live characters.
- SIGGRAPH’89 presents a facial animation tutorial, where a simple parameterized model is released to the public domain.

1990s: Facial animation activity explodes

- Creation of facial animation standard: MPEG-4 Facial Model Coding [Haratasch and Ostermann 1997; Ostermann 1998].
- Real-time speech synchronization [Bregler et al. 1997].
- Increased understanding of facial expressions [Noh and Neumann 1998].
- Facial animation is used as agents/avatars in interfaces [Cassell et al. 1994; Pelachaud and Poggi 1998].
- Casper (1995) is the first movie where the main actor is created exclusively using digital facial animation.
- Pixar releases Toy Story (1995), where all characters are synthetic.

- Computer facial animation techniques become a key storytelling component: *Dragonheart* (1996), *Geri's Game* (1997), *A Bug's Life* (1998), *Antz* (1998), *Stuart Little* (1999), *Star Wars Episode I: The Phantom Menace* (1999).

2000s: Commercial success

- Synthetic characters take the leading role in films and videogames:
  2001: *Final Fantasy, Shrek, Jimmy Neutron, Monsters Inc., LOTR: FOTR;*
  2002: *LOTR:TTT, Star Wars Episode II: Attack of the Clones, Ice Age;*
  2003: *LOTR: ROTK, The Hulk, The Matrix Reloaded, The Matrix Revolutions;*
  2004: *Polar Express, Shark Tale, The Incredibles, Shrek 2;*
  2005: *Madagascar, Robots, Chicken Little, King Kong, Star Wars Episode III: Revenge of the Sith;*
  2006: *Monster House, Over the Hedge, Happy Feet, Ice Age 2, Nightmare Before Christmas In 3D;*
  2007: *300, Teenage Mutant Ninja Turtles, Shrek the Third,* and at least 10 other films.

- Exponential growth of the entertainment industry.

- Facial animation is adapted by other sectors, like the medical and forensic industries [Kahler et al. 2003].

- Anatomical representation of the human head that improves existing muscle models [Kahler et al. 2002; Kahler 2003].

- Facial animation systems advance: physically-based simulations [Sifakis et al. 2005], performance-driven [Sagar 2006] and retargeting [Noh and Neumann 2001].

3.7 The Future of Facial Animation

Facial animation research is oriented towards getting better models and tools to control the face, to provide subtle and realistic detail. An interesting direction is to add a new natural language abstraction layer on top of current animation controls, to allow creating expressions through direct input of the Director or based on the environment, rather than requiring an animator to manipulate the controls. [Kondo et al. 2005] presented a framework where the directable animation is created with animator-specified keyframes and the motion trajectory of the deformable object, while maintaining a plausible realism. Their work does not include study of facial animations, so further research is needed to extend the framework to support more general and complex objects.

Besides the sophisticated applications that continue to emerge related to the entertainment and medical industries, there is an increased interest for new approaches in human-computer interaction, like research on embodied conversational agents and affective computing. The goal is to develop systems capable of detecting, processing and appropriately respond to the user's emotions. [Anderson and McOwan 2006] developed a fully automated realtime expression recognition system capable of distinguish between a range of expressions. The interaction between the user and the virtual agent is key for a sucessfull communication. The characters have to show a socio-emotive and socio-cultural response and be fully integrated in the scenario, so their facial model, expressions, visual style and personality play a vital role [Krenn et al. 2004]. For additional research on this area refer to [Ochs et al. 2005; Pelachaud 2005; Gulz and Haake 2006]. Also, using machine learning in data-driven approaches for facial synthesis is increasing and offers challenging research opportunities.

Consequently, for films and videogames, *emotion* seems to be the next big frontier. With the contribution of artificial intelligence (AI) methods, characters will be able to simulate complex behaviors emphasizing essential human qualities like reasoning, problem solving, learning via concept formation and other aptitudes related to what we call intelligence [Loyall and Bates 1997; Strong et al. 2007]. Before integrating AI into current facial synthesis techniques, it is important to understand the different viewpoints of animators and AI researchers when judging the quality of animations, and try to bridge the gap: animators care about the visual result, appropriate timing and clear expression of the emotions, while researchers care about following a logical representation of the data.

3.7.1 Facial Rigging and Unexplored Subjects

There are still several important unexplored issues related to facial rigging, like mapping of the human face physical behavior into arbitrary objects, facial puppeteering and the use of non-linear rigs [Pighin and Lewis 2006].

Transferring the facial rig to arbitrary 3D models remains a challenge: by arbitrary we mean any 3D object that doesn't resemble a human face, like a can, a plane or a lamp, but expresses itself as a talking head. Physically-based methods adds extra realism to animated objects. Mapping human facial expressions performance into any type of expressive digital object without a predefined face, will increase the believability of the character. To achieve a convincing result, the performance of the source model should adapt to the target, but most important, must respect the physics of the target model.

The artists method of choice for rigging and animating a face model is usually linear blend shape. This technique by itself doesn't provide sufficient detail to generate the subtle motion of real faces. Also, retargeting linear rigs normally constrains the results to the morphology of the target model. So non-linear rigs become necessary: they enable to adjust the blending weight and deform the geometry of the source model to the topology of the target, allowing the greatest generality and fidelity of the final results. Thus, it is required research of new methods for retargeting this type of rigs.

Above all, the main research goal is to dramatically decrease the time needed to go from modeling to animation, while maintaining the artist's freedom and face model requirements, and still generate high quality results.

Chapter 4

System Description

Reproducing the subtleties of a face through animation requires developing a sophisticated character rig. But, creating by hand the inner structure and controls of each character is a very labor-intensive and time-consuming task. Our application automatically transfers the rig and animations between characters, at least an order of magnitude faster than traditional manual rigging. It is independent of the appearance or shape of the model, so rig transfer between dissimilar characters is feasible. Artists can create their own rigs and are not forced to use predefined ones. As a result, motionless and inanimate faces come to life with the help of our technology. The system is divided into: rig definition, rig transfer and motion synthesis. We integrated these steps into a prototype system developed in Autodesk Maya 7.0 to demonstrate its usability, flexibility, speed and the quality of the results. The 3D models of this chapter were created by the film company Dygrafilms, who has also supervised the results and provided valuable feedback to improve the workflow of the application. After reading this chapter you should have a complete knowledge of the technology of our portable facial rigging system.

4.1 Problem Statement

The entertainment industry is the main driver for the development of advanced computer facial animation systems. This dissertation deals with the necessity of accelerating the rigging process for film and videogame productions, while keeping the cinematographic quality and real-time response. Today, facial animation is done manually by skilled artists, who carefully place and manipulate the animation controls to create the desired motion. As models become more and more complex, it is increasingly difficult to define a consistent rig that can work well for many different characters. So each facial rig has to be created individually by hand. This traditional method ensures high quality results, but it is slow and costly. Large film and videogame companies can afford hiring lots of artists, but this is not feasible for low budget productions. It takes an experienced digital artist from one to four weeks to create a complete facial rig, depending on its complexity. But if any change must be applied to an already created rig , the rigging process has to restart. Facial rigging becomes a serious bottleneck in any CG production.

Existing facial animation methods, like motion capture, can produce photorealistic results and speed up the animation process, but are unable to adapt the performance to dissimilar characters. The captured animation will look the same in all models, ignoring their different appearances. Our application overcomes this problem because it adapts the inner structure of the characters to the shape and facial features of the models. This allows creating and reproducing animations that respect the style and expressiveness of each character, making them unique. Motion capture focus on analyzing what data to transfer, while our approach focus on what data to transfer and how to represent it. Unlike prior work related to morphing [Blanz and Vetter 1999] and retargeting [Noh and Neumann 2001] that focused only on transferring animations, we transfer the complete facial setup in addition to animations. Previous methods do not deal directly with the artists needs, and most are oriented towards human look. Our approach is general, so artists can define their own rig and then quickly apply it to different models, even with disparate proportions and appearance (human, cartoon or fantasy). This gives artists complete freedom to manipulate the characters: they can create new animations and not be limited by pregenerated ones.

4.2 Our Solution

Our research can be summarized with the following question:

what if... we want to use the rig created
for one character in other characters?

We propose easy integration of facial rigs into traditional animation pipelines, which is a non-trivial task. We have developed a facial rigging system that automatically transfers a rig between dissimilar characters. It is 90-99% faster than traditional manual rigging. The system is capable of preserving the models visual look and its distinctive characteristics during animation. Artists can complete in one hour, tasks that before took them at one to two weeks; something like changing the weights, modifying an animation control position or transferring animations between characters can be achieved “instantly”.

The rest of the chapter describes our proposal and the main problems we had to overcome.

4.3 Challenges in Facial Animation

The face and the facial expressions are fundamental for conveying emotions, so videogames and films require believable facial animation. But generating realistic face movements is hard. Arriving at a unified, elegant, easy to use and flexible rig structure is often complex and time-consuming. Developing a portable character rigging system entails facing many technical and artistic problems. We focus on finding a solution to the technical problems.

4.3.1 Artistic Problems

Creating convincing and appealing facial expressions, like smiling or blinking, seems simple in concept. But, it requires deep understanding of the incredibly complex system that lies beneath a face and very good knowledge of animation principals, to perfectly reproduce facial movements that look realistic. Such realism is constrained by two main factors:

- *inconsistency of facial movements*: humans are very talented in identifying unnatural behavior, due to their everyday familiarity and sensitivity to facial appearance. Non-realistic animations will generate in the audience unpleasant and sometimes repulsive feelings, disturbing the visual perception;

- *diversity of faces*: human faces have different facial features that emphasize their uniqueness, caused by the variation in size and proportions of their bones and muscles. On the other hand, cartoons or fantastic creatures might not follow a human structure definition, which increases the diversity range.

The uniqueness of faces makes facial synthesis so challenging. The smallest anomaly in the face shape, proportion, skin texture or movement is immediately detected and classified as incorrect (see section 3.1 for discussion on the Uncanny Valley hypothesis). To obtain good animations, it is necessary to create a facial rig with the right controls to reproduce every subtle movement of the face. Artists can control the rig by a puppet-like interface of the model that deforms the surface. To create believable facial poses, all controls must react as one and correctly mix each movement in every region of the face.

The quality of animations depends on the character rig created for the face model. But our technology has proved to be independent of the quality and the shape of the rig. If the quality of the source rig is low, the rig and animation transfer to the target model is still successful, but the results are of comparable quality. Then, defining the visual appearance of the face model is limited only by the artistic skills.

4.3.2 Technological Problems

The rig allows artists to edit the model using a set of controls, instead of having to position each vertex of the surface on the desired location. Today, most rigging systems present some of the following technical problems:

- *no standard*: artists do not follow a formal criteria or methodology when creating a rig; they do whatever “feels right”, so all rigs end up being different. Defining a standard would help create a solid platform for artists to build upon;

- *changing the geometry or resolution*: it is very common to change the face model during production, to improve the deformation details or simply because it looks better. Any minor modification in the model surface (a bigger nose, more resolution around the lips) after the character is rigged, causes the rigging process to restart;

- *reusing weight maps*: the weight distribution defined for one character will not work on others. If it is not possible to reassign the appropriate weights in the new model, the resulting animation will present undesirable deformations, specially if the models differ in geometry and proportions;

- *number of shapes*: many productions use rigs based on hundreds of shapes. Usually, too many shapes make it hard to use the rig. Likewise, if a shape is added during production it can generate two problems: the shape conflicts with existing animations, making it necessary to rework some shots; or the new shape does not mix nicely with the the others;

- *preserving a consistent look*: placing by hand the animation controls leads to different artistic interpretations of where to position each element of the rig. This makes it difficult to easily reproduce the same facial pose between different characters. Consequently, it becomes hard to guarantee a consistent look throughout the production;

- *complexity of the UI*: artists place many controls to handle every region of the face and reproduce the subtleties of each expression. If controls are not efficiently organized in a few layers, it can lead to an extremely complex UI with a steep learning curve. It will take animators too much time to master the UI;

- *adding rig controls:* a rig usually goes through multiple iterations before an animator starts working on it. It is very common to change animation controls or add new ones during production. This leads the rig to be readjusted and tested, to verify if it still deforms correctly.

4.4 System Features

Our system solves the common technological problems of facial animation and provides many interesting features to the entertainment industry:

- standardization of the underlying rigs;
- complete freedom to create the source rig, artistically and technologically;
- interactive control over the character's rig components;
- combining motion capture data with keyframe animation;
- changing the target model's topology and instantly reapply the source rig or the previously customized target rig;
- modifying the source rig and quickly propagate the changes to the target models;
- transferring textures between dissimilar characters;
- automatically skinning a character using a predefined rig and weight distribution;
- mapping motion capture data into a source rig and retarget the animations to dissimilar target models.

4.5 System Overview

Creating and placing by hand each component of the rig (bones, muscles, controls) quickly becomes impractical when complexity grows. The system we propose can handle simple and complex rigs based on a new approach. It implements a method that is:

- *generic:* the facial rig can have any type of configuration and does not force the use of a predefined rig;
- *flexible:* the rig has no initial constraints;
- *independent of the shape:* a facial rig can be transferred between models that have different topology, look and appearance;
- *enhances artistic freedom:* artists can use any tool or deformation method to create the rig.

The system deals with the setup of the character rig. It allows using the rig created for one character in others. The rig provides a way of describing facial expressions and animations with a small number of controls that move the vertices of the model via a simpler interface. The rig constrains the expressions of the model, so by transferring it to the new character we generate valid expressions.

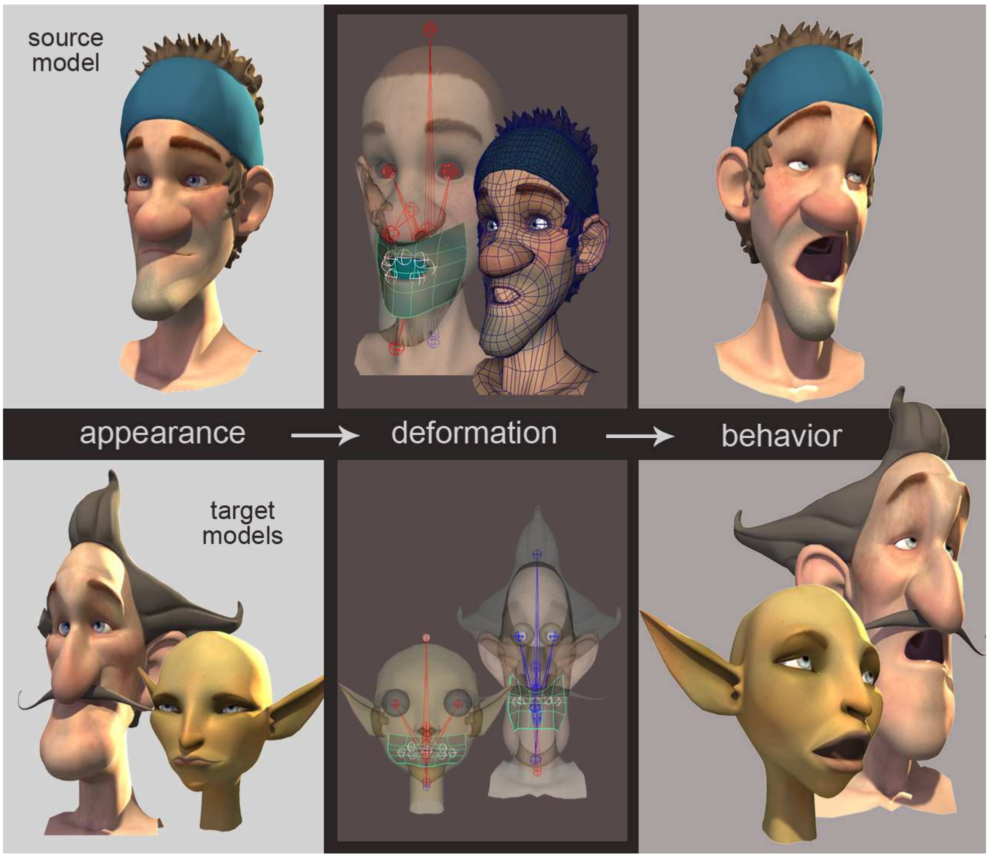


Figure 4.1: Overview: define the source and target models; adapt the source model geometry to fit the target; transfer attributes and shapes; bind the influence objects and skeleton to the target. The result is a model ready to be animated. (Copyright 2005 Dygrafilms)

We begin with two 3D face models. The first one, we call *source model*, is rigged and includes a series of attributes: a control skeleton, a number of influence objects that represent the inner structure of the face and animation controls, facial expressions (shapes) and animation scripts. The rig doesn't have initial constraints and can be created by an artist. The second model, we call *target model*, doesn't have a character rig associated to it (figure 4.1 shows an overview). The source and target models can have different descriptors: one can be defined as a polygonal mesh and the other as a NURBS surface. Also, the faces do not need to have the same number of vertices.

The source rig information is used as the direct input for transferring the setup to the target model. First, our deformation method deforms the source model surface to match the geometry of the target. We landmark the facial features to keep correspondence between source an target model, and then employ a computer vision interpolation technique named Thin Plate Splines (TPS) [Bookstein 1989], as our deformation kernel function. After the TPS, the source surface only has exact deformation at the landmark positions of the target model, while the rest of the points lay outside the target surface. We solve this by applying a *dense correspondence* algorithm, which projects every point of the warped surface to the closest point of the target and determines the correspondence between every source and target vertex.

Using as reference the previously deformed source surface, we call *guide model*, the method accurately places the source rig attributes into the target model, even if they have different geometric proportions. We had to adapt the TPS to properly deal with each attribute specific characteristics. The dense correspondence avoids placing additional landmarks on the influence objects or on the skeleton structure. The deformation process achieves excellent results in positioning the source rig attributes in the correct regions of the target face. For example, joints and NURBS surfaces are relocated in the target model, based on the correspondent position they have in the source model. They are also transformed to fit the shape and size of the target (figure 4.2).

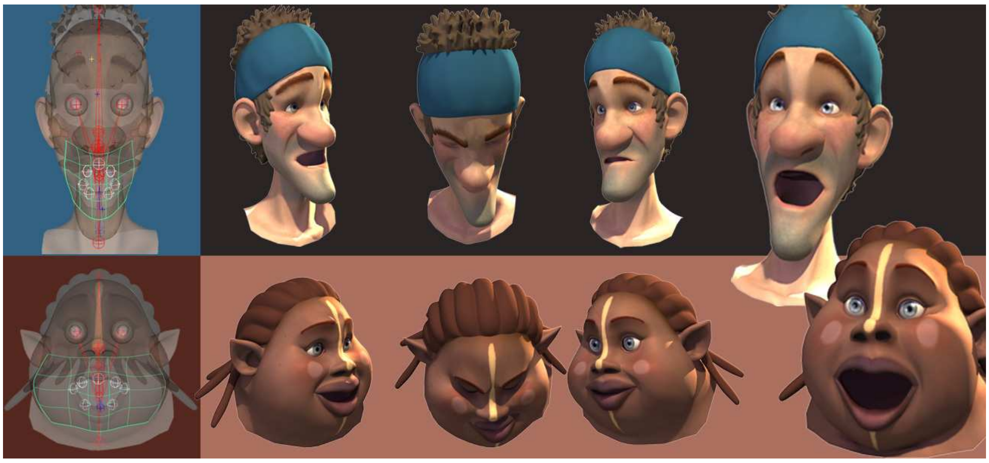


Figure 4.2: Transfer rig and animations between models: source (first row) and target (second row) have different triangulations. Notice the different proportions of the models: the source has a big nose and skinny face, while the target has a small nose and a fat face. (Copyright 2005 Dygrafilms)

After the deformation step comes the skinning, based on a smooth binding algorithm. It binds the transferred attributes to the target model using the adjusted weights of the source, avoiding the need for manual weighting. The weights at the target are calculated using the deformation method. Each vertex of the target model accurately adapts the blending weight of the joints and influence object, based on the source model weight distribution, to properly represent the target facial look and behavior (see figure 4.4). Last, as the target model is already rigged and weighted, transferring facial animations is a straightforward process. The method only needs to scale and adapt the animation curves to fit the proportions of the target. The end result are face models ready to be animated with production quality rigs. Figure 4.3 shows an overview of the system pipeline and illustrates the rig transfer process with two dissimilar characters.

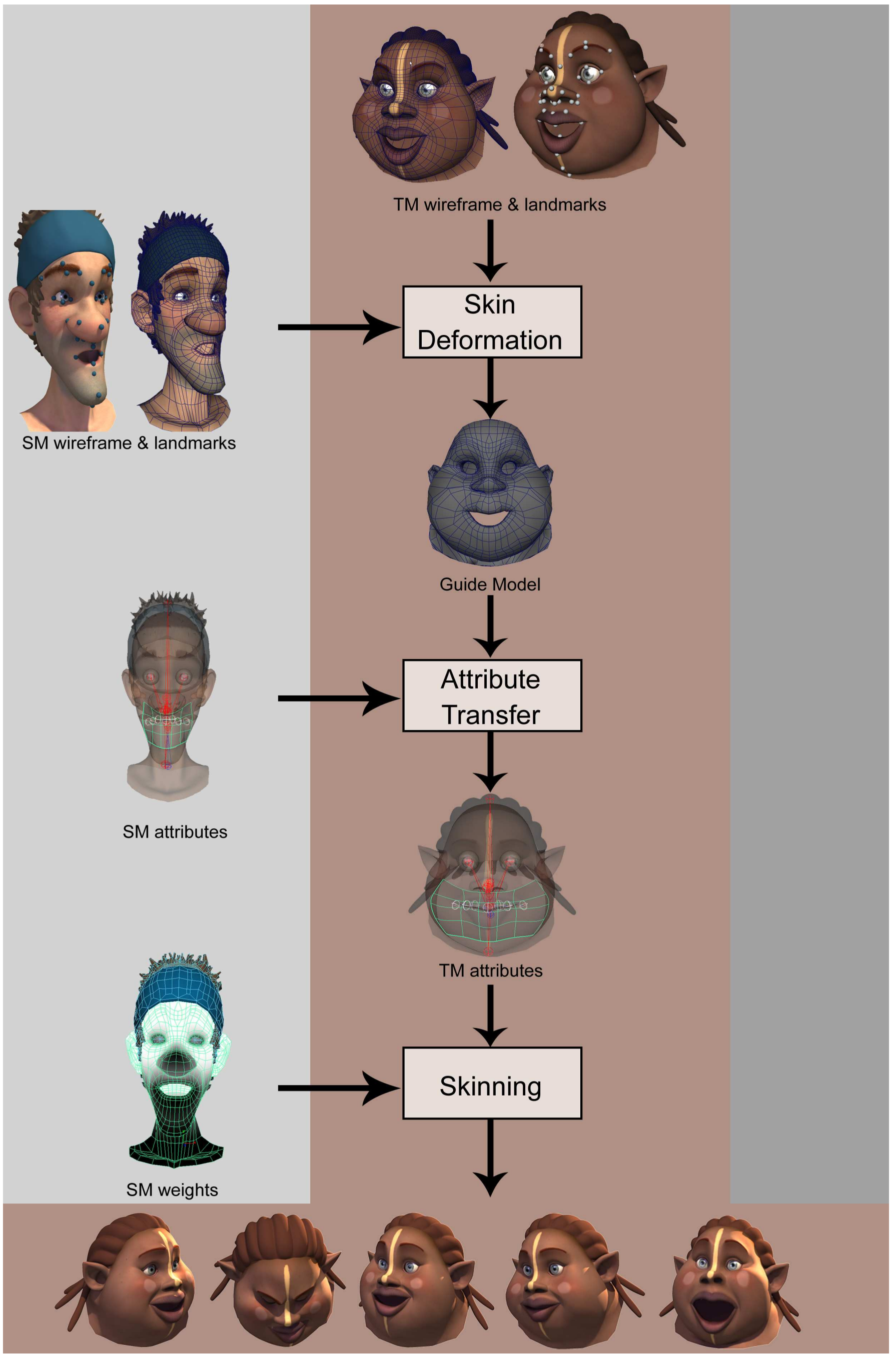


Figure 4.3: System pipeline. Shows the three main steps needed to transfer a facial rig: skin deformation, attribute transfer and skinning. The output of the skin deformation is the guide model, which serves as reference for the rest of the transfer process. The output of the attribute transfer is the target model with the rig components positioned in correspondence to the source. Last, after skinning the character using the source model weights, the target model is rigged and ready to be animated. (Copyright 2005 Dygrafilms)

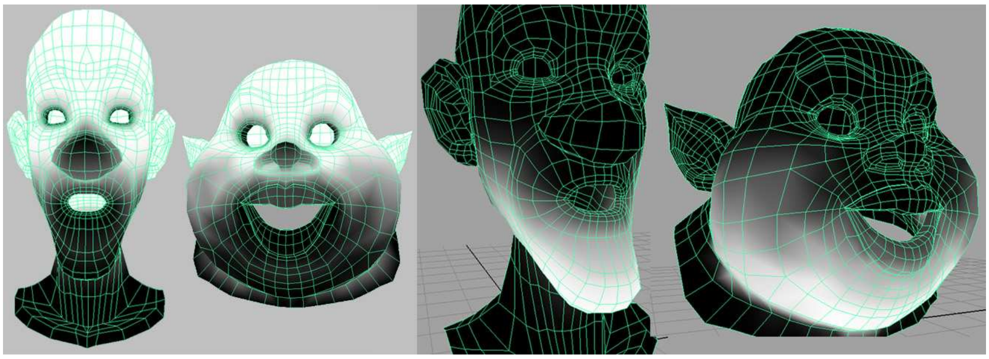


Figure 4.4: Weight distribution on the source and target rigs: whole head (left); jaw bone (right). The models have different wireframe and proportions: the source has a big nose and skinny face, while the target has a small nose and fat face. Also, the source model has less resolution around the mouth compared to the fat lips of the target model. (Copyright 2005 Dygrafilms)

4.6 The Source Rig

Figure 4.5: The source rig determines the components that influence the physical appearance of the target models.

Central to our system is the notion of source rig $\mathcal{S}$, and we use the model in figure 4.6 to illustrate it. The rig is formed by different layers of abstraction that we refer to as attributes: skin surface $\mathcal{S}_S$, influence objects $\mathcal{S}_O$, skeleton bones $\mathcal{S}_B$, facial features landmarks λ, shapes $\mathcal{S}_H$, animation scripts $\mathcal{S}_A$ and other components for representing the eyes, teeth and tongue. We can assign additional attributes to each of these layers: weight, texture, muscle stress, etc. [Haber et al. 2004].

The source rig helps define the appearance of the characters. It establishes the character setup standard shared by all the models. Artists can create their own source rig, because they are free to use any type of controls and components to achieve the desired visual look.

The **source rig** $\mathcal{S}$ has been modeled manually and is a highly deformable structure of a face. During the modeling process, we used facial features and regions to guarantee realistic animation and reduce artifacts.

The **surface** $\mathcal{S}_S$ is the external geometry of the character that determines the skin of the face, using polygonal surfaces composed by a set of vertices $\mathbf{r}$ and a topology that connects them.

The source rig is tagged with **landmarks** λ, distributed as a set of sparse anthropometric points. We use the landmarks to define specific facial features to guarantee correspondence between models. Our rig has 44 landmarks

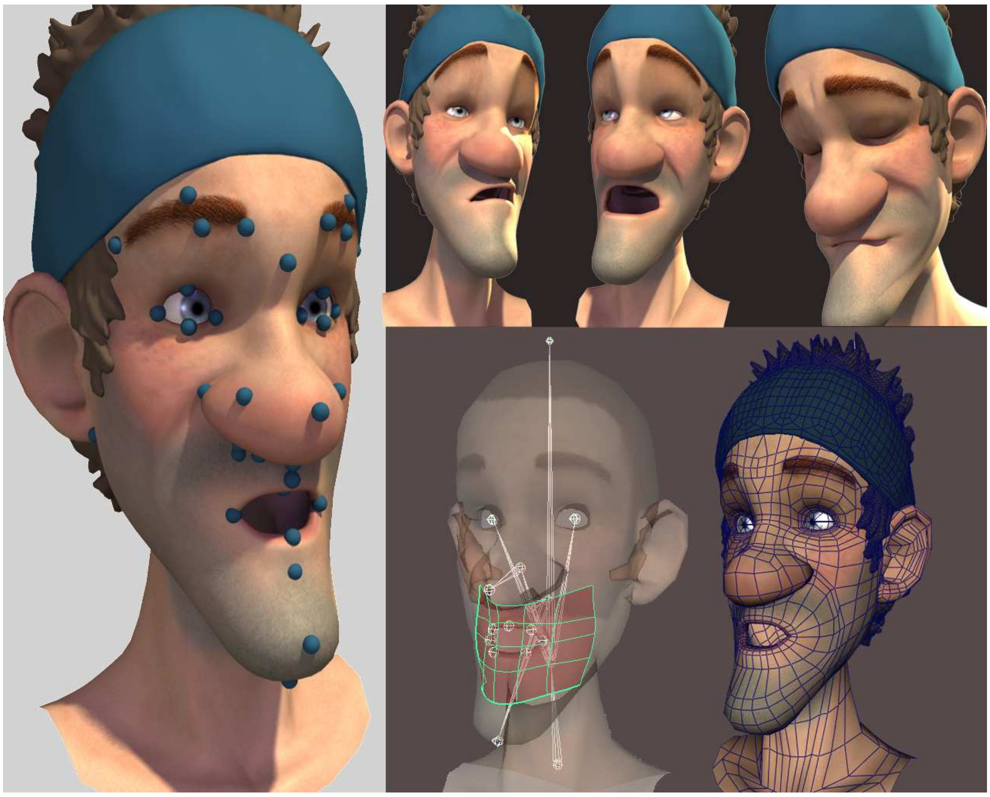


Figure 4.6: Source rig used in our examples and its different attributes: landmarks, expressions (created using shapes), joints and NURBS surface. (Copyright 2005 Dygrafilms)

placed on the surface. These 44 anatomical points are the most prominent and distinctive points on human like face geometries [Farkas and Munro 1987] [DeCarlo et al. 1998]. It is possible to use a different number of landmarks.

The **skeleton** $\mathcal{S}_B$ is a group of bones positioned under the skin. It defines the pose of the head and controls lower level surface deformation. Each bone is defined by two joints, one at each end of the bone.

The **influence objects** $\mathcal{S}_O$ are objects that affect the shape of the skin and help artists control the 3D models. They include: NURBS surfaces, NURBS curves, lattice deformers, cluster deformers, polygon mesh, and others. Figure 4.6 (middle bottom image) shows a NURBS surface used to simulate the behavior of the Orbicularis mouth muscle.

The **shapes** $\mathcal{S}_H$ are new 3D face models created by applying deformations over the geometry $\mathcal{S}_S$ of the character. A shape is a 3D facial pose of the source model, where $\mathcal{S}_H$ and $\mathcal{S}_S$ have the same geometry. Shapes are usually modeled manually by an artist. They represent facial expressions or partial deformation of a specific area of the face. They are used to create blend shapes, which let you change the shape of one object into the shapes of other objects. The interpolation between shapes results in facial animations.

The **animation scripts** $\mathcal{S}_A$ consist of a list of animation curves that determine motion. Each animation curve represents changes in the value of an attribute, like shapes or bones.

4.7 Rig Transfer

appearance
deformation
behavior

Figure 4.7: Deformation process adapts the source rig to the target models.

The system automatically transfers the source rig structure to individual 3D face models. Rig transfer has three main steps: first, we deform the source rig surface to match the geometry of the target face model; second, we adapt the influence objects, skeleton, shapes and attributes of the source rig to the target model; third, we bind the transferred elements to the target model and obtain a rig ready to be animated.

The face model that inherits the source rig setup is referred to as $\mathcal{F}$. It has a face surface $\mathcal{F}_S$, which determines the face geometry and shape, and a set of landmarks μ placed on $\mathcal{F}_S$. $\mathcal{F}_S$ is composed by a set of vertices that define the connectivity between the points. The landmarks are positioned manually by the artist, to guarantee correspondence with the source rig landmarks (see section 4.6). Even though the source rig has 44 landmarks, it is not necessary to use them all to transfer the rig.

4.7.1 Geometric Transformations

To deform the rig $\mathcal{S}$ into $\mathcal{F}$ we use a computer vision interpolation technique named Thin Plate Splines (TPS) [Bookstein 1989], which is a special case of Radial Basis Function Warping [Carr et al. 1997]. The TPS has been mainly used in 2D images for pattern recognition and face correspondence [Lu and Jain 2005] and can be thought as the two-dimensional analog of the cubic spline in one dimension.

In the physical setting, where the TPS name comes from, the deflection of a thin sheet of metal is bent in the direction orthogonal to the plane. To apply this idea to the problem of coordinate transformation, one interprets the lifting of the plate as a displacement of the three spatial coordinates. Thus, in general, the TPS is stated as an interpolation map $\bar{\mathbf{x}} = T_{\lambda}^{\mu}(\mathbf{x})$ from $\mathbf{R}^3$ to $\mathbf{R}^3$. Given a set of n landmarks λ_i in $\mathcal{S}_S$ and the corresponding set μ_i in $\mathcal{F}_S$, the TPS function satisfies $T(\lambda_i) = \mu_i$ and it is defined as a general function

$$\begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} w_{i,x} U(\mathbf{x}, \lambda_i) + a_{1,x} + a_{x,x} x + a_{y,x} y + a_{z,x} z \\ \sum_{i=1}^{n} w_{i,y} U(\mathbf{x}, \lambda_i) + a_{1,y} + a_{x,y} x + a_{y,y} y + a_{z,y} z \\ \sum_{i=1}^{n} w_{i,z} U(\mathbf{x}, \lambda_i) + a_{1,z} + a_{x,z} x + a_{y,z} y + a_{z,z} z \end{bmatrix}. \qquad (4.1)$$

Following [Bookstein 1989; Rohr et al. 2001], in order to minimize the bending energy of the deformation, we use the kernel function $U(\mathbf{x}, \lambda_i) = |\mathbf{x} - \lambda_i|$, which is the 3D fundamental solution of the *biharmonic equation* $\Delta^2 U = 0$. The summation terms in (4.1), related to the non-linear local deformation, are a weighted combination of the centered kernel functions, with scalar weights $w_{i,\cdot}$. The other terms in (4.1) are related to the linear (affine) global deformation.

The TPS method works well when using scattered, unstructured and unsorted data. The non-linear deformation can be decomposed into the superposition of the *principal warping* directions, computed as features of bending at successively higher levels of bending energy. This means that we have a decomposition of the deformation into orthogonal features of progressively smaller geometrical scales that would be used, for instance, to deal with curving edges between landmarks.

To state the interpolation map defined in (4.1) we have to compute the coefficients (weights and affine parameters) for the present landmark sets. This is achieved imposing the correspondence between source and target landmarks points, obtaining a $(n+4) \times (n+4)$ vectorial linear system of equations:

$$\left[\begin{array}{c|c} \mathbf{K} & \mathbf{P} \\ \hline \mathbf{P}^T & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathbf{W} \\ \mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \mathbf{0} \end{bmatrix}. \qquad (4.2)$$

K is $n \times n$ matrix defined by elements as $\mathbf{K}_{i,j} = U(\lambda_i, \lambda_j)$, **0** is a 4×4 zero matrix, and **P** is a $n \times 4$ matrix defined by rows from the source landmark coordinates $\mathbf{P}_i = (1, x_i, y_i, z_i)$ as

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & z_n \end{bmatrix}. \qquad (4.3)$$

The unknown matrices are decomposed in the $n \times 3$ matrix **W** and the 4×3 matrix **A** defined as

$$\mathbf{W} = \begin{bmatrix} w_{1,x} & w_{1,y} & w_{1,z} \\ \vdots & \vdots & \vdots \\ w_{n,x} & w_{n,y} & w_{n,z} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} a_{1,x} & a_{1,y} & a_{1,z} \\ \vdots & \vdots & \vdots \\ a_{z,x} & a_{z,y} & a_{z,z} \end{bmatrix}. \qquad (4.4)$$

Finally, the independent terms on the right hand side of equation (4.2) are a $n \times 3$ matrix **Q**, obtained by rows from the target landmarks coordinates $\mathbf{Q}_i = (\bar{x_i}, \bar{y_i}, \bar{z_i})$, and a 4×3 matrix **0**.

$$\mathbf{Q} = \begin{bmatrix} \bar{x}_1 & \bar{y}_1 & \bar{z}_1 \\ \vdots & \vdots & \vdots \\ \bar{x}_n & \bar{y}_n & \bar{z}_n \end{bmatrix} \tag{4.5}$$

The term $\mathbf{KW} + \mathbf{PA} = \mathbf{Q}$ in (4.2) ensures the exact point matching of the source points in the target. The term $\mathbf{P}^T\mathbf{W} = 0$ represents a boundary condition that regularizes the non-linear warp in order to vanish its energy at infinity.

Denoting by $\mathbf{L}$ the global system matrix in (4.2) and by $\mathbf{L}_n^{-1}$ the $n \times n$ upper submatrix of its inverse, then the principal warping directions correspond to the eigenvectors of the bending energy matrix defined as $\mathbf{B}_n = \mathbf{L}_n^{-1}\mathbf{K}\mathbf{L}_n^{-1}$. These directions, interpreted as deformations, are a canonical description of the modes according to which the points are displaced regardless of global affine transformations. The bigger the eigenvalue associated to a warping direction, the smaller physical scale of the corresponding feature of deformation. Due to the dependence of the warping directions on $\mathbf{K}$, which is a matrix depending of the adjacency of the landmarked geometry, it turns out that the principal warps of the TPS are closely tied to the geometry of the landmarks. Therefore, another important feature of the TPS is that with strongly intercorrelated landmarks, which is usually the case of facial correspondence, it allows using only a few landmarks to obtain good deformation results.

Surface Deformation

Given a set of source and target landmarks, λ and μ respectively, we denote the map correspondence defined in equation (4.1) by

$$\bar{\mathbf{x}} = \mathrm{T}_{\lambda}^{\mu}(\mathbf{x}) \tag{4.6}$$

that for each point $\mathbf{x}$ minimizes the energy of the surface deformation.

Then, we project every point of the warped surface to the closest point of the target surface. As a result, we get the correspondent point in the target surface for every vertex of the source surface. This is called *dense correspondence* [Hutton et al. 2001] between surfaces, defined in our method as function D, which computes the correspondence of points $\mathbf{r}$ between the source rig $\mathcal{S}$ and the face model $\mathcal{F}$:

$$\mathbf{r}|_{\mathcal{F}} = D_{\mathcal{F}_S}\left(T_{\lambda}^{\mu}\left(\mathbf{r}\right)\right). \tag{4.7}$$

This mapping can present undesirable folds in areas with high curvature or if the distance between origin and target points is large. [Lorenz and Krahnstover 2000] and [Hilger et al. 2004] worked on solutions to avoid these folds. This problem did not show up in our tests with different face models (human, cartoon and fantastic creature). Figure 4.8 shows the result of applying the TPS and dense correspondence between a source and target 2D curves.

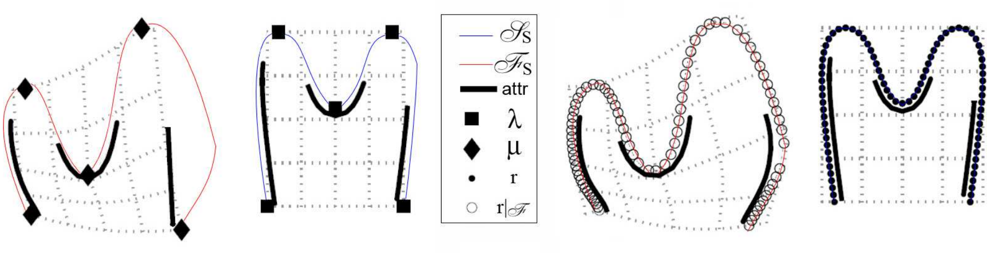


Figure 4.8: Left: a curve uniformly sampled into another curve, using a reduced set of sparse landmarks. Only the landmark positions have exact deformation, while the rest of the source curve points lay outside the target curve. Right: the result of applying the dense correspondence between the source and target curves. We can see that using dense correspondence guarantees better positioning of the structure (represented by the thick black lines) in the target curve compared to just using the TPS (left).

4.7.2 Attribute Transfer

The source rig $\mathcal{S}$ has a set of attributes on the surface vertices defined as scalar or vectorial fields. We have to transfer each of these attributes to surface $\mathcal{F}_S$, which must fit the proportions of the target character. Figure 4.14 shows different attributes (shapes, joints, NURBS surfaces, weights) transferred between dissimilar characters.

For each surface vertex $\mathbf{f}_i$, we find its closest point on $\mathcal{S}_S|_{\mathcal{F}}$, get the interpolated attribute value and assign it to $\mathbf{f}_i$. Based on the dense correspondence between $\mathcal{S}_S$ and $\mathcal{F}_S$, we can adapt to $\mathcal{F}_S$ the source rig attributes, like influence objects $\mathcal{S}_O$, skeleton $\mathcal{S}_B$, weight, shapes $\mathcal{S}_H$ and animation scripts $\mathcal{S}_A$ (the animation scripts have to be transferred after skinning). The dense correspondence ensures an adequate positioning of the attributes and avoids placing additional landmarks on the influence objects or on the skeleton structure, saving time and improving the quality of the results. Of course, each type of attribute requires a specific implementation adapted to its characteristics. For example, when a transferring NURBS surface we apply the TPS to the control points, while when dealing with a mesh we apply the TPS to the vertices. The following sections describe the implementation details for the most relevant rig attributes.

Shape Transfer

To *transfer shapes* we start by calculating the displacement between the corresponding vertices of the source surface at rest and the source shape. Second, we find the dense correspondence between the vertices of the source and the target surfaces. Last, using the TPS, we compute the new position of the vertices of the target surface creating the target shape, which is adapted to the target model proportions.

Figure 4.9 illustrates the shape transfer process. Figure 4.10 shows an example of the shape transfer process with 3D models. Figure 4.11 compares the result of transferring a shape to target models with different mesh resolution.

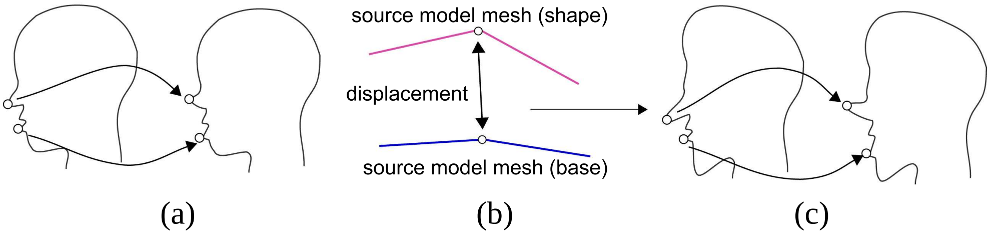


Figure 4.9: 2D representation of the shape transfer between source and target models: (a) source model at a neutral position (left) and deformed at the nose and mouth creating a shape (right); (b) displacement vector between the correspondent source model mesh vertices from the base and shape; (c) displacement vector applied to the base target model mesh (left) to create the shape (right).

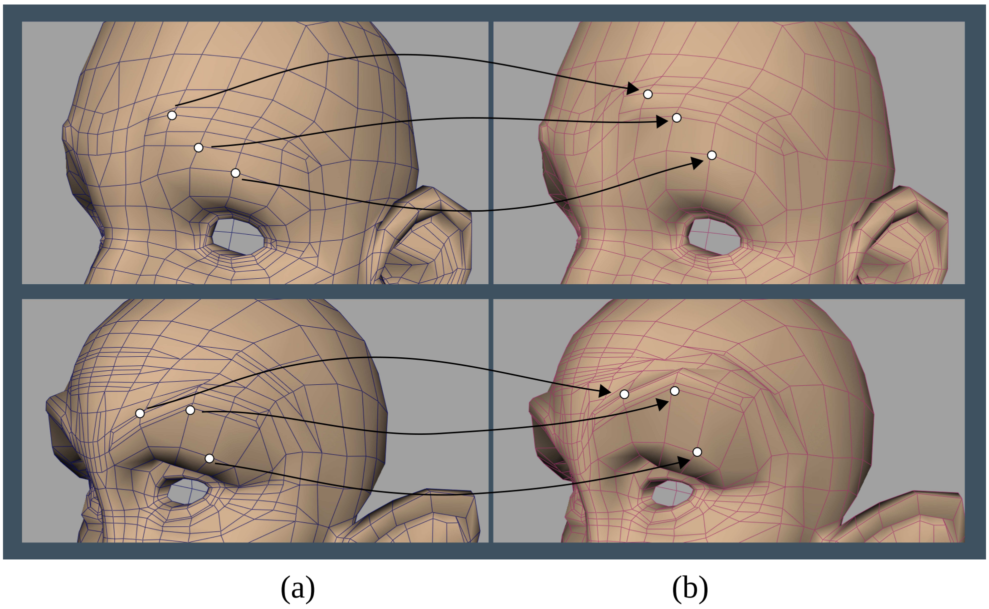


Figure 4.10: Base (a) and shape models (b); source mesh (top) and target mesh (bottom). The dots show a vertex in the base model and the correspondent one in the shape model. The target model shape is calculated based on the source model information. (Copyright 2005 Dygrafilms)

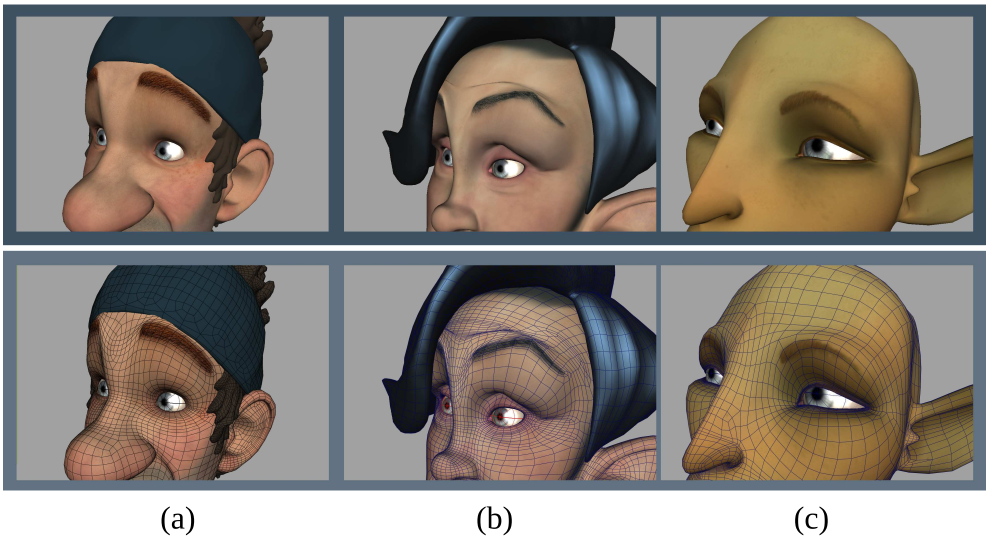


Figure 4.11: Source (a) and target models (b)(c). Notice how the wireframe resolution influences the visual appearance of the model. The shape is correctly adapted to the 3D model mesh. Model (b) is prepared to generate wrinkle effects: has a better deformation region and more resolution around the eyebrows and on the forehead compared to model (c). See the wrinkles in model (b), while (c) is free from wrinkles. (Copyright 2005 Dygrafilms)

Facial Animation Transfer

Changing the position of the rig components (joints, influence objects) by manipulating the controls of the character during a period of time, creates animations. With our system, and usually in any traditional animation pipeline, animation is done after skinning the character (section 4.7.3 describes our skinning approach).

To *transfer animations*, we iterate through all animation curves, filter by type of component and apply the TPS function to find the correspondence between source and target components. Using the TPS allows to adjust the translate, rotate and scale value to the target model, so animations will keep the same proportions as the source but adjusted to the target model's volume.

Figure 4.12 shows the result of transferring the rig components animation curves from the source to the target rig. To obtain these results, it was crucial that all the source rig attributes were scaled and adapted to the new characters.

Adapting the animation curves to the size of the target model is critical for a correct display of the intensity of the movements. Imagine, for instance, that the source model is four times smaller than the target model. Then, if the animations of the source model are directly copyed to the target model, the movements in the target model will be very subtle and will not produce the expected visual result. The same problem happens if the source model is four times bigger than the target model. Then, the animations in the target model will be out of proportion with the size of the character. Our method solves this problem by adapting the animation curve of each component of the rig to

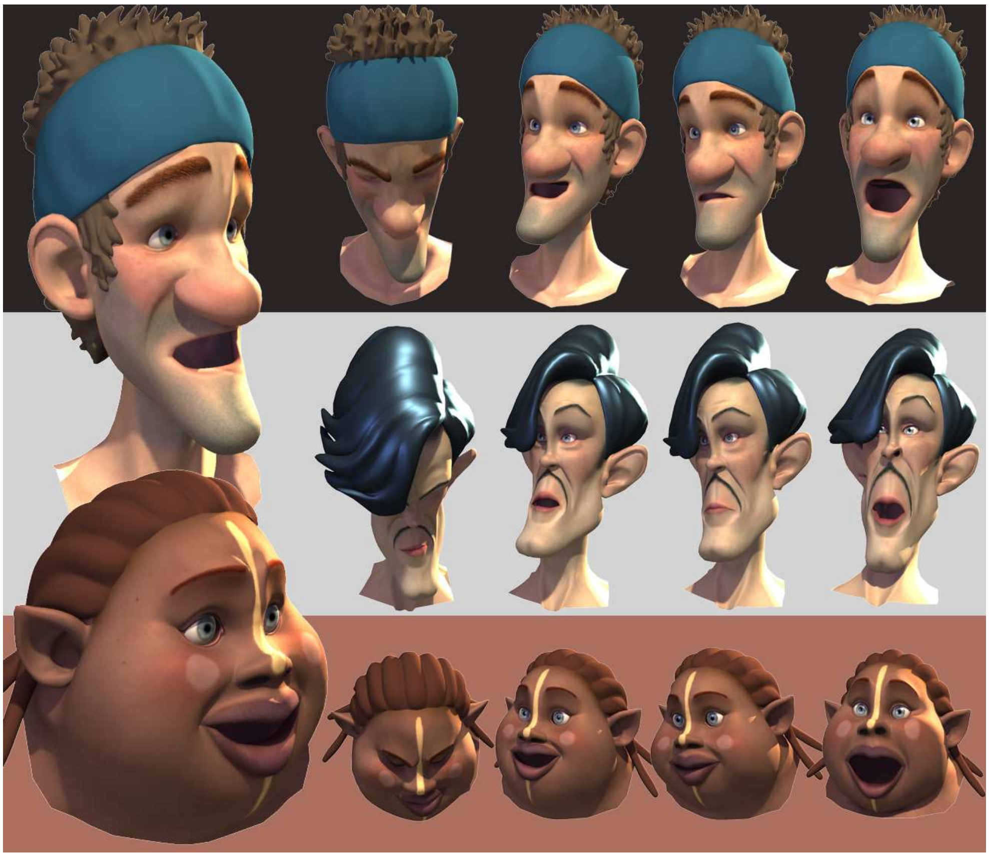


Figure 4.12: Transfer expressions and animations between models: the source (first row) and target (second and third rows) have different triangulations. Notice the different proportions of the models: the source has a big nose and skinny face, while one target has a small nose and the other has a fat face. (Copyright 2005 Dygrafilms)

the target model.

Transfer Other Attributes

Following similar steps to shape and animation transfer, we adapted the method to transfer other rig attributes: skeleton, NURBS surfaces, NURBS curves, polygonal meshes, locators and lattice, among others.

To transfer a generic attribute type, we iterate through all attributes of that type and apply the TPS using the position of each attribute in the source model, which calculates the position on the target model of the attributes. Last, each attribute is transferred to the target model, following the source model hierarchy and precalculated positions.

Implementing the method for each type of attribute takes into consideration the characteristics of the attribute. The input data is different for each version of the transfer method:

Joints: each skeleton joint defines the position of the attribute in the source model. To create the skeleton in the target model, the method uses the joint hierarchy defined in the source model and the new calculated position of each joint;

NURBS Surfaces and Curves: the control points of the NURBS surfaces and NURBS curves define the position of the attributes. To create the NURBS surface or the NURBS curves in the target model, the method uses the previously calculated positions of the control points;

Polygon Meshes: the vertices of the polygonal mesh define the position of the attribute and are used to create the new mesh in the target model;

Lattice: the vertices of the lattice define the position of the attribute and are used to create the new lattice in the target model;

Locators: define a position in space. The method uses this position to create the new locator in the target model.

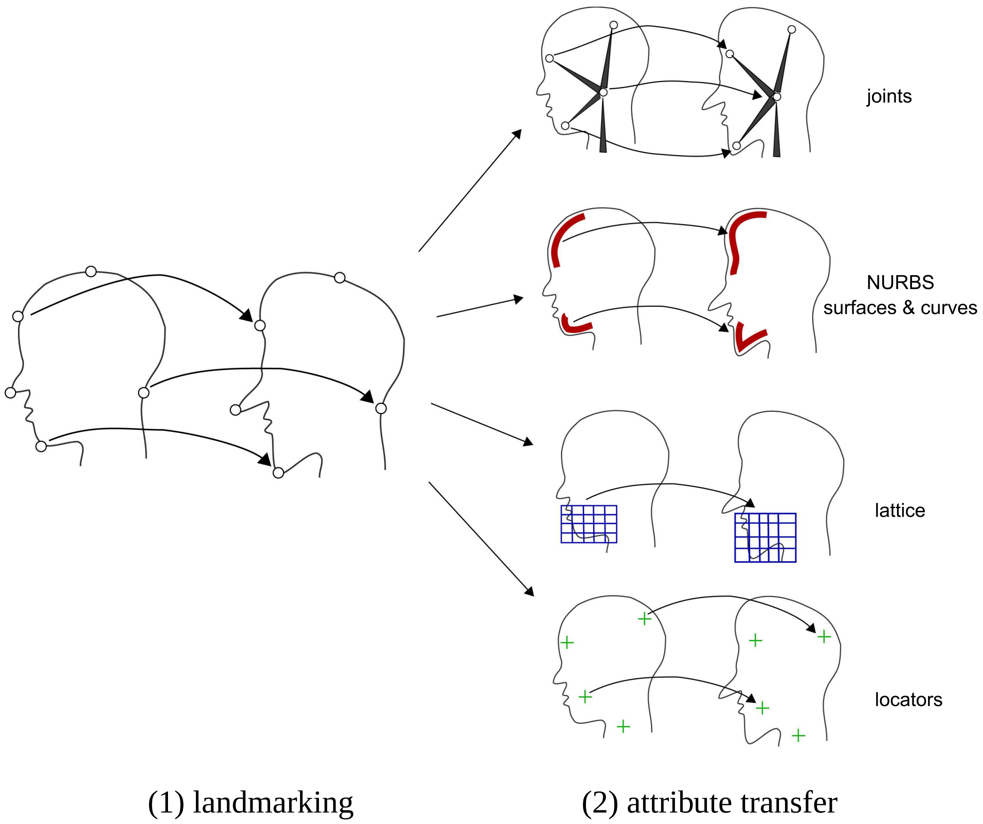
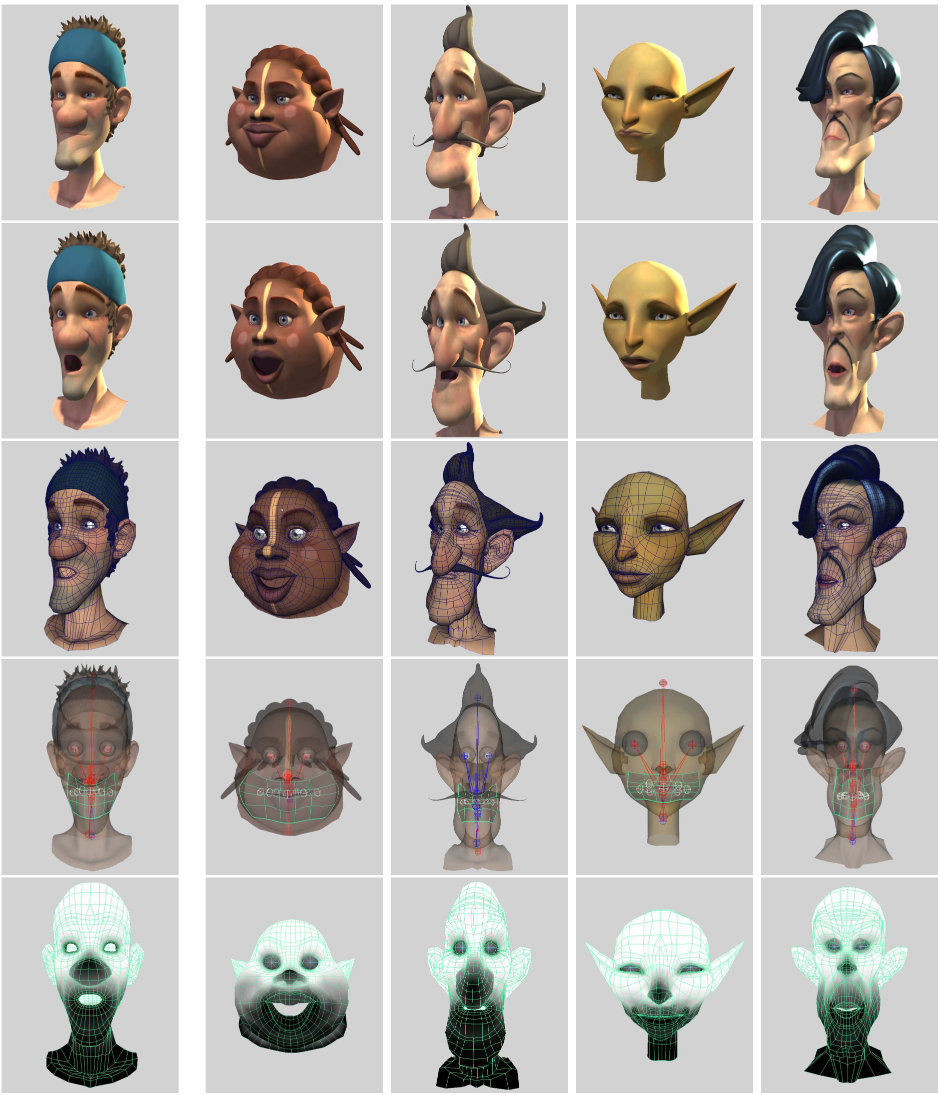


Figure 4.13: 2D representation of the source model, target model and attributes: (1) landmarking guarantees the correspondence between the source and target models; (2) shows how different attributes are transferred from the source to the target model.

Figure 4.13 shows a 2D representation of the transfer process for several different types of attributes. Figure 4.14 shows the result of transferring joints and NURBS surfaces between 3D models.



(SM) Lisandro 3146 triangles

(TM) Mostaza 3098 triangles

(TM) Teseo 3412 triangles

(TM) Hada 2702 triangles

(TM) Demetrio 3126 triangles

Figure 4.14: The first column shows the source model (SM) and the rest show the target models (TM); first row shows the look and appearance of the models; second row is a facial expression example; third row shows the wireframe, which is different for each model; fourth row details the facial rig that includes 21 joints and 1 NURBS surface; and fifth row shows the weight distribution.

4.7.3 Weighting and Skinning

In animation, skinning is the process of binding deformable objects (influence objects and animation controls) to an underlying articulated skeleton. There are several approaches to skinning varying on the degree of complexity and realism [Schleifer 2002]. The skinning process involves assigning each vertex of the skin to one or more skeleton joints and setting the appropriate blending weights. The weight will be the degree of influence of each skin vertex during deformation.

Weighting a 3D character usually takes anywhere from several hours to several days. If after weighting a new animation control is added, the process must restart. Because a non-optimal weight distribution can lead to undesirable deformations, assigning the appropriate weights to a model is a crucial step during the skinning process.

Our skinning process only takes a few seconds. We start by using the source rig $\mathcal{S}$ weight attribute to attach the previously deformed skeleton, influence objects and animation controls to the target model $\mathcal{F}$. The weights at the target are calculated using our TPS based deformation method. Figure 4.15 shows the weight distribution on the source and target models. Next, we automatically bind the skin and influence objects to the corresponding skeleton joints and set the calculated blending weight, using a smooth skinning algorithm [Weber 2000; Kavan and Zára 2005]. We used smooth skinning instead of other algorithms because it is fast and effective, and has been used in many occasions for real time and prerendered animations. The output of the skinning process is the target model rig.

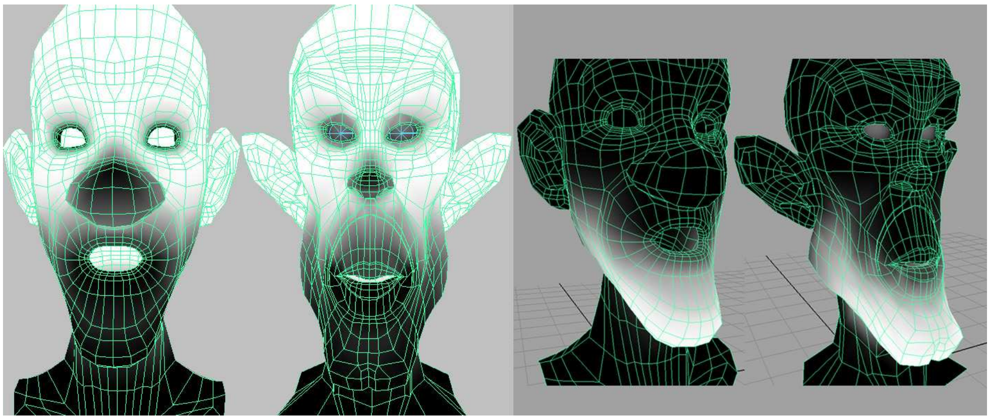


Figure 4.15: Weight distribution on the source and target rigs: whole head (left); jaw bone (right). Notice the different wireframe and proportions of the models: the source has a big nose, while the target has a small nose. (Copyright 2005 Dygrafilms)

4.8 Motion Synthesis

appearance
deformation
behavior

Figure 4.16: The target rigs are ready to be animated and produce convincing behaviors.

After defining the source rig and transfer its attributes to the target models, the characters are ready to be animated. The motion synthesis step creates facial movements to simulate different behaviors. The animations can be generated by manipulating the rig, or by using motion capture data or predefined animations.

Rig manipulation: after the rig transfer, artists are free to change the target model. Artists edit the rig to create new animations or customize predefined ones. For instance, if an animation of a human model is transferred to a cartoon, the model might need more caricatural expressions, so artists have to edit the cartoon rig to achieve the desired effect.

Predefined animations: consists on applying the source model animations to another model with same structure (connectivity and types of joints, influence objects, animation controls). Even when two models share the rig structure, the animation of one may not trivially apply to the other, requiring further adaptation.

Motion capture: target models are animated based on motion data captured from the source model. It is a special case of predefined animations.

The fundamental benefits of the motion synthesis process over conventional techniques are:

1. dramatically reduces time and cost of animation productions;
2. produces fully interactive 3D characters for animation;
3. automatically creates facial animations.

The results of the motion synthesis process depend on the rig definition and the deformation process. It is important to notice that the rig transfer technology is indeed independent of the quality of the rig. So, if the rig quality is low, the transfer is still successful but the results on the target will be of comparable quality. Thus, obtaining convincing behaviors is highly dependent on the rig that is used.

The steps of the motion synthesis process to reuse animations between different characters are (see figure 4.17):

1. select the source model animations to be transferred to the target model;
2. the method checks that the source and target models have identical rigs, allowing direct mapping of attributes between the characters;
3. computes the translation and scale values of the source model to fit the proper size of the target model; these values define the new animation curve in the target model;
4. applies the new animation curve to the target model;
5. (optional) if the results of step 4 do not satisfy the expected visual results, artists can manually edit the rig to modify the animation effects.

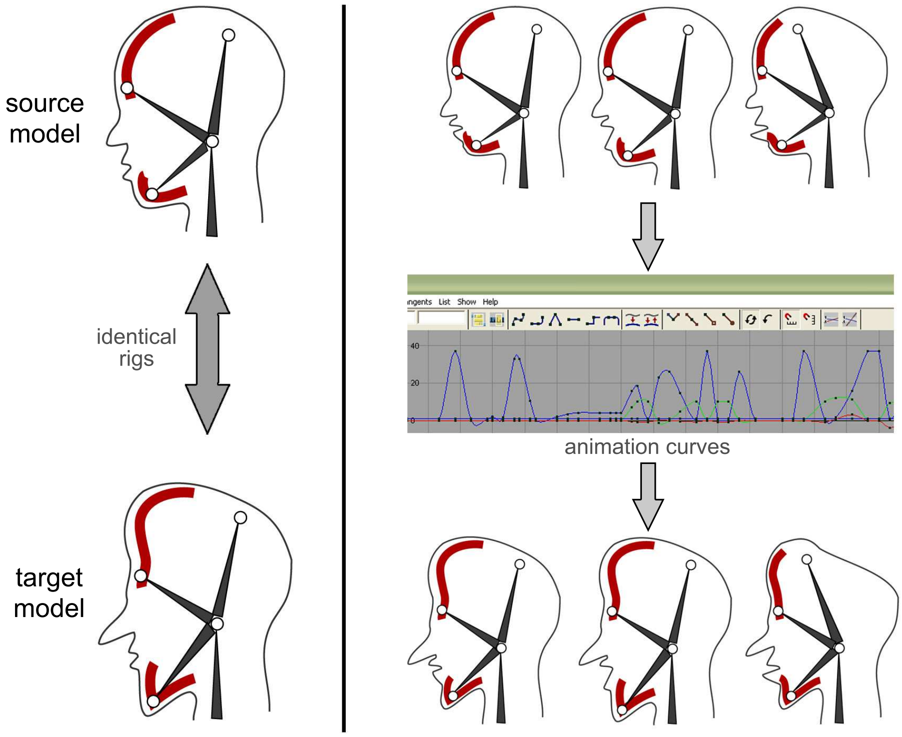


Figure 4.17: Reusing facial animations: black dots in the animation curve graph are keyframes; lines represent the transformation values of the jaw bone.

4.9 Framework

The primary motivation of our research was to find a solution that speeds up the rigging process in CG productions, and lead to the development of an application based on our method that can be integrated into animation pipelines. It is implemented in C++ as a plug-in for Maya [1] 7.0 and was used for testing with film and videogame companies. The plug-in includes a simple user

[1] http://www.autodesk.com

interface to ease the landmarking and assist the transfer process (see figure 4.18). The modular design of the application makes it simple to integrate into existing animation pipelines.

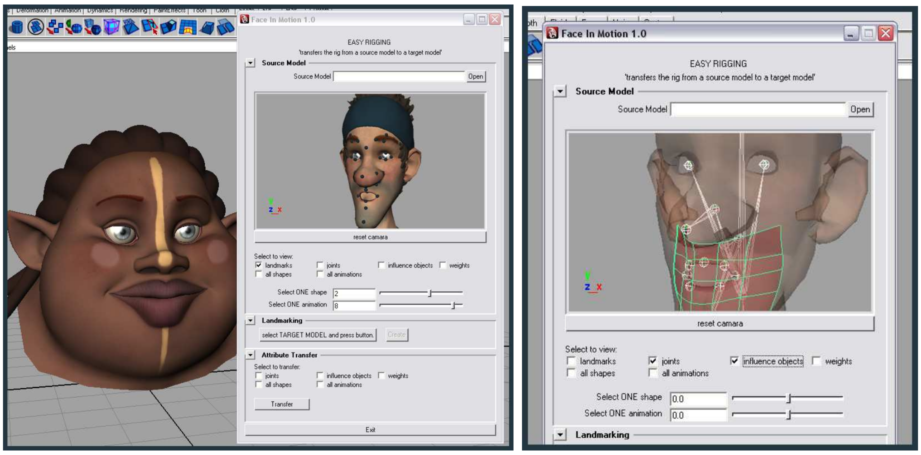


Figure 4.18: Application user interface running on Maya: assisting the landmarking process (left); close up of the source rig viewer with joints and influence object selected (right). (Models copyright 2005 Dygrafilms; Application copyright 2007 Face In Motion)

The application enables artists to:

- fit automatically the rig from the source to the target model;
- manipulate the target as if they were using a puppet;
- adjust animation parameters in the target model or animate the target using predefined source animations.

The application is divided into three main modules:

1. **Configuration:** allows setting initial values to customize the user interface. Eases the portability of the rig between 3D models that use different naming conventions or grouping criteria. It also allows defining the animation control interface or using a predefined one.
2. **Rig transfer:** it is the core module of the application and implements the attribute transfer methods. It allows *loading* the source and target model, *landmarking* the source and target models, *transferring attributes* between source and target models, and *exporting* the data.
3. **Animation and rig controls:** allows importing animations from the source model and apply them to the target model, only if the source and target rig share the same structure. Allows creating new animations using the target model rig. Creates the animation control interface for the target model defined in the configuration module.

It is also possible to run the plug-in in the command line. This gives the user flexibility to integrate it into automated scripts.

4.9.1 Facial Animation Pipeline

Animating virtual characters is challenging, as the face is capable of producing about 5000 expressions. A character like Shrek, from the 2001 movie, had over 500 commands arranged by facial features. For the right brow there was *raised*, *mad*, *sad*, with at least 15 possible commands that activated not only the brow, but also the other parts of the face, which needed to nicely mix and move in conjunction to produce a convincing expression [Falk et al. 2004].

Common facial animation techniques include *keyframe interpolation* [Terra and Metoyer 2004; Igarashi et al. 2006] and *motion capture* [Borshukov et al. 2006; Sifakis et al. 2006; Havaldar 2006]. We focus on keyframe interpolation, where artists use joint displacement and blend shapes to create the desired animation. It takes about two to four weeks and over 100 shapes to create the facial expressions and phonemes for a complex character. Semi-automated systems will take at least four to seven days to create the new target rig. Our application allows creating the rig in one hour, with some tasks being nearly instantaneous: changing the weights, modifying a control position or transferring animations. Contrary to other proposals, we also solve the problem of transferring animations between models with dramatically different proportions, because the displacement information is stored in the TPS.

Our rigging pipeline reflects the workflow of the application (figure 4.19 describes the framework diagram at a high level). The *input* to the pipeline is the source model $\mathcal{S}$ information. The *output* is a fully rigged target model $\mathcal{F}$ ready to be animated.

1. **Landmarking**: $\mathcal{S}_\lambda \leftarrow set(position)$, $\mathcal{F}_\mu \leftarrow set(position)$
   Defines the source and target model landmarks that will keep correspondence between models.

2. **Surface Correspondence:** $\mathcal{S}'_S \leftarrow \mathrm{T}^{\mu}_{\lambda}(\mathcal{S}_S)$
   Ensures the exact point matching at the landmarks and smoothly interpolates the deformation of other points.

3. **Surface Dense Correspondence:** $\mathbf{r}|_{\mathcal{F}} \leftarrow \mathrm{D}_{\mathcal{F}_S}(\mathcal{S}'_S)$
   Avoids placing additional landmarks by ensuring exact deformation of every surface point.

4. **Attribute Transfer:** $\mathbf{f} \leftarrow attributeTransfer(\mathbf{r}|_{\mathcal{F}})$
   Uses the method based on the TPS function.

5. **Skinning:** $\mathcal{F} \leftarrow skinning(\mathcal{F}_S, \mathcal{F}_O, \mathcal{F}_B, \mathcal{S}_w)$
   Binds the deformable objects, influence objects and surface to the skeleton of the target model.

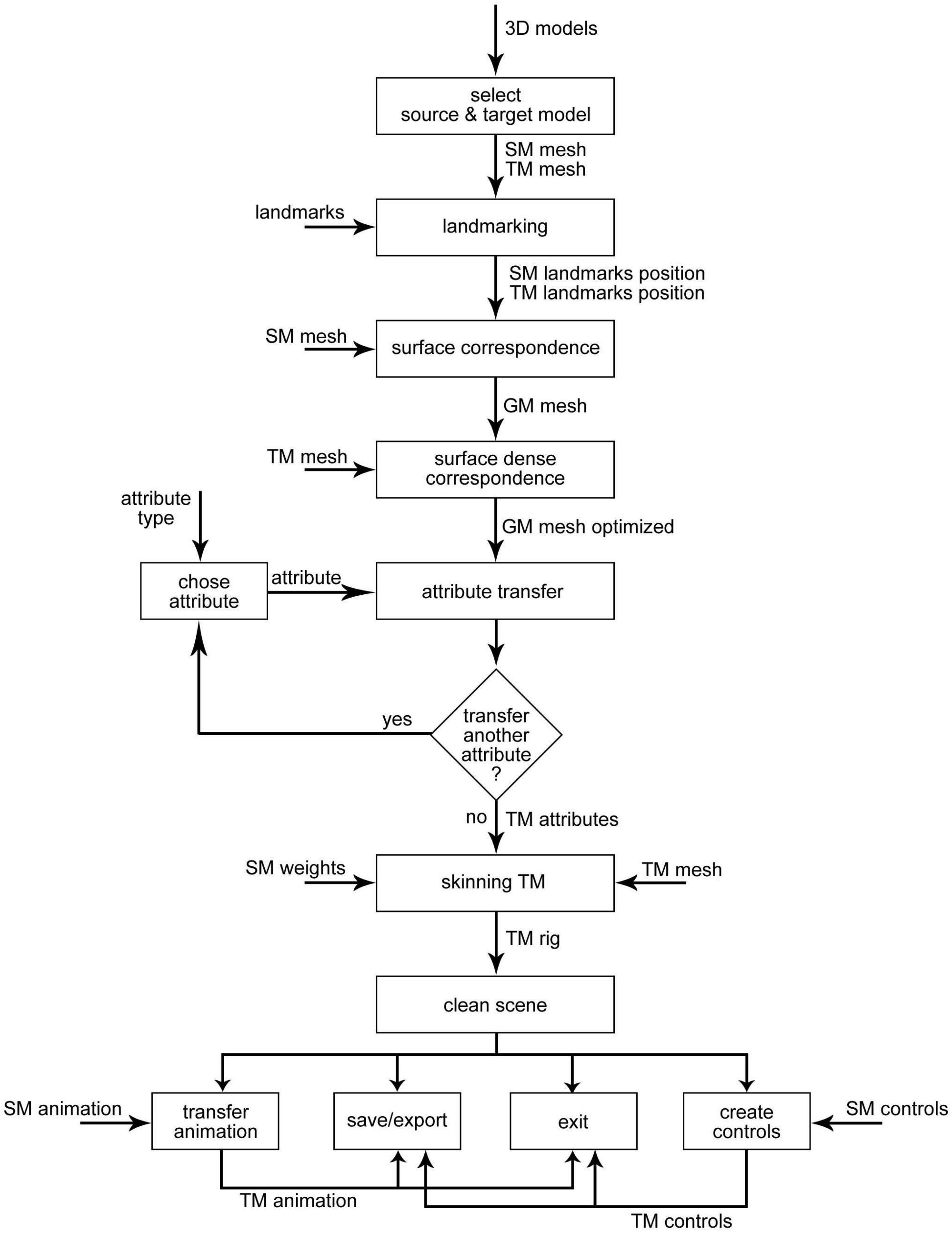


Figure 4.19: Framework diagram: the clean scene module deletes all unnecessary data from the scene, leaving only the target model and its rig. (SM: source model; GM: guide model; TM: target model)

4.9.2 Landmarking Facial Features

Landmaking the facial features of the source and target models is the starting point of the rig transfer process. It is the only manual step in the application. To ease this task, the user can draw curves on the source and target models that define facial regions (like the brow, the edge of the nose and the lips), instead of marking each facial feature. Of course, it is also possible to define landmarks at specific locations. Next, the user chooses the number of landmarks that each curve should have. Finally, the application generates the landmarks (locators in Maya) along the curves describing important facial features. Figure 4.20 describes the landmarking process diagram. Figure 4.21 shows the landmarking of the source and target models.

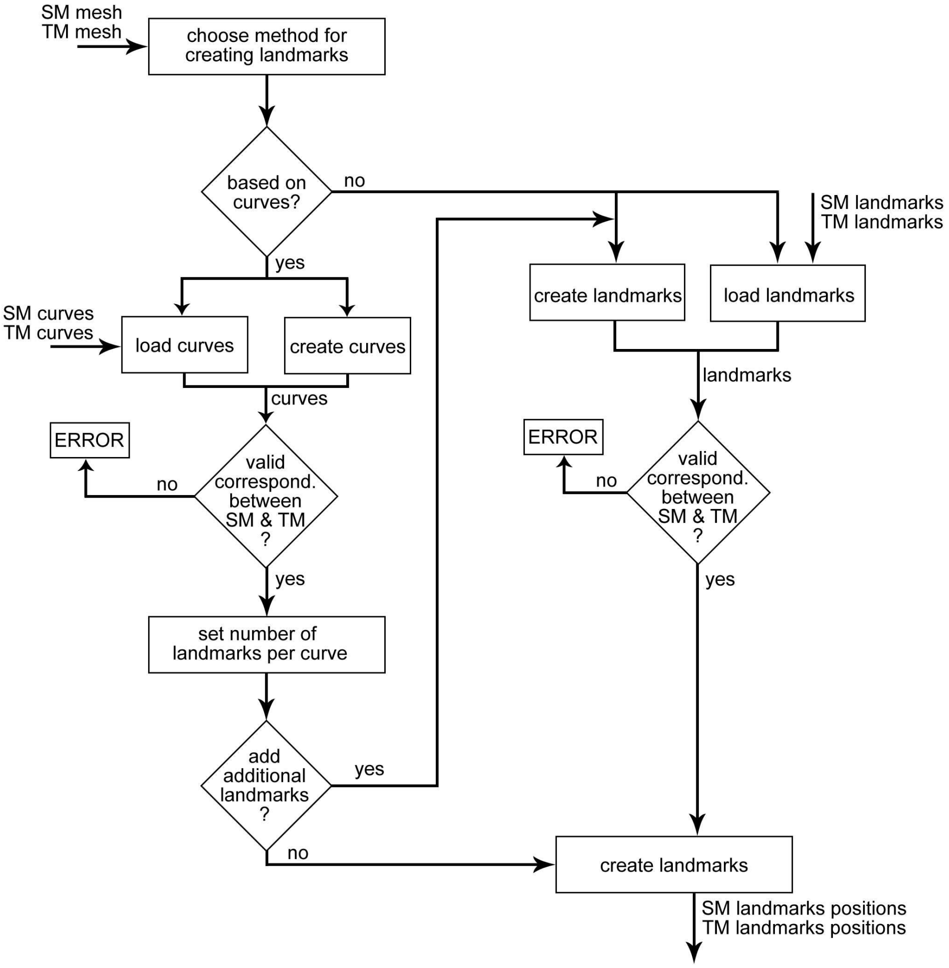


Figure 4.20: Landmarking process diagram.

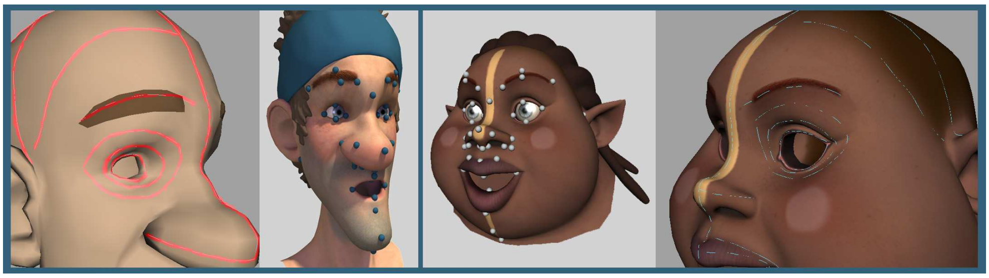


Figure 4.21: Landmarking the source (left) and target (right) models using curves and locators. (Copyright 2005 Dygrafilms)

Chapter 5

Results and Validation

The system was validated with a series of experiments. We used models and rigs from several companies (Electronic Arts, Radical, Dygrafilms), and for each, we transferred the rig and animations to different target models. Then, for the same models, we compared the output of our application with the results manually created by an artist. The results were supervised by Technical and Art Directors, who approved the quality of our rig and animations to be used in CG productions, replacing the artist generated ones. We also made several computations in order to quantify the deformation precision of our method. To verify the versatility of our system, we ordered three 3D models of very dissimilar characteristics and different type of rigs, to test extreme transfer cases. The chapter summarizes the discussions we had with different people from the entertainment industry throughout the research, focusing on the contributions of an automated facial rigging system and the main challenges in facial animation for films and videogames. By the end chapter, you can verify that our solution is capable of performing rig and animations transfer with a wide range of 3D model inputs.

5.1 Introduction

Central to our research is the validation step: we need to know that our system performs in a real world production. For testing the system, we used different types of face models that were created by artists (photorealistic, cartoon and fantastic style).

The tests were divided into: 3D models with cartoon style provided by *Dygrafilms* and 3D models with extreme features provided by *Face In Motion* (i.e., the models we custom ordered). The rigs used for validating our system technology were based on shapes, bones and a combination of both. To confirm the generality and versatility of our approach, the system was also tested with more complex rigs that included a variety of influence objects

(NURBS surfaces, polygon mesh, lattice). All the tests were made on a personal computer with a AMD Athlon 64 3500+ CPU with 2 GB of RAM, running Windows XP SP2 operating system.

Our goal is to reuse the rig created for one character in other characters, independently of how that rig was created. We aim to preserve the properties of the source rig in the target rig and be able to use the animations created on the source model in the target model. That is, if we start with an animation of a human face, we expect to end up with an animation of a cartoon face moving like a human. However, maintaining this fidelity to the source model is not always artistically desirable, because a cartoon model might need exaggerated expressions. We delegate the difficult creative decisions to the artists, who can adjust the target model to their specific needs.

This chapter includes quotes from several people in the entertainment industry to which we have demonstrated our system; they tested it, gave us valuable feedback and helped us fine tune our solution. In an industry with such a high employee turnover rate, many of them have since moved on to other companies or projects:

Juan Nouche: Former *Production Director & Innovation Director*, Dygrafilms.

Xenxo Alvarez: *Technical Director*, Enne Entertainment Studios.
Former *Technical Director*, Dygrafilms.

Dave Fracchia: *Vice President of Technology*, Radical Entertainment.
Former *Vice President Technology*, Mainframe Entertainment Inc.

Jean-Luc Duprat: *Senior Software Engineer*, Neoptica.
Former *Software Engineer, Researcher*, Electronic Arts.

Frederick Fowles: *CG Supervisor in charge of Characters*, Electronic Arts.

Crystal Wang: *3D Artist*, Electronic Arts.

5.2 Initial Considerations

Before describing the results of the validation tests, we start with general considerations and opinions of some entertainment industry professionals on facial animation.

Which are the main challenges for facial animation?

Juan Nouche: *"The main challenge is to achieve higher levels automation in the character creation process, without losing interpretative quality.*

Another important challenge is to have better products, easier to use and that stimulate creativity, contribute to improve the production pipeline and enable reusing work between productions."

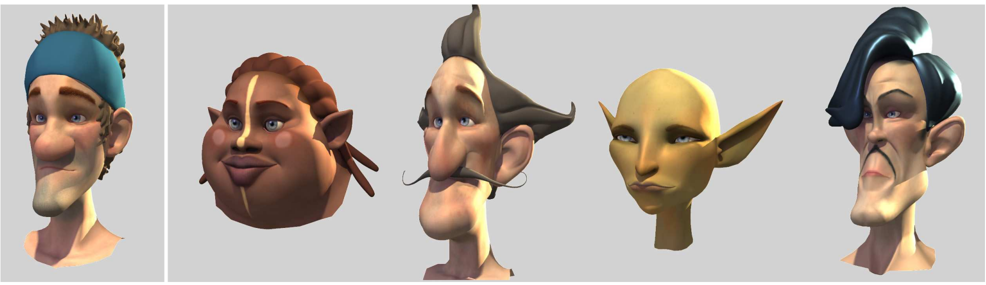
What makes facial animation such a challenging and time consuming task? What problems related to rigging or animating facial models does the film and videogame industry needs to overcome?

Jean-Luc Duprat: *"People are very good at noticing problems in animated realistic characters because of the basic way we evolved biologically. As a result, hand animating characters is very hard (impossible?) to do. Mocap [motion capture] is probably the only way to achieve convincing realism. For non-realistic characters the problem is to convey convincing emotions into a performance."*

Xenxo Alvarez: *"Facial animation is the most difficult part of animation because it needs to convey the character's emotion. Facial rigging and animation processes always use many resources (financial and human), so to make animation more competitive it is necessary to reduce production times without sacrificing quality"*

5.3 Rig Transfer Between Cartoon Models

Cartoon models don't generally follow human proportions. They have different composition and different visual representation of their behaviors. Figure 5.1 shows the source model and the four target models with cartoon style used for testing our system. The attributes transferred to the target model were: 21 joints, 1 NURBS surface, 26 shapes, weights and 2 animations clips (one with extreme facial poses and the other with lip-sync). The shapes were created to manipulate the eyebrow, eyes and cheeks facial regions. The joints control the head, jaw and mouth movements. The NURBS surface simulates the orbicularis muscle inside the mouth to provide subtle deformations. Refer to section 4.7.2 and figure 4.14 for more detailed descriptions. An important part of these tests was to compare the output of our application with the results manually created by an artist, for the same models. The tests were made in conjunction with Dygrafilms, who provided the 3D models, the artists and supervised the results.



Lisandro *Mostaza* *Teseo* *Hada* *Demetrio*

Figure 5.1: Source (left) and target models (right). (Copyright 2005 Dygrafilms)

5.3.1 Visual Validation

To visually validate our application, first, we verified that the rig and animation scripts were correctly transferred from the source to the target models. We checked that the target's joints positions, weight distribution and NURBS surfaces, corresponded to the source model. We also checked that the shapes were nicely reproduced in the target model. This means that the blending of different shapes on the target produces correct facial expressions and that these expressions correspond to the ones in the source model. For the animations, we verified that they maintain the size proportions of the target, even when they are different from the source (see in figure 5.2 the Lisandro source model has a thin face with big nose, while the Mostaza target model that has a fat face with a proportionally small nose).

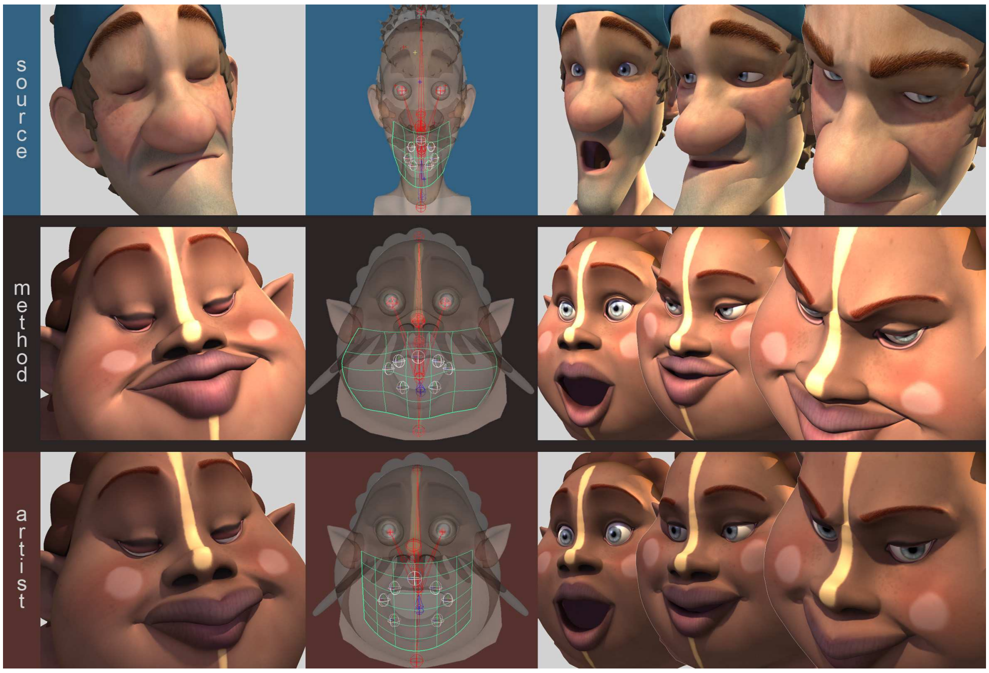


Figure 5.2: Comparison of the target models automatically generated by the method and manually created by an artist. The images on the middle show a NURBS surface that simulates the Orbicularis mouth muscle: we can see that the method is more precise than the artist when adapting it to the target geometry. This is a also an example of transferring animations between dissimilar geometries: the method preserves the intensity of the facial expressions and animations of the source. (Copyright 2005 Dygrafilms)

Second, we compared the target rigs automatically created by the application to the ones manually created by the artists. We focused our attention on two aspects: if the rig was as easy and simple to control and if the generated animations were of the same quality. We also analyzed the positioning of the skeleton joints. We found that the joints of the mouth and the eyes, which are crucial for correct animation, were positioned closely (with some even coinciding) to the same joints created by the artists. Other joints presented bigger deviations but did not affect the final results, because they had less influence on the deformations. Figure 5.2 shows that our method was able to produce a new rig capable of obtaining the results expected by the artists, even though we didn't follow the same procedures or artistic concepts. For instance, artists didn't take into consideration the source model. In figure 5.2 it is possible to compare that the target model generated by the method and the model created by the artist have different lip design. The model created by the method maintains the visual appearance from the source model, while the artist models don't follow the same pattern. The method guarantees homogeneity and consistency in style between all the characters.

The cartoon models allowed us to check the behavior of the method between dissimilar face geometries. The models have very different face proportions and appearance. The challenge is to transfer animations from the source and still keep the target model's intensity and look. Figure 5.2 shows that our method adapts better the NURBS surface that simulates the orbicularis mouth muscle to the target model's geometry, which is crucial to correctly simulate the mouth behavior. This is a particulary difficult task for the artist; he has to make sure that the NURBS parameters are homogeneous to avoid strange deformations or artifacts in the mouth (like pinching in the lips).

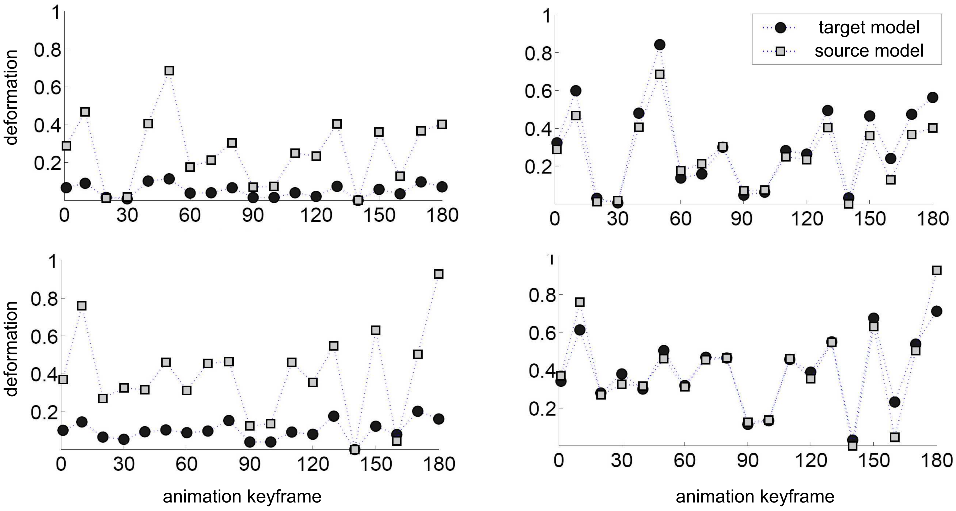


Figure 5.3: Animation curves for the source and target models of the upper middle lip joint (top graphs) and the lower left lip joint (bottom graphs); the left column graphs show the result of exporting the animation from the source model and importing it to the target model using Maya - see how the deformations on the target are weak and damped when compared to the source; the right graphs show the animation after being transferred with our application - the curves on target model closely follow the curves on the source, proving that our method preserves the precision and intensity of the deformations.

Figure 5.3 compares the transfer of a bone animation using our application with exporting and importing the same bone animation using Maya. The curves show that our method preserves the intensity and definition of the animation movements. The deformations are defined in relative terms, to be able to compare them between the characters. The models are normalized according to their bounding boxes. Computing the distance between two joints at the initial moment l_0, and the same distance in the next frame l_1, quantifies the deformation value. Thus, the deformation is an adimensional quantity defined by $\varepsilon = \frac{l_1 - l_0}{l_0}$.

Finally, figure 5.4 shows that our application convincingly captures the complex effect of simulating a talking head, to be used in a film.

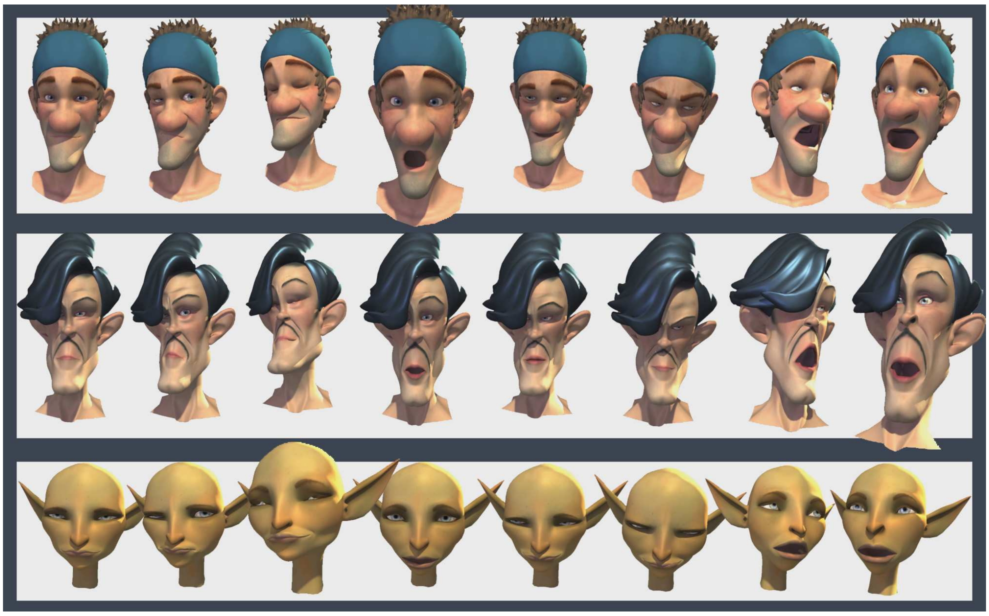


Figure 5.4: Source and target models animation sequence: frames from a video that integrates facial animation with lyp-sync. (Copyright 2005 Dygrafilms)

5.3.2 Timing Validation

The rig and animations generated by the application fulfilled the Art Directors quality expectations. Now, we had to make sure that our method is quicker than traditional manual setups. We compared the time it takes to create the animation controls (joints and NURBS surfaces), weight the model, create the shapes and the animation scripts. Table 5.1 shows the run time of every step of the transfer with the four target models of figure 5.1.

Using our application, we took between 30 and 60 minutes to transfer the rig from the source to the target model and verify the results. Landmarking both models took about 5 minutes. Then, it took us 20 to 40 minutes to visually verify the quality of the transferred shapes, while the animation scripts were nearly instantaneous. Overall, the complete character setup can take one hour. Verifying the quality of the shapes and rig is not part of the method, but we decided to add this information to be able to do a precise comparison between our automated process and the artist's manual job, to whom verifying the results is intrinsic.

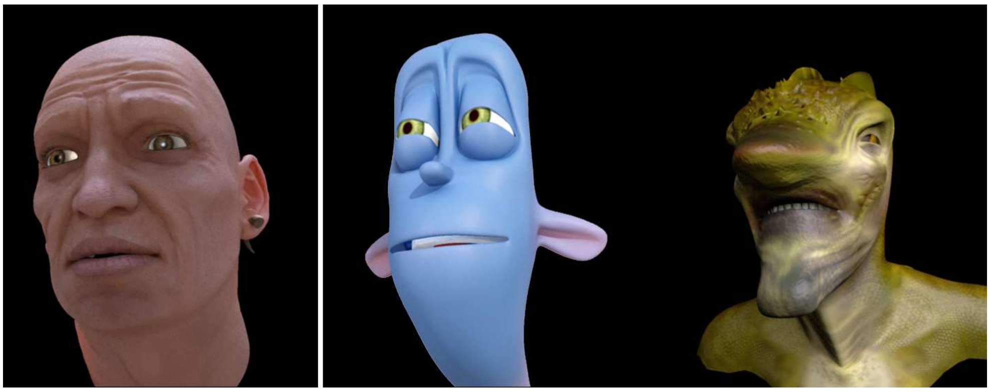
	Teseo	Demetrio	Mostaza	Hada
Triangles	3412	3126	3098	2702
Skin Deformation (TPS)	0.02	0.01	0.01	0.01
Dense Deformation	1.41	1.33	1.35	1.15
Joints Transfer	16.09	16.56	16.55	15.32
Shape Transfer (each)	5.50	4.57	4.41	3.42
NURBS Transfer	21.14	21.35	21.36	21.17
Weight Transfer	8.17	7.36	7.45	7.16
Animation Transfer	0.30	0.21	0.25	0.20

Table 5.1: Application run times of each method step (in seconds).

The *artists* took about one to two days to position the joints, create the NURBS surface and weight the model, two to three days to create the 26 shapes and another two days to create the animations. The character setup for these rigs took an artist about 5 working days. These timings are all from one production company (Dygrafilms). They told us that for more complex characters to be used in production, it usually takes 2 artists working on a rig for at least one week and if the models suffer any modification on the topology the rigging process has to be redone.

So, the tests show that with our method the speed up can be of an order of magnitude or more, and still meet the high quality animation needs of the entertainment industry. We have achieved accurate and fast rigging.

5.4 Rig Transfer Between Extreme Models



photorealistic *cartoon* *fantastic*

Figure 5.5: Source (left) and target models (right). (Copyright 2007 Face In Motion)

To test the method to the extreme of its possibilities, we ordered three very different 3D models: a photorealistic human, a cartoon and a fantastic creature. They were custom built by a team of experienced artists. The models differ enormously in artistic style, deformation behavior, shape and proportions. The human character was used as the source model. It follows an anthropomorphic design that has some emphasized facial features, to avoid falling into a neutral character definition. The cartoon character was used as a target model. Its design concept was thought to test the transfer of facial expressions from a model with subtle expressions to a model that requires exaggerated expressions, and to test the rig transfer method to a non-anthropomorphic model. The fantastic creature character was also used as a target model. Its facial features are very different from the source model, which allows testing the rig transfer between non-anthropomorphic characters. Figures 5.5 and 5.6 show the 3D models used for the tests.

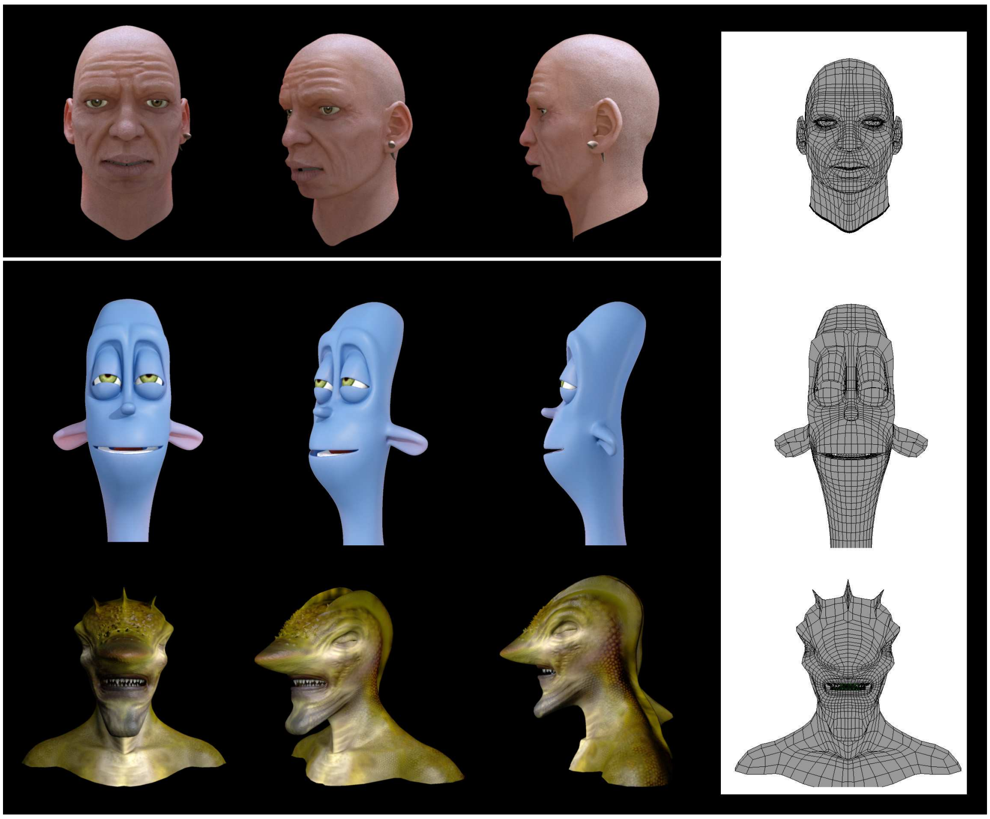


Figure 5.6: Photorealistic source model with 3652 triangles (top row), cartoon target model with 2770 triangles (middle row) and fantastic target model with 3116 triangles (bottom row). (Copyright 2007 Face In Motion)

5.4.1 Preliminary Character Study

Before creating the 3D models, it was necessary to study the morphology and design of the characters. This previous analysis helped define 3D models that had different animation requirements and performance styles. Figures 5.7, 5.8 and 5.9 show the source and target models design. The human model needed very subtle expressions, while the cartoon model required very extreme expressions to convey the same expressions as the human model. The fantastic

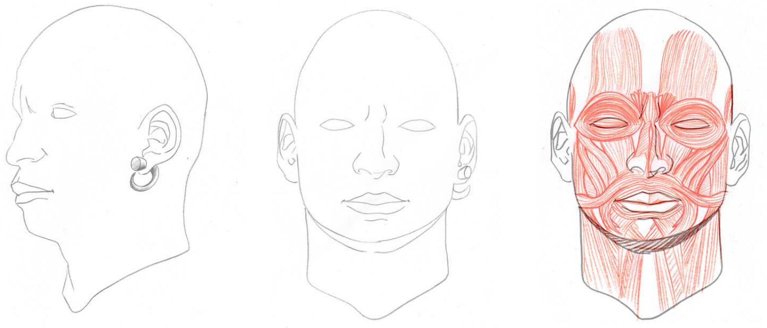


Figure 5.7: Photorealistic: source model illustrations. (Copyright 2007 Face In Motion)

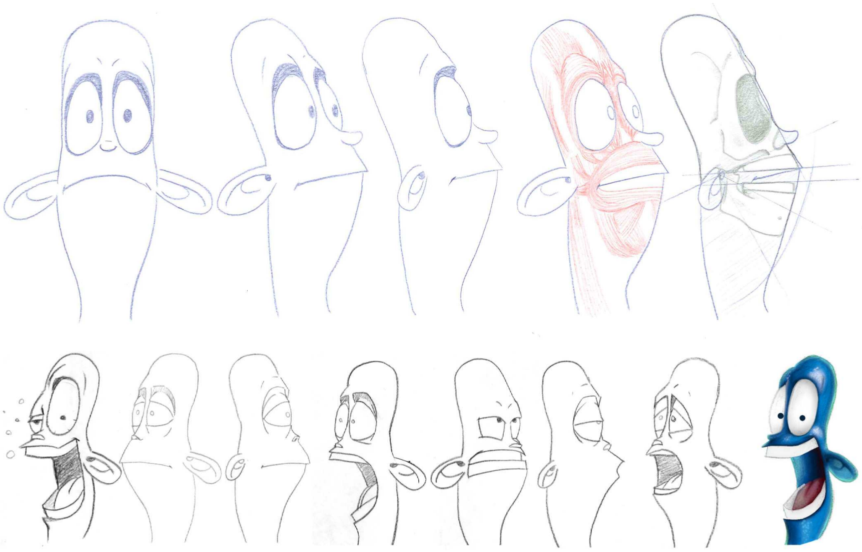


Figure 5.8: Cartoon: target model design and expressions illustrations. (Copyright 2007 Face In Motion)

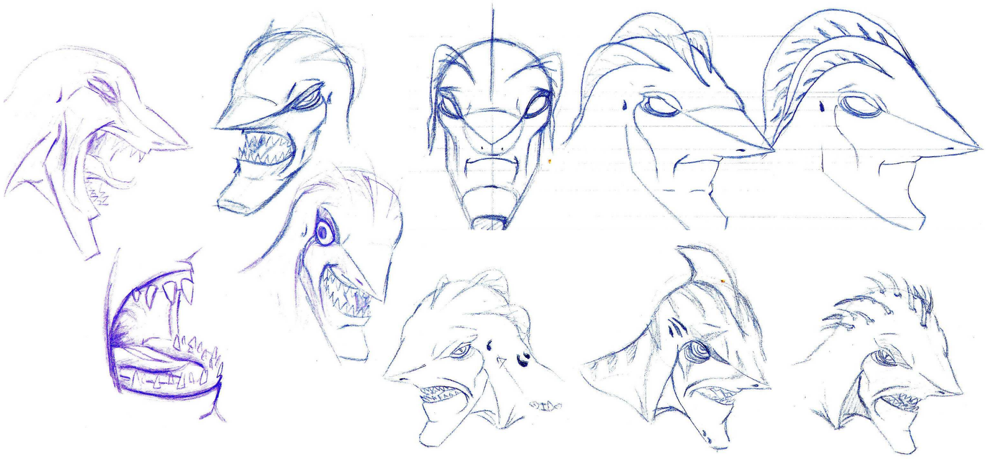


Figure 5.9: Fantastic creature: target model design and expressions illustrations. (Copyright 2007 Face In Motion)

creature didn't need expressions as exaggerated as the cartoon or human. Its physical constitution differs very much from the human model.

5.4.2 Visual Validation

For the visual validation, we were concerned with verifying that the rig was correctly transferred and that the animations performed according to the character's style. Like section 5.3, we checked that the different shapes nicely mixed in the target and that the animations preserved the characters size and look.

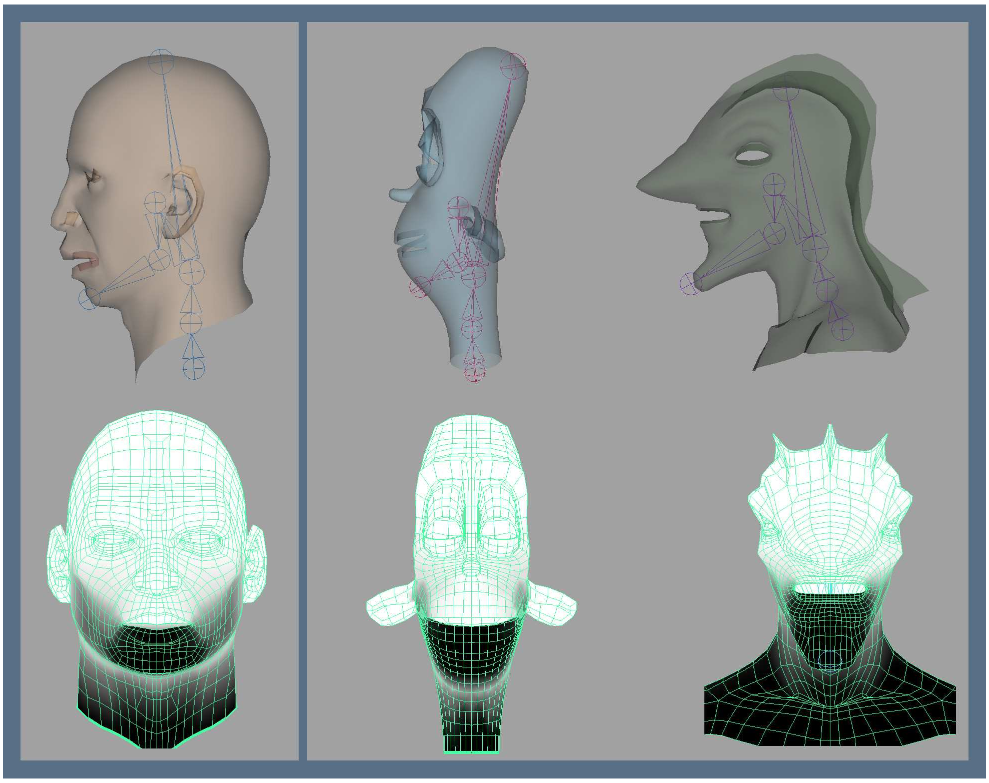


Figure 5.10: Source (left) and target models (right). Skeleton structure (top) and weight distribution (bottom). (Copyright 2007 Face In Motion)

The requirement for the source model rig was to include many different types of deformers that could interact between each other, and still maintain a user friendly interface to manipulate the rig. The rig was divided into three layers: blend shapes, deformers and joints. Each layer provided specific control over the model to create unique expressions. The blend shapes were used to create the main poses. The deformers (NURBS curves, lattice) provided subtle control over certain regions of the face that needed more precise deformation, like the mouth or the eyes. The joints defined the lowest layer of control, which allowed controlling the movements of wider regions of the character, like the head, teeth and tongue.

The source model rig includes: 2 NURBS curves around the mouth, 1 lat-

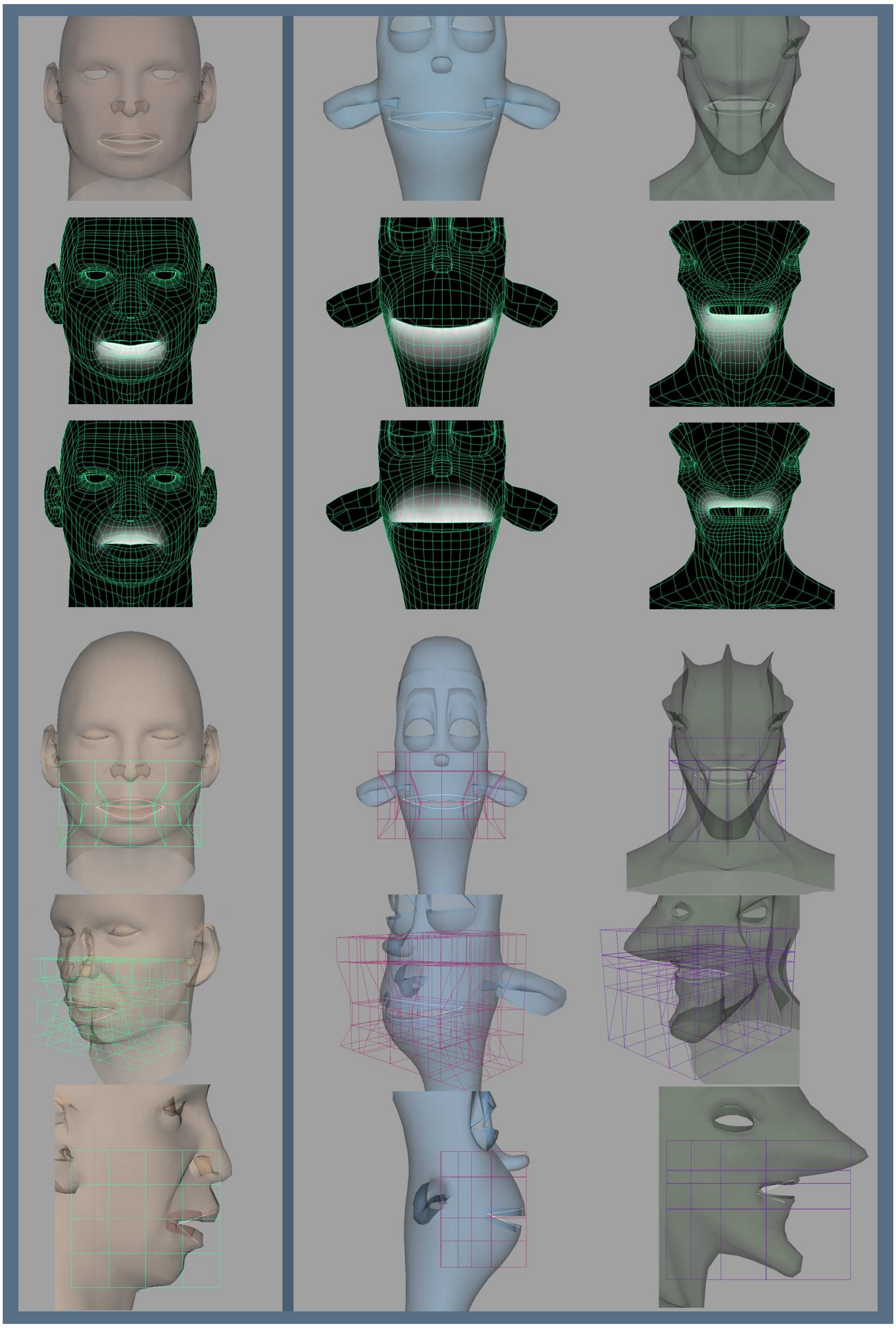


Figure 5.11: Source (left) and target models (right). On the top half, NURBS curves to control the upper and lower lips (top) and NURBS curves weight distribution (bottom). On the bottom half, lattice that controls the jaw: front (top), 3/4 (middle) and side views (bottom). (Copyright 2007 Face In Motion)

tice for the jaw, 6 joints for the head, 5 joints for the tongue, 3 joints for the teeth, 47 shapes and 2 animations clips (one with extreme facial poses and the other with lip-sync). The source and target models are symmetric so it was possible to reuse the shapes created for one side of the head, like eyebrow up. There were a total of 81 shapes. If the models were asymmetric, then it would have been necessary to create the shapes for each side of the head separately, which increases production time. Figure 5.10 shows the positioning of the head joints and the weight of jaw joint. Figure 5.11 shows the weight distribution of the NURBS curves of the mouth and the position of the jaw lattice.

The source rig included more shapes than the one from section 5.3, specially to control the mouth. The method successfully transferred the shapes in the mouth region, which is very complex due to the variety of poses it can perform. But the drawback of transferring shapes between characters with different styles is that the target models inherit the movements of the source. We obtained a cartoon model simulating a human character. This faithfulness wasn't what we wanted, so we needed to come up with an alternative solution.

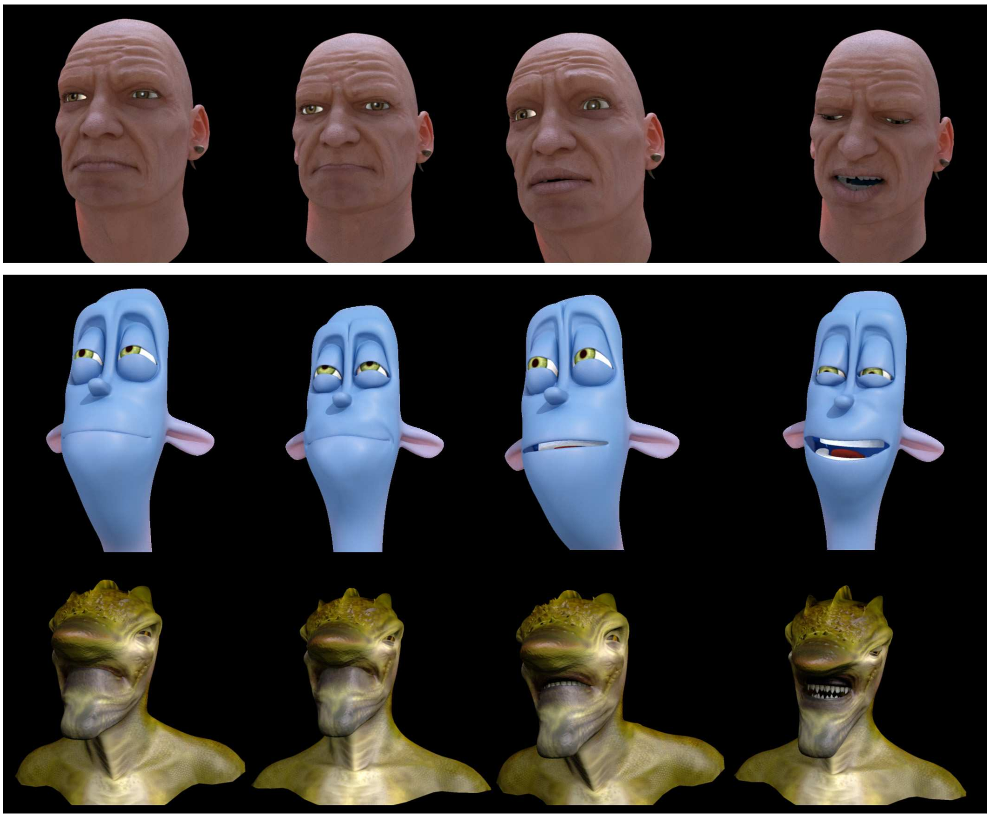


Figure 5.12: Source (top) and target models (bottom). Keyframes extracted from a video sequence to show different poses transferred from the source to the target models. The poses were created by manipulating the source rig. (Copyright 2007 Face In Motion)

Figure 5.12 shows key poses based primarily on a combination of blend shapes. We focused our attention in the quality of the animations, specially, if the movements corresponded to the behavior the targets were supposed to perform. The transfer was successful for the fantastic creature target. But, after analyzing the results with the Technical Director, we concluded that the

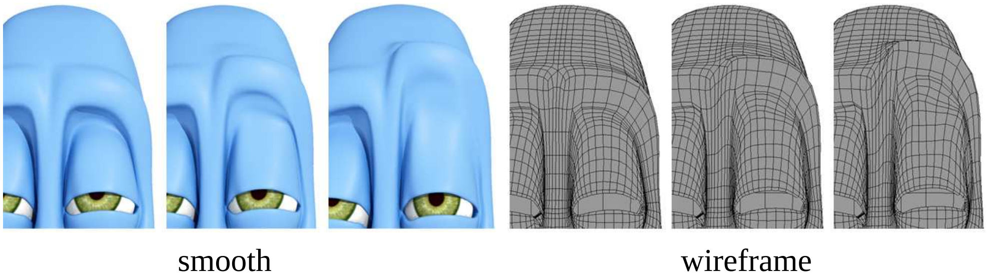


Figure 5.13: Shape transfer from the source model to the cartoon target model: neutral pose (left), shape (middle) and exaggerated shape (right) (Copyright 2007 Face In Motion)

animation transferred from the source to the cartoon target was correct, but the model lacked the movement exaggeration expected by a cartoon design. So we had to develop an additional module in our system that allows defining a percentage value for exaggerating the movements, and applies it to the shapes. Figure 5.13 shows the difference between the "normal" shape and the exaggerated shape.

During the test, we realized that when the source model has the eyes and the mouth completely closed, the attribute transfer results show some artifacts. This a current limitation of our solution. To obtain artifact free transfers, it is recommended to have the eyes and mouth of the source and target models slightly opened.

Last, figure 5.14 shows that our system allows creating believable characters that can be straightforward integrated in a film scene.

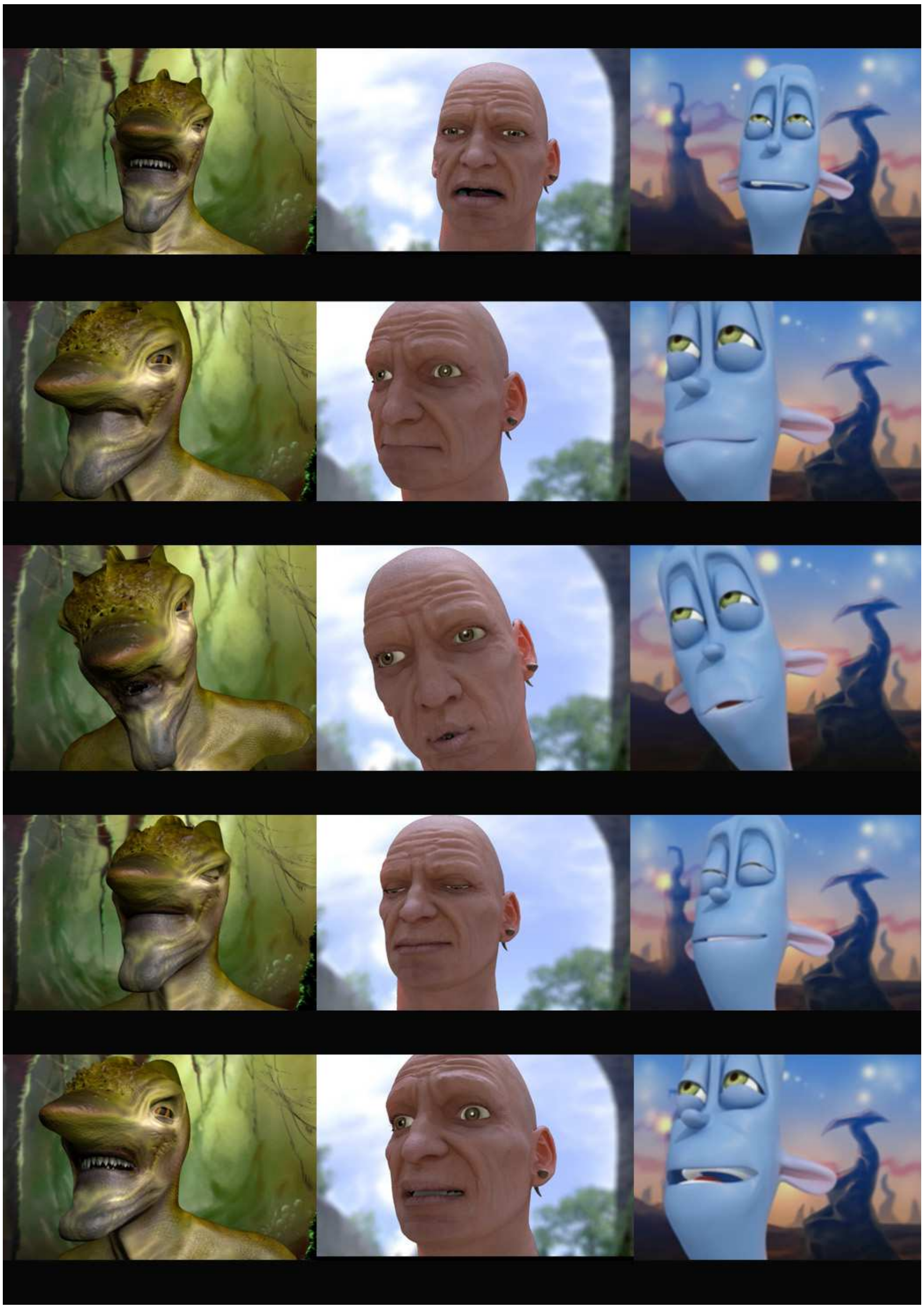


Figure 5.14: Keyframes of the 3D models integrated in a scene: fantastic creature target (left), photorealistic source (middle) and cartoon target (right). (Copyright 2007 Face In Motion)

5.4.3 Timing Validation

The results of this test confirmed that our method was quicker than traditional manual setups. The artists took 132 hours to create the complete source rig and additional 136 hours to create the animations: one clip of different expressions (1min 33s) and another with lip-sync (19s).

Table 5.2 shows the run time of every step of the transfer with the two target models.

	Cartoon	Fantastic
Skin Deformation (TPS)	0.01	0.01
Dense Deformation	1.12	1.03
Joints Transfer	19.05	19.18
Shape Transfer (each)	2.36	2.42
Additional Deformers	15.32	15.05
Weight Transfer	6.10	6.57
Animation Transfer	1.25	1.33

Table 5.2: Application run times of each method step (in seconds). The additional deformers include the lattice and the NURBS curves.

5.5 Rig and Texture Transfer for Videogames

We have also made two tests to validate our system with videogame characters from Electronic Arts (EA). Unfortunately, we cannot show the results of the tests, because the 3D models are still under NDA, as the videogame has not yet been released.

On the first test, the models had a joint based rig and the animations were created with motion capture data. All the facial models had the same number of triangles and a human look. We transferred the joints, the weights and the animations.

But the second test was the interesting one. After seeing a demo of our system, EA wanted to know if our deformation method could be used for texture transfer. Imagine that a videogame has 30 characters and they are all textured. In the middle of the production a different mesh is defined. What would happen to the 3D models that had already been textured? Usually, the artists would need to create all the textures again, because the textures UV coordinates would not work for the new model mesh. The test showed that our system can overcome this problem and avoid retexturing each character by hand.

We started with a source model with texture and a target model without texture. The models had different amount of vertices, shapes and proportions. The target model had a better deformable mesh. The method deformed the target model to match the source model topology and transferred the texture of the source to the target model by reassigning the source model's UV coordinates. We successfully based the texture transfer on our rig transfer method.

5.6 Validation From the Experts

After the testing phase, we inquired several people involved in the validation process of our system to describe the benefits it brings to their productions and animation pipelines.

Which are the main contributions, for the film and videogame industry, of a system that automates the facial rigging process?

Jean-Luc Duprat: *"Improves work flow: decouples the work of the animators and the modelers, they can be working in parallel on the same character. This leads to fewer bottlenecks which increases productivity and reduces cost."*

Dave Fracchia: *"Speed/efficiency: replaces manual creation, reduces errors, etc.*

Consistency/reuse: ensures character heads rigged in the same way allowing for animation reuse amongst various characters."

Juan Nouche: *"In my opinion there are two main contributions:*

1. Improves the quality of the output and saves costs and effort. The system enables savings on specialized workforce for the character setup process, which saves time and money, and this extra time can be used to improve the production quality.

2. Increases the competitivity of the production studio. Considering the first contribution and the fact that the system is prone to be integrated into the studio's production pipeline, it allows for an important improvement of the productivity of the studio, which represents a competitive advantage.

Xenxo Alvarez: *"Reduction of the production times while maintaining homogeneous quality of the facial rigs, with the consequential economic savings."*

Frederick Fowles: *"The big win for us with the system would be a perfect transfer of joint position and weighting with little interaction in the process, so a junior artist could easily set up new characters."*

Which are the main challenges the system has to overcome?

Jean-Luc Duprat: *"Different topologies are incredibly difficult to match accurately. Success depends on finding the right heuristics, and the ones you came up with do work!"*

Dave Fracchia: *"Performance (acting): does the rig adequately capture the range of motion required by the character?*

Flexibility: can the rig be tweaked/modified easily to suit a particular character (often, main characters have their own unique rigs)? Ties into performance;

Speed of use (for videogames): is the rig lightweight enough for real-time control?

Often automated solutions are relegated to secondary characters, where similar rigs/models are used. The trick is having a system which would work well for primary characters. Think about how well your system would be for creating a great rig for the Shrek characters (Shrek, Donkey, Fiona). This is the real challenge."

Xenxo Alvarez: *"The integration of the system in any type of production pipeline: animation films or videogames."*

Which are the main benefits of using our rigging system in current animation pipelines? (Facial rig and animation transfer)

Jean-Luc Duprat: *"Workflow as mentioned above."*

Xenxo Alvarez: *"Speed of the production processes without significant/excessive quality loss"*

Frederick Fowles: *"Setting up new characters for facial animation, making it a very streamlined process. And helping with a topology change mid production.*

[To transfer animations between 3D models] it would be a great solution in our NIS (Non Interactive Sequence), so a character could be animated with a stand in until the model of the intended character is ready."

Crystal Wang: *"I think the method you're doing right now is fine. I'm a lot more interested in perserving the shape and details on the face than whether or not each vertex is snapped to each other. Because it's the overall end look and likeness that's the most important.*

The result is quite good. I'm happy with how well the tool preserves the entire face shape. (...) We can always come back to tweak the deformation details later."

5.7 Other Discussions

Besides validating our system with the industry players, we discussed other topics regarding facial animation, trying to find out where it is heading and the new trends, but also, to look for future avenues of research.

Would it be useful to include a muscle system to animate face models?

Jean-Luc Duprat: *"I don't think so personally."*

Xenxo Alvarez: *"It depends of the type of character. For film animation and special effects it might be possible, while for videogames i think we are still faraway due to the limitations of the current hardware. In either case, a facial muscle system would improve realism."*

Do you think that in a near future most facial animation will be done with motion capture? How hard is it to reuse motion capture data with different 3D models?

Jean-Luc Duprat: *"Only for realistic human characters. You couldn't mocap Shrek for example."*

Chapter 6

Conclusion and Future Directions

This thesis describes a new automatic approach for facial rigging, based on facial feature landmarking and a deformation method that adapts the rig and animations between different models. The system generates high quality results and dramatically reduces production time and resource requirements, compared to current animation methodologies. Its modular design allows for easy integration into existing animation pipelines. Our approach can serve as the base technology to assemble advanced solutions for rigging every type of character.

6.1 Conclusion

We present a comprehensive portable character rigging system that speeds up the rigging process within the facial animation pipeline. Allows *creating* the rig, animation controls and scripts for *one model* (source), and *reuse* them in *many* different target models. In film and videogame productions, artists are often given one base model to make all new faces (shapes). Also, it is common that afterwards they are asked to use a different 3D face, because it has improved deformation details or simply looks better. Currently, all shapes need to be remade to reflect the topology of the new face. But our system makes sure that previous work can be transferred and artists time is not wasted.

Our facial animation system can be integrated into existing animation production pipelines. It provides a solid foundation for setting up consistent rigging strategies: at the beginning of a production, artists can define the required rig parameters and afterwards use them as a template for all models. This rig becomes the *building block* for all characters. Our approach helps film and videogame studios overcome the current lack of a standard rigging methodology. It guarantees that all rigs generated by the system produce homogeneous results, ensuring that the models share a common vision and consistent artistic style.

The rig transfer method was tested on human, cartoon and fantastic models that followed entertainment industry requirements, which was essential to prove that the results were suitable for productions demanding high quality. Exhaustive experiments with several companies confirmed the big time savings achieved on the rigging process, usually an order of magnitude faster. We got very positive feedback and some showed interest in integrating the application into their pipelines. We tested the precision of the attribute transfer and the accuracy of the rig created in the target model. We compared our output with the same results manually created by an artist. The Technical and Art Directors approved the quality of the rig and shapes that where automatically created in the target model. This is a crucial result: if the output still requires a lot of tuning, then the system is useless in production.

As the animation results of our work are hard to reproduce in static printed form, please refer to *http://www.FaceInMotion.com* to view the videos and additional material related to this research.

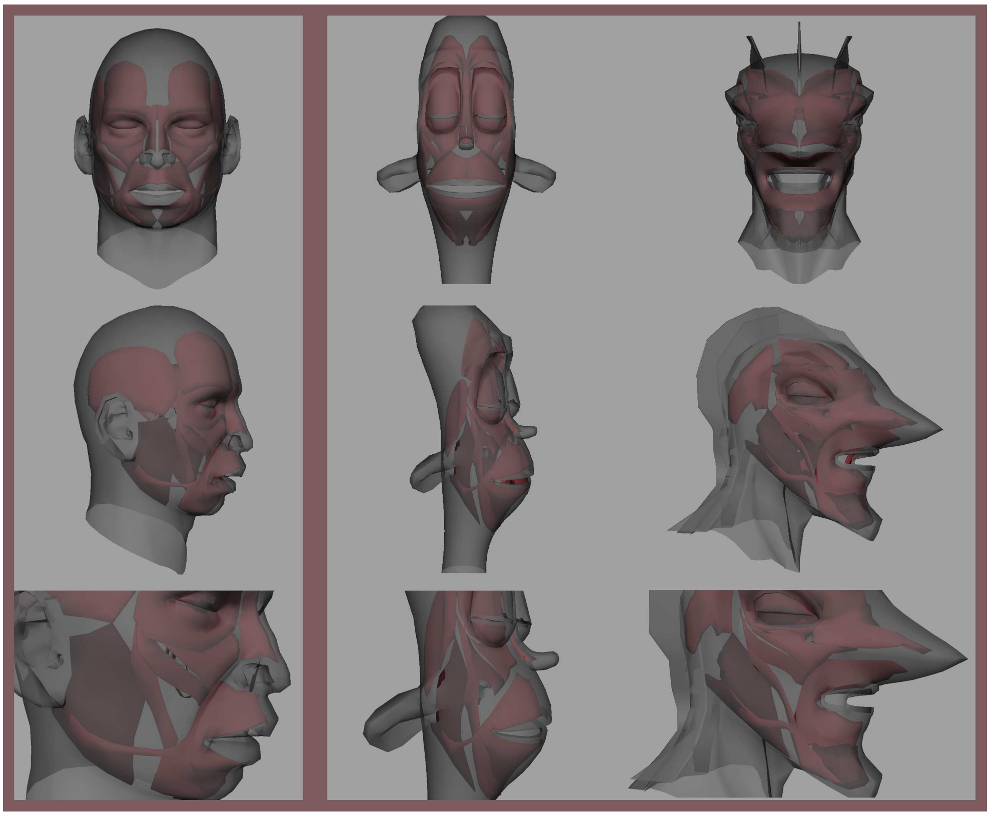


Figure 6.1: Transfer of anatomical muscle geometries from a realistic source model (left) to a cartoon target model and a fantastic creature target model (right). (Copyright 2007 Face In Motion)

6.2 Future Directions

To create a generic facial rigging system, we dealt with the complex subject of geometric deformation between very dissimilar surfaces. There are many interesting directions for future research that can benefit from our work and extend our current system:

Motion capture: motion capture systems are constantly improving. In the near future, it will be easy but costly, both in hardware expenses and production time, to use motion capture to animate realistic human characters. However, it is still going to be considerably difficult to use it for cartoon or fantastic characters, like Shrek. We performed some tests on mapping motion capture data into the source rig, and later transfer it to the target model. We could set up an adaptive version of our method, to capture the performer's movements and map the sensor data into the rig, following work on performance-driven facial animation by [Pighin and Lewis 2006].

Physically-based rig: *could a physically-based method be capable of simulating muscle movement and improve the skin deformation of a model?* [Kahler 2003; Sifakis et al. 2006] have already taken some steps in this direction. Their approach is based on human anatomy and works well for human models. It can be interesting to implement a muscle-based rig that produces believable animations in cartoons and fantastic creatures. The muscles become an additional layer in our rig transfer system. They could be activated by physical reactions of the model or manipulated by the artists. But, it will be a great challenge to provide artists with an easy to use interface. Personally, I don't think that a muscle-based rig could improve the animation quality and the visual results for cartoons and fantastic characters. This opinion was also corroborated and shared by several people from the entertainment industry. Implementing such a system would require extensive research: it is still worth doing, if for nothing else, to be able to compare the results (see figure 6.1 for an example of muscle transfer).

Other industries: our work can serve as the basis for the development of applications in other fields, like medicine, telecommunications, forensics or virtual reality. For instance, virtual reality systems can be used to study the interactions between people and virtual characters. They can help explore human behaviors, such as social phobia, where emotional artificial intelligence plays a key role for understanding the relationship between the character and the person. It would be interesting to develop a layer that links the emotional response of virtual characters with their visual representation. By supporting high quality rigs, this new application would be able to study if better 3D models alter the reactions of the people.

Body rig transfer: clearly, the most obvious, but interesting, new approach is to extend the method to be used on the complete body. Body animation has two main components: the skin of the body and the cloth that goes on top of the skin. In films, the cloth is attached to the skin, so it is only necessary to rig the skin; the cloth is animated using a cloth simulation method [Kang and Lee 2007; Thomaszewski and Blochinger 2007]. However, this technique is too computationally "heavy" to use in videogames; characters usually have a sole rig for both the cloth and the skin, so the transfer method must be prepared to handle cloth wrinkles.

6.2.1 Final Thoughts

Facial rigging is a complex and challenging job due to the great amount of detail involved in facial animation. With this thesis, we have accomplished the difficult task of creating a generic method that allows rig portability between different characters. Now, it is possible to quickly rig a character, control it in real-time and keep the consistency of the rig between models. Rigging is the key element that binds the model to its animation. It is a hot topic, that fueled by the entertainment industry, will continue to provide many research challenges for a long time. We hope our approach and work encourages others to come up with new ideas on this field.

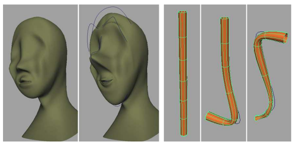
So, the answer to the underlying challenge that sparked our research is *yes... you can (now) use the rig created for one character in other characters.*

Appendix A

Publications and Awards

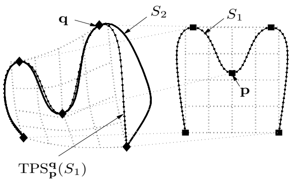
A.1 Publications Overview

These are the most significant publications that resulted from the PhD research, together with a brief description of how the work matured.



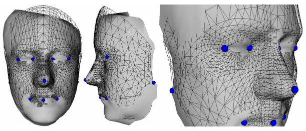
Costa Orvalho, V., and Susin, T. 2005. A Weighted Guide Curve Deformation Algorithm for Facial Animation. In *Technical Report*, UPF, Barcelona, Spain.

This report reimplements the wire deformer from the Maya software. It allows storing the weight deformation value in the curve instead of saving it in the surface, making it easier to control the deformation of different objects over the same surface. The development of the guide curve deformer helped acquire the necessary skills in Maya to develop further plug-ins. The fact that our methods were implemented in Maya was key to demo our system to film and videogame companies.



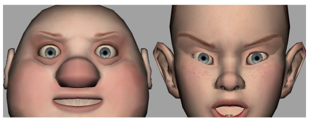
Costa Orvalho, V., Zacur, E., and Susin, T. 2006. Transferring Facial Expressions to Different Face Models. In *SIACG '06: Third Ibero-American Symposium in Computer Graphics*, Santiago de Compostela, Spain.

This paper presents a geometric deformation method based on thin-plate splines [Bookstein 1989]. Uses a set of anthropometric landmarks to keep the correspondence between the models. Results are limited to adapt polygonal meshes between face models. The paper describes the potential of the geometric deformation method for facial synthesis. When we wrote it, we didn't count with the support of any film or videogame company that could provide material to test our solution.



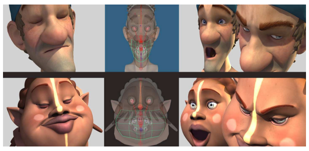
COSTA ORVALHO, V., ZACUR, E., AND SUSIN, T. 2006. A Maya Plugin to Transfer Facial Expressions to Different Face Models. In *iDiG '06: International Digital Games Conference*, Portalegre, Portugal.

This paper describes the implementation of our geometric deformation method as plug-in for Maya. It emphasizes how a tool that speeds up the character setup process can benefit the videogame industry. The paper details the internal architecture of the plug-in for rigging a face model.



COSTA ORVALHO, V., ZACUR, E., AND SUSIN, T. 2007. Transferring the Rig and Animations from a Character to Different Face Models. *Computer Graphics Forum*, EG, Blackwell Publishing. (Submitted for review)

This paper shows the first results of transferring the facial rig using 3D models from a film company (Dygrafilms). It explains the importance, for the film industry, of a system that can transfer the facial rig and animations between characters faster than traditional manual techniques. These tests, along with the feedback from the company, gave us further insight into the needs of the industry. It lead us to focus our research on optimizing the rigging system and improving the modularity of the application, to ease the integration into existing animation pipelines.



COSTA ORVALHO, V., AND SUSIN, T. 2007. Fast and Reusable Facial Rigging and Animation. In *SIGGRAPH '07: Sketches & Posters*, The 34th International Conference and Exhibition on Computer Graphics and Interactive Techniques, San Diego, USA, ACM Press. (Accepted)

This sketch best summarizes the contribution of our work and its importance for the entertainment industry. It describes the results of exhaustive testing and validation of our work with different film and videogame companies. It gives an overview of the framework implemented in Maya 7.0 that assists the rig transfer process.

A.2 Publications and Conferences

[2007]

[1] Costa Orvalho, V., and Susin, T. 2007. Fast and Reusable Facial Rigging and Animation. In *SIGGRAPH '07: Sketches & Posters*, The 34th International Conference and Exhibition on Computer Graphics and Interactive Techniques, San Diego, USA, ACM Press. (Accepted)

[2] Costa Orvalho, V., and Susin, T. 2007. Creating Faces: Reusing Character Rig and Animations. *SCA '07: Eurographics/ACMSIGGRAPH Symposium on Computer Animation*, San Diego, USA, ACM Press. (Accepted)

[3] Costa Orvalho, V., Zacur, E., and Susin, T. 2007. Transferring the Rig and Animations from a Character to Different Face Models. *Computer Graphics Forum*, EG, Blackwell Publishing. (Submitted for review)

[4] Costa Orvalho, V., and Susin, T. 2007. Creating the Virtual Self: Reusing Facial Animation in Different Avatars, *In SIGPHI '07: Mundos Digitales*, A Coruña, Spain.

[2006]

[5] Costa Orvalho, V., Zacur, E., and Susin, T. 2006. A Maya Plugin to Transfer Facial Expressions to Different Face Models. In Proceedings *iDiG '06: International Digital Games Conference*, Portalegre, Portugal. Editors Dionisio J., Ramires Fernades A., Proceedings iDiG '06, pp 12-20 (2006).

[6] Costa Orvalho, V., Zacur, E., and Susin, T. 2006. Transferring a Labeled Generic Rig to Animate Face Models. *In AMDO '06: 4th International Conference on Articulated Motion and Deformable Objects*, Springer Press, Port d'Andratx, Mallorca, Spain. Proceedings AMDO'06, Ed. Springer Lecture Notes, pp 21-31 (2006).

[7] Costa Orvalho, V., Zacur, E., and Susin, T. 2006. Transferring Facial Expressions to Different Face Models. In *SIACG '06: Third Ibero-American Symposium in Computer Graphics*, Santiago de Compostela, Spain. Editors Brunet P., Correia N., Baranoski G. Proceedings SIACG'06, pp 13-20 (2006).

[2005]

[8] Costa Orvalho, V., and Susin, T. 2005. A Weighted Guide Curve Deformation Algorithm for Facial Animation. In *Technical Report*, UPF, Barcelona, Spain.

A.3 Invited Talks and Workshops

[2007]

Workshop (24h), *Videogame Design and Development*, FCUP (Faculdade de Ciências da Universidade do Porto), Porto, Portugal, July 2007.

Invited Speaker, *Facial Synthesis and The Medical Industry*, FCUP (Faculdade de Ciências da Universidade do Porto), Porto, Portugal, June 2007.

Invited Panelist, *PhD Symposium for the PRESENCCIA project*, UPC (Universitat Politècnica de Catalunya), Barcelona, Spain, May 2007.

Invited Speaker, *A Facial Animation System for CG Films and Videogames*, FCUP (Faculdade de Ciências da Universidade do Porto), Porto, Portugal, February 2007.

Invited Speaker, *A Facial Animation System for CG Films and Videogames*, FEUP (Faculdade de Engenharia da Universidade do Porto), Porto, Portugal, February 2007.

[2006]

Invited Speaker, *Facial Animation for Films and Videogames Project Presentation*, Electronic Arts, Vancouver, Canada, November 23rd, 2006.

Invited Speaker, *Facial Animation for Films and Videogames Project Presentation*, Mainframe-Rainmaker,Vancouver, Canada, November 17th, 2006.

Invited Speaker, *Facial Animation for Films and Videogames Project Presentation*, Mercury Filmworks, Vancouver, Canada, November 17th, 2006.

Invited Speaker, *Facial Animation for Films and Videogames Project Presentation*, Radical Entertainment, Vancouver, Canada, November 9th, 2006.

Invited Speaker, *A Facial Animation System For CG Films and Videogames*, AMORE, University of British Columbia, Vancouver, Canada, October 18th, 2006.

Invited Speaker, *A Facial Animation System For CG Films and Videogames*, HCT Lab, University of British Columbia, Vancouver, Canada, October 16th, 2006.

[2005]

Workshop, *Videogame Design and Content Development for Videogame*, Faculdade de Jornalismo, Universidade Porto, Porto, Portugal, October 2005.

Invited Speaker, *The Videogame Industry a Short Overview*, Universidade Católica Portuguesa, Porto, Portugal, April 2005.

Workshop, *Videogame Design*, Universidade Católica Portuguesa, Porto, Portugal, April 2005.

[2004]

Invited Speaker, *Videogame Technology Overview*, Instituto Superior de Tecnologias Avançadas (ISTEC), Porto, Portugal, December 2004.

Workshop, *Introduction to Videogame Design*, Instituto Superior de Tecnologias Avançadas (ISTEC), Porto, Portugal, December 2004.

Invited Speaker, *Videogame Technology Overview*, Instituto Superior de Tecnologias Avançadas (ISTEC), Lisboa, Portugal, December 2004.

Workshop, *Introduction to Videogame Design*, Instituto Superior de Tecnologias Avançadas (ISTEC), Lisboa, Portugal, December 2004.

Workshop, *Videogame Technology*, Instituto Superior de Tecnologias Avançadas (ISTEC), Porto, Portugal, September 2004.

Workshop, *Videogame Design*, Instituto Superior de Tecnologias Avançadas (ISTEC), Porto, Portugal, September 2004.

A.4 Honors and Awards

These awards were for business plans related to our PhD research work.

Winner "AlumnIdea 2006". Organized by Alumni LEIC, FEUP in Portugal. September 2006. Project: *Face In Motion*. (http://alumnidea.fe.up.pt/)
Prize: 5.000 EUR

Finalist award in "Innovact 2006". Organized by L'Etudiant et L'Express in France. July 2006. Project: *Face Puppet, Automatic Facial Animation*. (http://www.innovact.com/)

3rd Place in the "IV Concurso Innova" for young entrepreneurship. Organized by Universitat Politècnica de Catalunya (UPC), EAE and CIDEM (Generalitat Catalunya). June 2004. Project: *Photorealistic Facial Animation and Recognition*. (http://pinnova.upc.es/)
Prize: 1.200 EUR

Finalist award in "INNOVACT 2004". Organized by L'Etudiant et L'Express in France. July 2004. Project: *Photorealistic Facial Animation System*. (http://www.innovact.com/)

A.5 Grants

CIDEM (Centre Innovacio i Desenvolupament Empresaria): 9.376,79 EUR, Programa GENESIS (http://www.cidem.com). Project: *Support the creation of content and the development of a prototype to test and validate the facial system technology*. Generalitat de Catalunya. Co-financed by the European Union - FEDER. January 2007.

AdI (Agência de Inovação): 100.000 EUR, Inicitiva Neotec (http://www.neotec.gov.pt/). Project: *A Facial Animation System for CG Films and Videogames*. POSC, MCTES (Ministério da Ciência, Tecnologia e do Ensino Superior). Co-financed by the European Union - FEDER. July 2006.

References

AMIDROR, I. 2002. Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging 11*, 2, 157–76. Lab. de Syst. Peripheriques, Ecole Polytech. Fed. de Lausanne, Switzerland.

ANDERSON, K., AND MCOWAN, P. 2006. A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36*, 1, 96–105.

BADLER, N. I., AND PLATT, S. M. 1981. Animating facial expressions. In *Proc. SIGGRAPH'81*, 15(3):245–252.

BALCI, K. 2004. Xface: Mpeg-4 based open source toolkit for 3d facial animation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, ACM Press, New York, NY, USA, 399–402.

BASU, S., OLIVER, N., AND PENTLAND, A. 1998. 3d modeling of human lip motion. In *ICCV*, 337–343.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 187–194.

BLANZ, V., BASSO, C., POGGIO, T., AND VETTER, T. 2003. Reanimating faces in images and video. In *EUROGRAPHICS 2003 the European Association for Computer Graphics, 24th Annual Conference*, Blackwell, Granada, Spain, P. Brunet and D. W. Fellner, Eds., vol. 22 of *Computer Graphics Forum*, The Eurographics Association, 641–650.

BOOKSTEIN, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell. 11*, 6, 567–585.

BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J. P., AND TEMPELAAR-LIETZ, C. 2003. Universal capture: image-based facial animation for "the matrix reloaded". In *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications*, ACM Press, New York, NY, USA, 1–1.

BORSHUKOV, G., PIPONI, D., LARSEN, O., LEWIS, J. P., AND TEMPELAAR-LIETZ, C. 2005. Universal capture - image-based facial animation for "the matrix reloaded". In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, ACM Press, New York, NY, USA, 16.

BORSHUKOV, G., MONTGOMERY, J., AND WERNER, W. 2006. Playable universal capture: compression and real-time sequencing of image-based facial animation. In *SIGGRAPH '06: ACM SIGGRAPH Courses*, ACM Press, New York, NY, USA, 8.

BREGLER, C., COVELL, M., AND SLANEY, M. 1997. Video rewrite: Driving visual speech with audio. SIGGRAPH 1997, 353–360.

CARR, J., FRIGHT, W., AND BEATSON, R., 1997. Surface interpolation with radial basis functions for medical imaging.

CASSELL, J., PELACHAUD, C., BADLER, N., STEEDMAN, M., ACHORN, B., BECKET, T., DOUVILLE, B., PREVOST, S., AND STONE, M. 1994. Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 413–420.

CHADWICK, J., HAUMANN, D., AND PARENT, R. 1989. Layered construction for deformable animated characters. In *SIGGRAPH '89: Proc. of Conf. on Computer graphics and interactive techniques*, ACM Press, NY, USA, 243–252.

CHAI, J., HORVATH, V., KAMBHATLA, N., NICOLOV, N., AND STYS-BUDZIKOWSKA, M. 2001. A conversational interface for online shopping. In *HLT '01: Proceedings of the first international conference on Human language technology research*, Association for Computational Linguistics, Morristown, NJ, USA, 1–4.

CHAI, J. X., XIAO, J., AND HODGINS, J. 2003. Vision-based control of 3d facial animation. In *SCA '03: Proc. of ACM SIGGRAPH/Eurographics Symp. on Comp. animation*, ACM Press, Aire-la-Ville, Switzerland, 193–206.

CHANG, E., AND JENKINS, O. C. 2006. Sketching articulation and pose for facial animation. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 271–280.

CHOE, B., AND KO, H.-S. 2001. Analysis and synthesis of facial expressions with hand-generated muscle actuation basis. In *Computer Animation*.

CHOE, B., LEE, H., AND KO, H.-S. 2001. Performance-driven muscle-based facial animation. *The Journal of Visualization and Computer Animation 12*, 2 (May), 67–79.

CHUANG, E., AND BREGLER, C. 2002. Performance driven facial animation using blendshape interpolation. In *Computer Science Technical Report*.

COHEN, E., RIESENFELD, R. F., AND ELBER, G. 2001. *Geometric modeling with splines: An introduction*. A. K. Peters, Ltd, Natick, MA, USA.

COQUILLART, S. 1990. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 187–196.

DECARLO, D., METAXAS, D., AND STONE, M. 1998. An anthropometric face model using variational techniques. In *SIGGRAPH '98: Proc. of Conf. on Computer graphics and interactive techniques*, ACM Press, NY, USA, 67–74.

DENG, Z., CHIANG, P.-Y., FOX, P., AND NEUMANN, U. 2006. Animating blendshape faces by cross-mapping motion capture data. In *SI3D '06: Proc. of Symp. on Interactive 3D graphics and games*, ACM Press, New York, NY, USA, 43–48.

EKMAN, P., AND FRIESEN, W. V. 1975. *Unmasking the Face. A guide to recognizing emotions fron facial clues*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

EKMAN, P., AND FRIESEN, W. 1978. *Facial Action Coding System*. Consulting Psychologist Press, Palo Alto, CA, USA.

ESSA, I., AND BASU, S., 1996. Modeling, tracking and interactive animation of facial expressions and head movements using input from video.

ESSA, I., AND PENTLAND, A., 1997. Coding, analysis, interpretation, and recognition of facial expressions.

FALK, R., MINTER, D., VERNON, C., ARETOS, G., MODESTO, L., LAMORLETTE, A., WALKER, N., CHEUNG, T., RENTEL-LAVIN, J., AND MAX, H. 2004. Art-directed technology: anatomy of a shrek2 sequence. In *SIGGRAPH '04: ACM SIGGRAPH Course Notes*, ACM Press, NY, USA, 13.

FARKAS, L., AND MUNRO, I. 1987. Anthropometric facial proportions in medicine. Charles Thomas ltd., Springfield, IL, USA.

FIDALEO, D., NOH, J. Y., KIM, T., ENCISO, R., AND NEUMANN, U. 2000. Classification and volume morphing for performance-driven facial animation. In *Proc. of International Workshop on Digital and Computational Video*.

FORDHAM, J. 2003. Middle earth strikes back. In *Cinefex*, Don Shay, Riverside, CA, USA, vol. 92, 71–142.

FORDHAM, J. 2006. Beneath the barnacles. In *Cinefex*, Don Shay, Riverside, CA, USA, vol. 107, 67–89.

FORDHAM, J. 2006. Return of the king kong. In *Cinefex*, Don Shay, Riverside, CA, USA, vol. 104, 42–81.

FRASER, N. 1997. Assessment of interactive systems. In *Handbook of Standards and Resources for Spoken Language Systems*, Eds. Mouton de Gruyter, New York, NY, USA, 564–614.

FRATARCANGELI, M., SCHAERF, M., AND FORCHHEIMER, R. 2007. Facial motion cloning with radial basis functions in mpeg-4 fba. *Graph. Models 69*, 2, 106–118.

GLEICHER, M. 1998. Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 33–42.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 1998. Making faces. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 55–66.

GUENTER, B., GRIMM, C., WOOD, D., MALVAR, H., AND PIGHIN, F. 2005. Making faces. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, ACM Press, New York, NY, USA, 10.

GULZ, A., AND HAAKE, M. 2006. Design of animated pedagogical agents-a look at their look. *International Journal of Human-Computer Studies 64*, 4, 322–339.

HABER, J., TERZOPOULOS, D., PARKE, F., WILLIAMS, L., BLANZ, V., AND BORSHUKOV, G. 2004. Facial modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH Course Notes*, ACM Press, New York, NY, USA, 6.

HANSON, D., OLNEY, A., PEREIRA, I. A., AND ZIELKE, M. 2005. Upending the uncanny valley. *AAAI '05, National Conference on Artificial Intelligence*.

HARATASCH, E., AND OSTERMANN, J. 1997. An animation definition interface: Rapid design of mpeg-4 compilant animated faces and bodies. International Workshop on Synthetic-Natural Hybrid Coding and 3D Imaging, Rhodes, Greece, 216–219.

HAVALDAR, P. 2006. Sony pictures imageworks. In *SIGGRAPH '06: ACM SIGGRAPH Courses*, ACM Press, NY, USA, 5.

HILGER, K. B., PAULSEN, R. R., AND LARSEN, R. 2004. Markov random field restoration of point correspondences for active shape modelling. In *SPIE - Medical Imaging*.

HSU, W., HUGUES, J., AND KAUFMAN, H. 1992. Direct manipulation of free-form deformation. ACM Press, Proc. SIGGRAPH '92, 177–184.

HUTTON, T. J., BUXTON, B. F., AND HAMMOND, P., 2001. Dense surface point distribution models of the human face.

HYNEMAN, W., ITOKAZU, H., WILLIAMS, L., AND ZHAO, X. 2005. Human face project. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, ACM Press, New York, NY, USA, 5.

IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. Spatial keyframing for performance-driven animation. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 107–115.

IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2006. Spatial keyframing for performance-driven animation. In *SIGGRAPH '06: ACM SIGGRAPH Courses*, ACM Press, NY, USA, 17.

ISHIKAWA, T., SERA, H., MORISHIMA, S., AND TERZOPOULOS, D. 1998. Facial image reconstruction by estimated muscle parameter. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, IEEE Computer Society, Washington, DC, USA, 342.

JASON SMITH, J. W. 2006. Blockparty: Modular rigging encoded in geometric volume. In *SIGGRAPH '06: Sketch*, ACM Press, New York, NY, USA.

JOSHI, P., TIEN, W., DESBRUN, M., AND PIGHIN, F. 2003. Learning controls for blend shape based realistic facial animation. In *SCA '03: Proc. of ACM SIGGRAPH/Eurographics Symp. on Computer animation*, ACM Press, Aire-la-Ville, Switzerland, 187–192.

KAHLER, K., HABER, J., AND SEIDEL, H., 2001. Geometry-based muscle modeling for facial animation.

KAHLER, K., HABER, J., YAMAUCHI, H., AND SEIDEL, H.-P. 2002. Head shop: generating animated head models with anatomical structure. In *SCA '02: Proc. of ACM SIGGRAPH/Eurographics Symp. on Computer animation*, ACM Press, New York, NY, USA, 55–63.

KAHLER, K., HABER, J., AND SEIDEL, H.-P. 2003. Reanimating the dead: reconstruction of expressive faces from skull data. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM Press, New York, NY, USA, 554–561.

KAHLER, K. 2003. *A Head Model with Anatomical Structure for Facial Modeling and Animation.* PhD. Thesis, Universitat des Saarlandes.

KALRA, P., MANGILI, A., MAGNENAT-THALMANN, N., AND THALMANN, D. 1992. Simulation of facial muscle actions based on rational free form deformations. A. Kilgour and L. Kjelldahl, Eds., vol. 11, 59–69.

KANG, M. K., AND LEE, J. 2007. Technical section: A real-time cloth draping simulation algorithm using conjugate harmonic functions. *Comput. Graph. 31*, 2, 271–279.

KAVAN, L., AND ZÁRA, J. 2005. Spherical blend skinning: a real-time deformation of articulated models. In *SI3D '05: Proc. of Symp. on Interactive 3D graphics and games*, ACM Press, New York, NY, USA, 9–16.

KERLOW, I. 2004. Creative human character animation: The incredibles vs. the polar express. In *VXF World*.

KOENEN, R. 2002. Overview of the mpeg-4 standard. www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm.

KOHNERT, C. J., AND SEMWAL, S. K. 2006. Automatic lip-synchronization using linear prediction of speech. In *SPPRA'06: Proceedings of the 24th IASTED international conference on Signal processing, pattern recognition, and applications*, ACTA Press, Anaheim, CA, USA, 210–215.

KONDO, R., KANAI, T., AND ICHI ANJYO, K. 2005. Directable animation of elastic objects. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 127–134.

KOUADIO, C., POULIN, P., AND LACHAPELLE, P. 1998. Real-time facial animation based upon a bank of 3d facial expressions. In *Computer Animation 98*, IEEE Computer Society, 128–136.

KRENN, B., NEUMAYR, B., GSTREIN, M., AND GRICE, M. 2004. Lifelike agents for the internet: A cross-cultural case study. *In Agent Culture: Human-Agent Interaction in a Multicultural World, PAYR, S AND TRAPPL, R. Eds. Lawrence Erlbaum Associates, NJ*, 197–229.

LARBOULETTE, C., CANI, M.-P., AND ARNALDI, B. 2005. Dynamic skinning: adding real-time dynamic effects to an existing character animation. In *SCCG '05: Proceedings of the 21st spring conference on Computer graphics*, ACM Press, New York, NY, USA, 87–93.

LASSETER, J. 1987. Principles of traditional animation applied to 3d computer animation. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 35–44.

LAYBOURNE, K. 1998. *The Animation Book : A Complete Guide to Animated Filmmaking–From Flip-Books to Sound Cartoons to 3-D Animation*. Three Rivers Press, New York, USA.

LAZARUS, F., COQUILLART, S., AND JANCENE, P. 1994. Axial deformations: an intuitive deformation technique. Computer-Aided Design, 607–613.

LEE, Y., TERZOPOULOS, D., AND WATERS, K. 1993. Constructing physics-based facial models of individuals. In *Proceedings of Graphics Interface '93*, 1–8.

LEE, Y., TERZOPOULOS, D., AND WALTERS, K. 1995. Realistic modeling for facial animation. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 55–62.

LESTER, J., BRANTING, K., AND MOTT, B. 2004. Conversational agents. CRC Press LLC.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proc. of Conf. on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Co., NY, USA, 165–172.

LITWINOWICZ, P., AND WILLIAMS, L. 1994. Animating images with drawings. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 409–412.

LORENZ, C., AND KRAHNSTOVER, N. 2000. Generation of point-based 3D statistical shape models for anatomical objects. *Computer Vision and Image Understanding: CVIU 77*, 2, 175–191.

LOYALL, A. B., AND BATES, J. 1997. Personality-rich believable agents that use language. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, ACM Press, New York, NY, USA, 106–113.

LU, X., AND JAIN, A. K. 2005. Deformation analysis for 3d face matching. In *WACV-MOTION '05: Proc. of IEEE Workshops on Application of Computer Vision - Vol 1*, IEEE, Washington, DC, USA, 99–104.

MACCRACKEN, R., AND JOY, K. I. 1996. Free-form deformations with lattices of arbitrary topology. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 181–188.

MAGNENAT-THALMANN, N., E.PTIMRSU, AND THALMANN, D. 1988. Abstract muscle action procedures for human face animation. The Visual Computer, 3(5):290–297.

MARAFFI, C. 2003. *Maya Character Cretion: Modeling and Animation Controls*. New Riders Publishing.

MCTEAR, M. F. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM Comput. Surv. 34*, 1, 90–169.

MILLER, J. 2006. Inside universal capture. In *IGN Entertainment, Inc.*, sports.ign.com, 1.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM Press, New York, NY, USA, 562–568.

MÖLLER, T., AND HAINES, E. 1999. *Real-Time Rendering*. A. K. Peters. MÖL t 02:1 1.Ex.

MORI, M. 1982. *The Buddha in the Robot.* C. E. Tuttle.

NOH, J., AND NEUMANN, U., 1998. A survey of facial modeling and animation techniques.

NOH, J. Y., AND NEUMANN, U. 2001. Expression cloning. In *SIGGRAPH '01: Proc. of Conf. on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 277–288.

OCHS, M., NIEWIADOMSKI, R., PELACHAUD, C., AND SADEK, D. 2005. Intelligent expressions of emotions. *Affective Computing and Intelligent Interaction*, 707–714.

OSTERMANN, J. 1998. Animation of synthetic faces in mpeg-4. Computer Animation, Philadelphia, USA.

PANDZIC, I. S., AND FORCHHEIMER, R. 2002. The origins of the mpeg-4 facial animation standard. In *MPEG-4 Facial Animation, The Standrad, Implementation and Application.*, John Wiley & Sons Ltd.

PANDZIC, I. S. 2003. Facial motion cloning. *Graph. Models 65*, 6, 385–404.

PARKE, F. I., AND WATERS, K. 1996. *Computer Facial Animation*. A. K. Peters, Ltd., Natick, MA, USA.

PARKE, F. I. 1972. Computer generated animation of faces. In *ACM'72: Proc. of the ACM annual Conf.*, ACM Press, New York, NY, USA, 451–457.

PARKE, F. I. 1974. A parametric model for human faces. PhD thesis University of Utah, Salt Lake City, Utah.

PARKE, F. I. 1982. Parametrized models for facial animation. IEEE Computer Graphics and Applications, 2(9):61–68.

PAVEY, E. 2007. Ready fro their close-ups. In *IEEE Spectrum*.

PELACHAUD, C., AND POGGI, I. 1998. Multimodal communication between synthetic agents. In *AVI '98: Proceedings of the working conference on Advanced visual interfaces*, ACM Press, New York, NY, USA, 156–163.

PELACHAUD, C. 2005. Multimodal expressive embodied conversational agents. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, ACM Press, New York, NY, USA, 683–689.

PIGHIN, F., AND LEWIS, J. P. 2006. Facial motion retargeting. In *SIGGRAPH '06: ACM SIGGRAPH Courses*, ACM Press, New York, NY, USA, 2.

PIGHIN, F. H., SZELISKI, R., AND SALESIN, D. 1999. Resynthesizing facial animation through 3d model-based tracking. In *ICCV (1)*, 143–150.

PIGHIN, F., HAVALDAR, P., TOLLES, T., LEWIS, J. P., BREGLER, C., ZHANG, L., SULLIVAN, S., HERY, C., SAGAR, M., KANG, T., AND BORSHUKOV, G. 2006. Performance-driven facial animation. In *SIGGRAPH '06: ACM SIGGRAPH Courses*, ACM Press, NY, USA, 1.

PLATT, S. M., AND BADLER, N. I. 1981. Animating facial expressions. In *SIGGRAPH '81: Proc. of Conf. on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 245–252.

PRO, P. 2006. Seven days in second life. In *PC Pro: Computing in the real world*, http://www.pcpro.co.uk.

PYUN, H., KIM, Y., CHAE, W., KANG, H. W., AND SHIN, S. Y. 2006. An example-based approach for facial expression cloning. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, ACM Press, New York, NY, USA, 23.

REVERET, L., AND ESSA, I. 2001. Visual coding and tracking of speech related facial motion. In *IEEE CVPR: International Workshop on Cues in Communication*.

RICHIE, K., ALEXANDER, O., AND BIRI, K. 2005. *The Art of Rigging Vol. 2*. CG Toolkit.

ROHR, K., STIEHL, H., SPRENGEL, R., BUZUG, T., WEESE, J., AND KUHN, M. 2001. Landmark-based elastic registration using approximating thin-plate splines. *Medical Imaging, IEEE Transactions on 20*, 6, 526–534.

RUSS, J. 2004. Inside the incredibles. In *Computer Arts*.

SAGAR, M. 2006. Facial performance capture and expressive translation for king kong. In *SIGGRAPH '06: ACM SIGGRAPH Courses*, ACM Press, NY, USA, 1.

SCHLEIFER, J. 2002. Character setup from rig mechanics to skin deformations: A practical approach. In *SIGGRAPH '02: ACM SIGGRAPH 2002 Course Notes*, ACM Press, New York, NY, USA.

SEDERBERG, T., AND PARRY, S. 1986. Free-form deformation of solid geometric models. In *SIGGRAPH '86: Proc. of Conf. on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 151–160.

SEN TANG, S., LIEW, A. W.-C., AND YAN, H. 2004. Lip-sync in human face animation based on video analysis and spline models. In *MMM '04: Proceedings of the 10th International Multimedia Modelling Conference*, IEEE Computer Society, Washington, DC, USA, 102.

SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. In *SIGGRAPH '05: ACM SIGGRAPH Papers*, ACM Press, New York, NY, USA, 417–425.

SIFAKIS, E., SELLE, A., ROBINSON-MOSHER, A., AND FEDKIW, R. 2006. Simulating speech with a physics-based facial muscle model. In *ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, edited by M. Cani, J. O'Brien, 261–270.

SINGH, K., AND FIUME, E. L. 1998. Wires: a geometric deformation technique. ACM Computer Graphics, Proc. SIGGRAPH '98 Conf., 405–414.

SINGH, K., OHYA, J., AND PARENT, R. 1995. Human figure synthesis and animation for virtual space teleconferencing. Virtual Reality Annual International Symp. '95. IEEE, 118–126.

SLOAN, P.-P. J., III, C. F. R., AND COHEN, M. F. 2001. Shape by example. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, ACM Press, New York, NY, USA, 135–143.

STRONG, C. R., MEHTA, M., MISHRA, K., JONES, A., AND RAM, A. 2007. Emotionally driven natural language generation for personality rich characters in interactive games. In *AIDE'07: Third Conference on Artificial Intelligence for Interactive Digital Entertainment*.

TERRA, S. C. L., AND METOYER, R. A. 2004. Performance timing for keyframe animation. In *SCA '04: Proc. of ACM SIGGRAPH/Eurographics Symp. on Computer animation*, ACM Press, New York, NY, USA, 253–258.

TERZOPOULOS, D., AND WATERS, K. 1990. Physically-based facial modelling, analysis, and animation. *The Journal of Visualization and Computer Animation 1*, 2, 73–80.

THOMASZEWSKI, B., AND BLOCHINGER, W. 2007. Physically based simulation of cloth on distributed memory architectures. *Parallel Comput. 33*, 6, 377–390.

TURNER, R., AND THALMANN, D. 1993. The elastic surface layer model for animated character construction. Proc. of Computer Graphics International '93, 399–412.

WANG, X. C., AND PHILLIPS, C. 2002. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, New York, NY, USA, 129–138.

WARD, A. 2004. *Game Character Development with Maya*. New Riders Publishing.

WATERS, K., AND FRISBIE, J. 1995. A coordinated muscle model for speech animation. In *Graphics Interface '95*, 163–170.

WATERS, K., AND LEVERGOOD, T. 1994. An automatic lip-synchronization algorithm for synthetic faces. In *MULTIMEDIA '94: Proceedings of the second ACM international conference on Multimedia*, ACM Press, New York, NY, USA, 149–156.

WATERS, K. 1987. A muscle model for animation three-dimensional facial expression. In *Proc. SIGGRAPH'87*, 17–24.

WATERS, K. 1989. Modeling 3d facial expressions: Tutorial notes. SIGGRAPH '89 Course Notes 22: State of the Art in Facial Animation, 127–152.

WATERS, K. 1992. A physical model of facial tissue and muscle articulation derived from computer tomography data. vol. 1808, SPI Visualization in Biomedical Computing, 574–583.

WEBER, J. 2000. Run-time skin deformation. Game Developers Conference.

WILLIAMS, L. 1990. Performance-driven facial animation. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 235–242.

WU, Y., KALRA, P., AND THALMANN, N. 1996. Simulation of static and dynamic wrinkles of skin. Proc. of Computer Animation '96, 90–97.

YANG, X., AND ZHANG, J. J. 2006. Stretch it - realistic smooth skinning. In *CGIV '06: Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation*, IEEE Computer Society, Washington, DC, USA, 323–328.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, ACM Press, New York, NY, USA, 548–558.

ZHAO, W., CHELLAPPA, R., PHILLIPS, P. J., AND ROSENFELD, A. 2003. Face recognition: A literature survey. *ACM Comput. Surv. 35*, 4, 399–458.