# CROWD MODELING: GENERATION OF A FULLY ARTICULATED CROWD OF CHARACTERS

A Thesis

by

KARTHIK SWAMINATHAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2005

Major Subject: Visualization Sciences

# CROWD MODELING: GENERATION OF A FULLY ARTICULATED CROWD OF CHARACTERS

A Thesis

by

KARTHIK SWAMINATHAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

| | |
|---|---|
| Chair of Committee, | Ergun Akleman |
| Committee Members, | Louis Tassinary |
| | Derya Akleman |
| Head of Department, | Phillip Tabb |

December 2005

Major Subject: Visualization Sciences

# ABSTRACT

Crowd Modeling: Generation of a Fully Articulated Crowd of Characters..

(December 2005)

Karthik Swaminathan, B.Arch, Madras University, India

Chair of Advisory Committee: Dr.Ergun Akleman

In this thesis I present a fast, efficient, and production friendly method to generate a crowd of fully articulated characters. A wide variety of characters can be created from a relatively few base models. The models that are generated are anatomically different from each another, while maintaining the same topology. They all have individual characteristics and features, that distinguish them from the others in the crowd. This method is easily adaptable to different kinds of characters, from hyper-realistic characters to highly stylized characters, and from human characters to insects like spiders. The crowd character models generated by this method are fully articulated and are ready to be animated.

To My Guru: Om Shri Gurubhyo Namah

## ACKNOWLEDGMENTS

I would like to thank the various people who helped, inspired, motivated and mentored me along the way. I am deeply grateful to my parents, without whose blessings and support, I would not be where I find myself today. Thanks to my greatest sources of inspiration and motivation, my grandmother Sita ma, my twin brother Kailash, and Chaithanya. I cannot thank you all enough for giving me all your support, encouragement and motivation.

I have to thank my committee chair Dr. Ergun Akleman for his valuable input, both technical and artistic comments, suggestions and ideas. Special thanks to my committee members, Dr. Louis Tassinary and Dr. Derya Akleman, who lent their valuable time and expertise, and always had an open door in case I needed some guidance with my research. Thanks to Dr. Tassinary for sharing his wealth of knowledge in the vast field of psychology, and for helping me with the relevant literature. I thank Dr. Derya Akleman for helping me with the huge amount of data during the early stages of my research, and for all her motivation and encouragement.

I met some of my best friends while at A&M; I thank them all for their support, help and motivation, Aparna, Vinod, Pallavi, Sandy Shah, Nitin, Rashmi, Siddarth, Sunny, Sapna & Kiran. Thanks! And thanks for all the great times! Thanks to Sajan Skaria (Pixar Animation Studios) for keeping the passion burning within me, for motivating, encouraging and inspiring me every single day.

Thanks to the Visualization Lab community. Glen Vigus, thanks for keeping your office door open at all times and always eager to help. Special thanks to Eric Androas, Jake Brooks, Radhika, Ellen Trinhe, Kevin Singleton, Piotrek Krawczyk and Josh Rowe for all their timely help. Thanks to Kurt Phillips, for sharing the same

undying love for yet another Freebirds monster burrito. Thank you to all Former Students who have graduated from the lab, who instilled the sense of community and the underlying spirit of the lab.

Thank you to all my online mentors, who have taken the time from their busy schedule to patiently answer all my questions, comments, suggestions, comments. Thanks to the following: Brian Green(Pixar), Jason Schleifer(PDI,/ex Weta), Laurent Kermel(PDI,/ex MILL). Bill Sheffler(Pixar), Scotty Sharp(BlueSky Studios), Robert Moyer(Pixar), Paul Thuriot, Jeff Unay(Weta Digital), Brad Clark(Ex.Weta Digital), John Root(Id Software), Lance Thornton(Pixar), Bryan Ewert. Most of all, thanks to Pixar Animation Studios, for keeping the passion burning within me. Tommy Clark, thank you for giving me an opportunity to help you, to learn from you. You inspired me every single day, and will continue to inspire me forever. Thank You !

And FreeBirds, for making the worlds best ”monster” burrito.

# TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

# LIST OF FIGURES

FIGURE Page

FIGURE Page

# CHAPTER I

## INTRODUCTION

In this thesis, I present a fast, efficient and production friendly method to generate a crowd of fully articulated characters. Using this method a wide variety of characters can be created from a relatively few base models. Although, the models look anatomically and visually different from each other, they maintain and share the same topology. Figure 1 shows an example of crowd characters generated using my method.

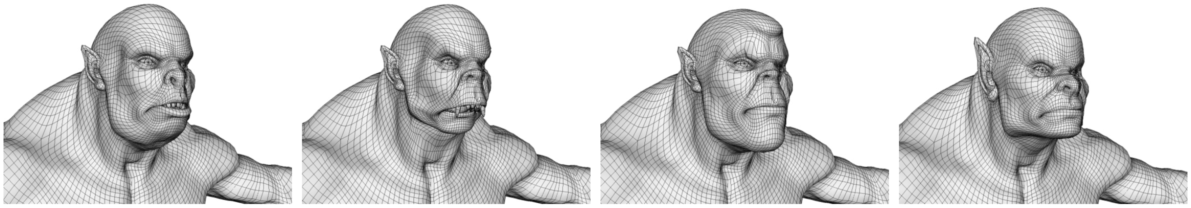

Fig. 1. Some examples of crowd characters generated using the system I developed. Model: Jeff Unay [1, 2, 3]

As you can see in Figure 1, the models that are created by using my method have individual, identifiable characteristics and features. They are therefore distinguishable when they are used in a crowd. This method is very robust and flexible and just not limited to humanoids and can be very easily adaptable to different types of characters, from humans to spiders. The resulting crowd characters generated by this method, are fully articulated and are ready to be animated. They can be animated either by using motion capture data, or from a library of animation cycles, and if required they can also be hand animated.

This thesis follows the style and format of *IEEE Transactions on Visualization and Computer Graphics.*

The method to generate a crowd of characters, as explained in this thesis, is based on the manipulation of the transformations on the skeleton that are primarily used for animation. This method is flexible to meet any kind of production demands and requirements, and can be easily incorporated into any production pipeline, from movies to video games, and can be used with any of the different modeling approaches, polygonal modeling, NURBS or subdivision surfaces.

## A. Motivation for Digital Crowds in Movies

Since the very beginning of cinema, film-makers have always wanted to bring epic mythological stories and legends to the big screen. These movies often recreate the larger-than-life stories of mythical, legendary or sometimes fictional characters, their lives and their times.

One of the main problems with epic movies is that they are very expensive to produce since they require elements such as elaborate sets, props and authentic period costumes. One common element/feature in epic movies is large crowd sequences. Creating a large crowd, be it spectators in *Ben Hur* or *Gladiator*, or huge massive armies fighting each other, like in *Lord of the Rings*, has always been a big challenge to film-makers. One of the first films to depict large crowds was the 1915 silent film, *The Birth of a Nation*, one of the first epic movies ever to be made [4]. Another important movie at about the same time was the 1925 silent film *Ben Hur: A Tale of Christ*. It is reported that about 125,000 extras were cast in some of the crowd sequences, and several Hollywood stars of that time appear as uncredited extras during the chariot race. However, the record for crowds belongs to 1982 movie, *Gandhi*, which had a phenomenal amount of extras - 300,000 appeared in Gandhi's funeral sequence alone.

Nowadays, hiring actors to play as extras is practically impossible because of the prohibitive costs involved. In fact, in the movie *Gandhi* among the 300,000 people more than 200,000 were volunteers and the rest were paid a very small fee. With the recent advances in the state of the art computer graphics (like faster processors, cheaper disk storage, superior computer hardware and high-end graphics software), it has become possible to replace real people in crowd sequences with digital characters. Coupled with the cost of different costumes, props, special makeup and prosthetics needed for each and every individual actor in the crowd, it is fairly obvious that using digital characters can help lower production costs and give the director absolute control over the crowd and its behavior, and almost every single aspect of the crowd, like costume and props etc, but at the same time be very cost effective. It also makes it much easier to recreate or redo the crowd scenes instead of re-shooting them with real actors. In fact, the recent epic movies such as the *Lord of the Rings* trilogy, *Alexander* and *Troy*, which involved huge crowd/battle sequences with thousands of characters fighting each other, was made possible by digital characters.

Digital actors also provide the perfect solution for scenes that might be unsafe for real actors. For instance in *Titanic*, digital characters were employed in scenes involving scores of people falling off the ship into the sea [5].

## B. Aim, Goals and Objectives

Until recently, whenever digital crowds were used in movies, the individual characters that constituted the crowd lacked specific identity, character or individuality. Their only character was that they were a part of the bigger crowd. They were most often far away from the camera, primarily due to the way they were created. Most studios or production houses, employ some kind of mix and match system to generate crowd characters and often used superficial detail to differentiate individual characters. Superficial details often included color of tunics and caps, presence of facial hair, color of hair, etc. For example, in *Shrek* (PDI/Dreamworks), the crowd team started with five basic body types (two men, two women and a child), as well as several types of heads and a variety of outfits that they could mix and match automatically [6, 7]. Figure 2, shows the different aspects of the crowd modeling process, from stand-in geometries to rough model of the crowd, to the crowd characters with hair and clothing and to the final rendered image from the movie *Shrek*.

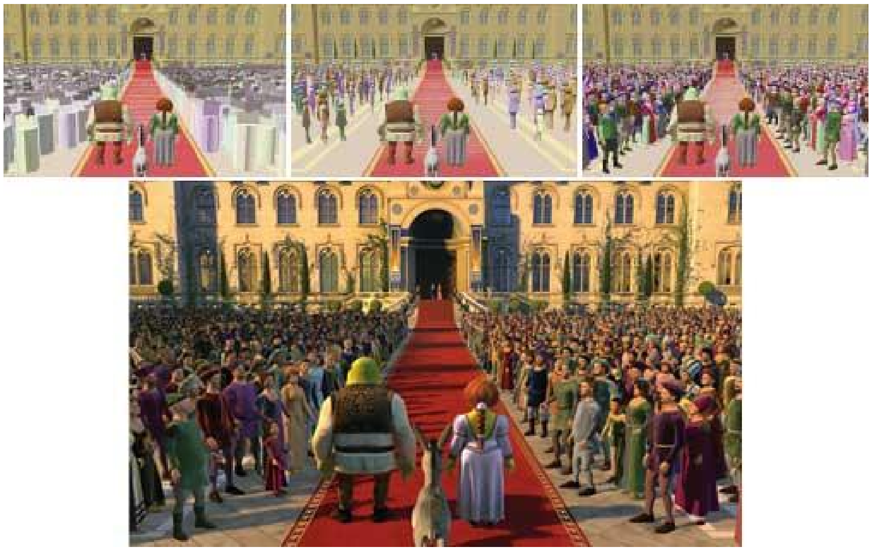

Fig. 2. Crowd development process from *Shrek* [6].

My main motivation is to make the crowd like a group of secondary characters. In this thesis, I have come up with a way to create a crowd of characters that are anatomically different from each other. The superficial details can be added to this varied crowd of characters to enhance the differences to make it a more organic looking crowd. The advantage of creating a detailed and organic crowd is that the camera can now get up close and into the crowd, and since the characters are articulated for animation, the whole crowd becomes more believable. This will allow the camera to interact much closely with the crowd and will be an effective story telling tool.

# CHAPTER II

# PRODUCTION PIPELINE FOR CHARACTER DEVELOPMENT

Highly realistic characters are rapidly becoming common in the movie and the gaming industry, even though their creation is still a challenging problem. The level of detail and the amount of work that goes into these digital characters depends on the characters on-screen presence. Digital characters can be classified as follows:

1. **Hero Characters:**

   These characters are typically the protagonists and the antagonists of a movie, those around whom the story or plot revolves. Shrek and Fiona, from *Shrek* [8] are examples of hero characters. Hero characters have a larger on-screen presence and are close to the camera for a good amount of time. Therefore, articulation of hero characters requires a complex system based on muscles for accurate and realistic skin deformation. This complex articulation for the hero characters, helps the character to deliver dialogues (lip-sync) and the ability to act and emote in a believable way. For instance, to create facial expressions of Dobby (in *Harry Potter and Chamber of Secrets*), Industrial Light and Magic (ILM) sculpted and interpolated 200 facial models that shows different facial expressions [9]. Another recent example is Gollum in *Lord of The Rings.* To create Gollum the special effect company Weta Digital created 675 sculpted expressions, with 9000 sculpted muscle shapes [10].

2. **Secondary Characters:**

   These are characters that appear in the movie for a couple of shots or sequences. These characters help to contribute to the storyline but they are there only for the story-telling. An example of secondary characters are the Royal Chef and the cooks in *Shrek* [8]. Secondary characters do not need to have full-range of expressions as a hero character. However, they still must have the ability to deliver basic acting with simple expressions and dialog delivery. However, they are still expected to be able to deliver dialogue and act up to a certain degree.

3. **Background Characters:**

   Background characters are characters that are predominantly part of the background. They are widely perceived as characters without any significant on-screen presence and without the ability to attract the viewers attention. These characters are typically used in crowds. For example, in the crowd sequences in Shrek, there are possibly hundreds of characters, not one of them matter to the main story directly, but they together as a crowd, assume a bigger importance and significance. They typically enhance the story telling and magnify the scale of the scene or narration. Background characters are typically articulated for animation, but not for acting or lip-sync. They typically would have minimal animation controls, depending on the demands of the production.

To withstand any last minute production changes, digital characters ought to be more flexible and robust, its important to perceive and create each crowd character as if they are secondary character. Even a character that is considered a background character may briefly come closer to the camera than what had been planned or anticipated. Therefore, our digital characters should be able to withstand any last minute changes and modifications (like story, camera/layout changes etc). In fact, some companies like Pixar Animation Studios, treats background characters like secondary characters. They create both, secondary characters and background characters, the same way using their proprietary system called Universal Man [11].

However, the creation of the convincing hero or secondary digital characters is still a challenging problem that faces the entertainment industry. Creating such characters suitable for animation is an extremely time consuming and labor intensive process [10, 9]. In the rest of this chapter, I discuss the state-of-the-art in character modeling, to emphasize how hard it is to model and create digital hero characters. Using modified and deformed versions of these painstakingly created hero characters as secondary characters, I claim that we can save huge amount time and labor.

## A. The Pre-Production Pipeline

Creating a digital character for a movie or video game is a multi-step process (loosely defined as *pipeline*). Almost each and every character in a movie production passes through the pre-production pipeline before it is realized and is ready for shot production. Typically, shot production involves animation, lighting and rendering, effects and finally compositing [12]. The pre-production pipeline can be broadly divided into four stages: (1) *Concept Art and Character Design*, (2) *Character Modeling*, (3) *Texturing/Surfacing and UV Layout*, (4) *Articulation/Character Setup.*

Typically almost every character we see on screen passes through a similar pipeline. Each and every step mentioned in the pipeline is extremely *labor intensive, tedious and time consuming.* Studios like Pixar and PDI/Dreamworks spend up to two to four years on the preproduction stage of the project, which includes, story, character and look development. The pipeline, as mentioned here, is really the heart of a production, the time and effort put into this stage of the production usually saves a lot of problems and glitches later on in the production pipeline.

### B. Concept Art and Character Design

Development of concept art and character design, starts very early on the project time-line and it the most important part of establishing the character for the entire production. This process is kept underwraps as the project is still in its early stages and movie studios guard the details about characters and character designs zealously early on in the project.

Recently, many books that document the art and development stages of a production have been published [13, 14, 15]. These books document the process of how characters evolves through the story development process to their final on-screen form/avatar. This stage is heavily dependent on art direction. Once the character design is approved by the director, art director and creative Leads, it heads to the modeling department.

### C. Modeling

Digital artists generally employ one of three different methods to create character models for movie or video games:

- **Sculpting and Digitizing**: Models are most often sculpted from a traditional medium like clay and then digitized. Clay is the medium of choice for most artists, because clay can be directly molded and carved to create both smooth surfaces and fine detail. However sculpting with clay and then digitizing the clay model ( i.e. the maquette ) has several limitations for digital animation. For example, much detail can be lost in the digitizing process and long term storage of clay maquettes is difficult. This works perfectly well with the main characters (also called hero characters) in a movie production, but the same method is not suited for a crowd of characters, which may comprise of hundreds of models, as creating individual characters that populate a crowd is both labor intensive and costly. Storing all these individual models also poses a challenge to any studio.

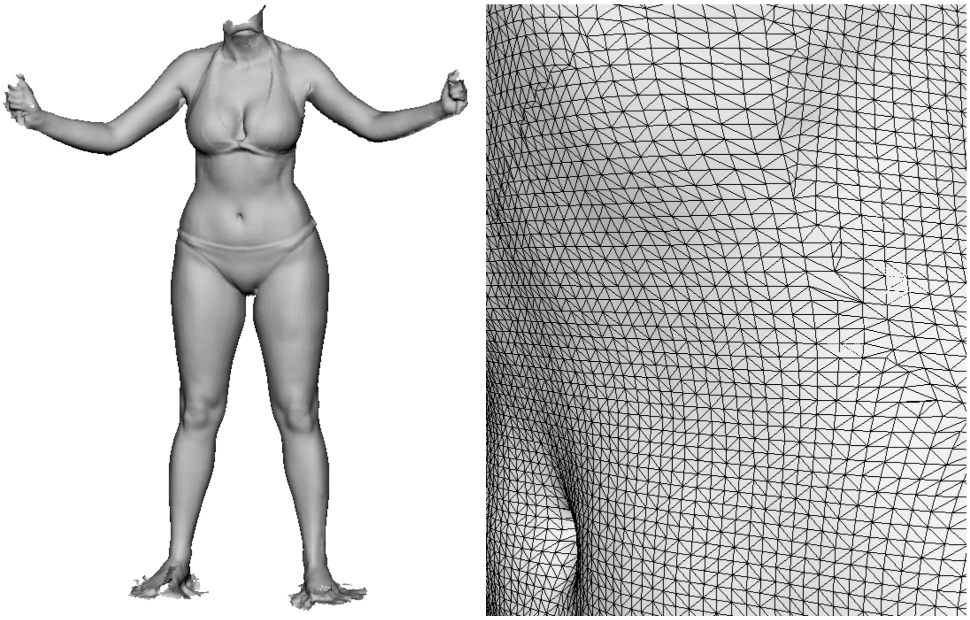

Fig. 3. Typical 3D scans and unstructured mesh topology. This mesh consists of 224,567 triangles which makes it unsuitable for a crowd of characters.

- **3D Range Scans**: Another method which is similar to the above mentioned approach is the use of 3D scans to create models. Recent advances in technology together with research in computer vision and graphics has resulted in systems for capturing surface models of complex objects, people and even interior spaces or environments [16, 17, 18, 19]. These new approaches produce accurate and hyper-realistic 3D models of complete objects with a level of detail not possible with previous manual techniques. Such techniques however, result in object models which are represented as unstructured polygonal meshes consisting of millions of polygons, as shown in Figure 3. Conversions of this noisy incomplete surface model (one which has holes in it), to a structured form, suitable for animation require labor intensive manual remeshing, almost making it economically unrealistic to be employed in a movie production. Moreover, it is rather time consuming to animate them by assigning skeleton appropriate skinning information [20].

For years, the goal has been to develop techniques to convert the scanned data into complete readily articulated models [20, 21, 22, 23, 24]. The most obvious drawback of the 3D scanning method is the resulting high dense mesh, typically in the range of 250,000 - 350,000 triangles per character mesh. Typically, there are holes in the mesh, due to occlusions and grazing angle views, as seen in Figure 4.

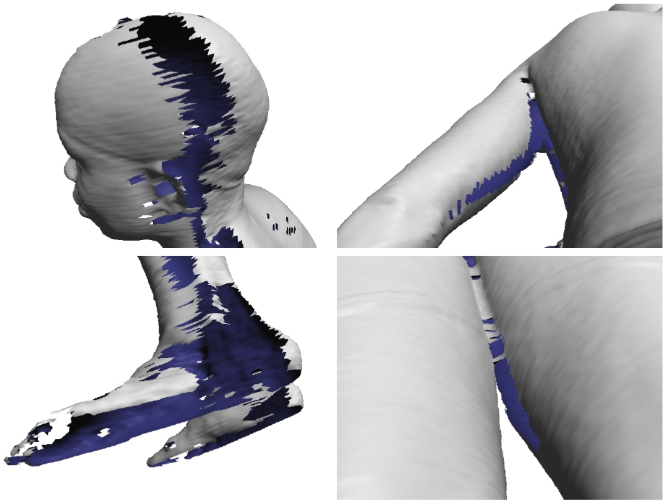

Fig. 4. A common problem with 3D scans, holes. These holes are typically due to occlusion and grazing angle views. [25]

There is considerable post-process work involved in cleaning the mesh for production. Because of the work involved and the resulting high density mesh, virtually eliminates this method from being employed in production, also the mesh generated by this method will practically will not work for a crowd of characters [25].

- **Modeling in 3D**: Animation artists also create (virtual) character models in 3D on the computer. They typically use one of several commercial softwares like Maya, SoftImage, or specialized modeling programs like ZBrush, Mirai, Modo etc. Most movie studios and production houses also have their own proprietary software and tools, which they use to model characters that suit their specific production pipelines and needs.

Commercial modeling systems have polygon, NURBS and subdivision surfaces to represent shape. These three representations are all edited by manipulating control vertices, requiring significant skill and patience as well as foresight and careful planning to ensure that models have enough control vertices where detail is desired.

### D. Texturing/Surfacing and UV Layout

This is an important step in finalizing the look of the character. The texturing and surfacing is what makes 3D models look true to its character, and makes it come to life. For example, Nemo, in *Finding Nemo*, has to look like a fish, with scales and surface properties that makes it look like a believable clown fish. Similarly, all the insects in *A Bugs Life* and *Antz* had to look like stylized versions of real insects to make it believable.

UV's[26] are way to represent how a 2-Dimensional image can be wraped around 3-Dimensional model. The surfacing and texturing artist, usually create 2D texture maps or procedural shaders, that will reveal the material and surface properties (rusted metal etc), other properties like age, color, wear and tear etc. Before texturing, the UV's have to laid out nicely so that the textures do not stretch or cause artifacts, when applied to the object. Utmost care needs to be taken when the UV's are laid out. Overlapping UV's cause weird artifacts and causes textures to stretch. Most often to boost render times, textures are baked from procedural shaders, and ambient occlusion shaders.

Most production houses/studios use high end software like Pixar's *RenderMan* or *Mental Ray*, and some studios like BlueSky Studios have their own proprietary renderer, *CG Studio.*

### E. Articulation/Character Setup

Characters in a 3D-animated movie or feature can be roughly classified, based on their screen presence (as in, their character roles, screen time etc) as *Hero characters, secondary character* and *crowd characters.* Depending on the category, the amount of work and detail that goes into the character varies.

Since the hero characters used in movies, need to act and emote, they need to have a high level of detail like wrinkles and skin folds etc to facilitate subtle movement of skin and facial muscles. This usually means that these models have high resolution with lots of detail (geometry detail like vertices/control vertices (CVs) as the in the case of NURBS ). This is one of the most cumbersome work in 3D computer graphics, pushing and pulling vertices in 3D to create clean, good quality models. To make these models, be able to act and emote, controls need to setup and the process is called articulation or character setup (rigging).

## F. Crowd Pipeline - Reusing Characters Effectively

It is fairly obvious that even creating one digital character is labor intensive and involves a lot of planning, and the work can take many months before the final character is realized and can be handed over to animators. 3D computer graphics, in general, is heavily dependent on preproduction work, unlike 2D animation.

Creation of a digital character, is a process which relies heavily on human input and artistic decisions and can hardly ever be automated. Creating a crowd of a few hundred characters, cannot be done using such a pipeline, as it would not be cost effective and time consuming. We have to slightly modify the pipeline to make sure we can be efficient and productive.

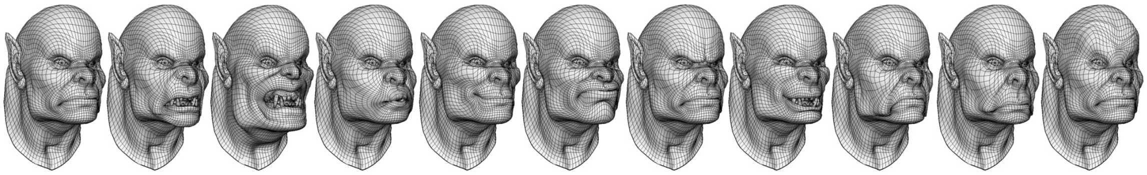

Fig. 5. Crowd facial animation. Efficient use of facial blendshapes now facilitates facial animation for all crowd characters. Model by Jeff Unay and blendShapes by Erick Miller.

In my thesis, my main contribution is to demonstrate a simple and effective crowd pipeline. It is compatible with the pipeline most studios follow. Therefore, it is easy to incorporate into any existing pipeline structure. I present a fast, efficient and production friendly method to generate a crowd of fully articulated characters from a few base characters. The pipeline for our crowd, follows the character pipeline but differs on the fact that, not all characters in our crowd pass through the pipeline. We take a typical character (the base model), develop the base crowd character as it passes the pipeline and use this model to create our crowd. The advantages of this method is, that we save time and studio resources in developing the crowd models. The work put into the base mesh can be reused effectively.

Using this approach, a secondary character that will be used in a crowd can inherit most acting capabilities of hero character that is used as base model. As evident in Figure 5, facial animation maybe incorporated into each character without any additional overhead and effectively reuse work done on the character.

A good quality model will make the task of a character setup Technical Director a lot easier, and create good deformation, which aids in creating believable acting and animation. Good articulation is the key for good animation and for a smooth pipeline work-flow.

# CHAPTER III

# HISTORY

In this thesis, I introduce the idea of effectively using skeletons to sculpt and deform geometry to create new crowd characters. In other words, using skeletons, which have traditionally been used as an animation tool, to sculpt and model geometry.

## A. Skeletons in Computer Graphics

In 3D computer graphics, skeletons have been primarily used for animation. It is well known in the entertainment and gaming industry, that the joint placements are very critical to attain good mesh deformation during animation. The joint placements have to be as close to the true anatomical bone/skeleton structure of the character in question, to get good and realistic deformation during animation.

In CG terminology, a skeleton is usually associated with a corresponding skin (3D model). The model is bound to the skeleton through a process called Binding/Enveloping. The process of binding is very crucial as well as critical one to achieve good deformation during animation.

Special care has to be taken while point-weighting to achieve smooth falloffs. When this skeleton is rotated while being animated, the associated part of the skin (namely the control vertices on the mesh) also rotates based on the weighting to individual skeletons. This rotation of the underlying skeleton creates animation. With advances in computing resources and the increased attention to detail, another technique recently used in movies involved modeling skin sub-structure, like muscle and tendons to drive the skin geometry [27, 28].

The most common and widely used skin computation model for organic characters is variously called smooth skinning, SSD (Skeleton Subspace Deformation), enveloping, liner blend skinning. Catmull first introduced the idea of a skeleton driven technique for skin deformation [29]. In 1976, Burtnyk and Wein, presented a 2D skeletal deformation technique [30]. This method, strictly 2 dimensional, provided the animator to develop a complex motion sequence by animating a stick figure (skeleton structure) representation of an image, , as shown in Figure 6. This control sequence is then used to drive an image sequence through the same movement. In 1988, Magnenat-Thalmann, et al. [31] and Komatsu [32], presented a 3D skeleton driven technique which deformed character meshes.

Early research in computer graphics and animation, developed 2D animation techniques based on traditional animation [33]. Techniques like storyboarding [34], keyframe animation [35, 36], inbetweening [37, 38], scan/paint and multiplane backgrounds [39], attempted to apply the traditional (cel) animation process to the computer.

In the early 1970's, Burtnyk and Wein, successfully demonstrated that keyframe animation techniques constitute a successful approach to animate of free form images [35, 36]. As 3D computer animation research matured. more resources were devoted to image rendering than to animation. Because 3D computer animation uses 3D models instead of 2D drawings, fewer techniques from traditional animation were applied. Early 3D animation systems were purely script based. Catmull introduced the concept of the hierarchy, parenting and a Motion Picture Language (MOP) and presented a script based approach to animation [29]. Early 3D animation systems were script based, followed by a few spline-interpolated keyframe system. Mostly these systems were developed by companies for internal production use, and so very few traditionally trained animators found their way into 3D computer animation. Around the mid 1980's, the first commercial software started to appear in the market from companies as Wavefront Technologies, Alias Research Inc, Abel Images Research, Vertigo Systems Inc, Symbolics Inc and others [40].

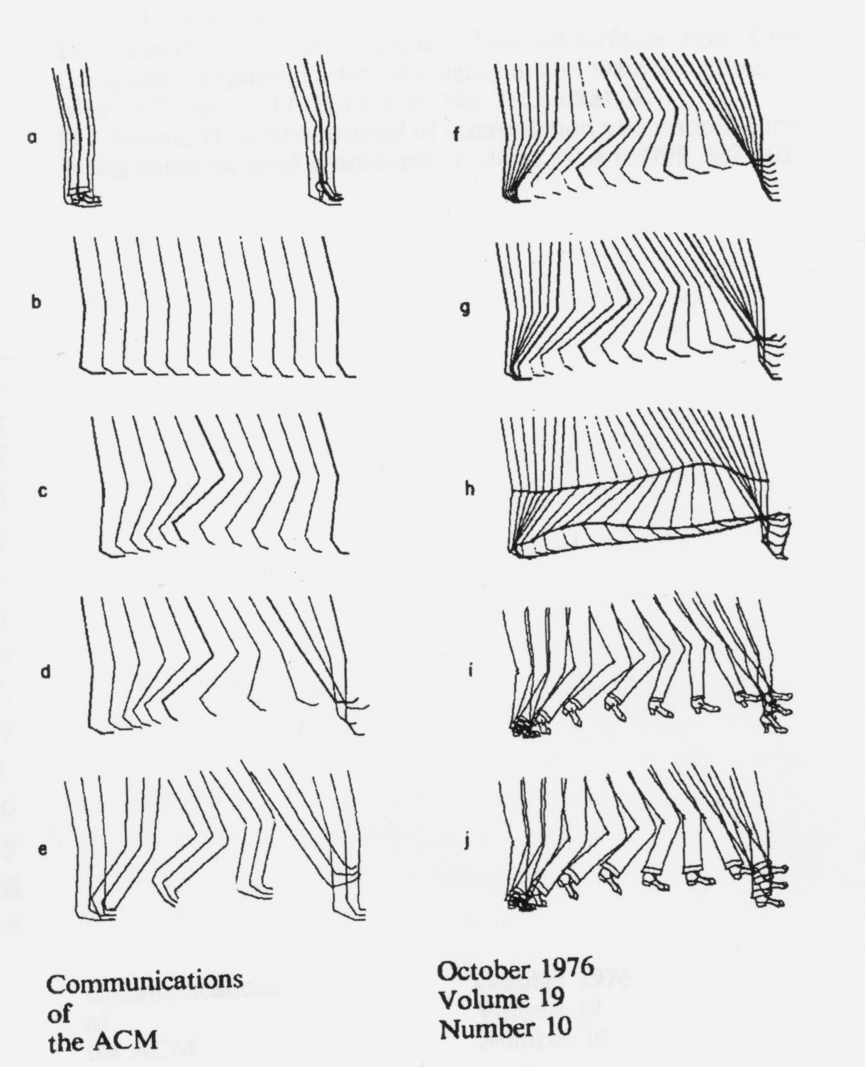

Fig. 6. Primitive concept of a control skeleton to represent 2D animation [30].

It was Badler and Smoliar's research that laid the foundation for the digital representation of human movement. Labanotation, (also known as Kinetography Laban) is one of two ways for recording human movement. Labanotation is based on the general abstraction of the human body. It is here that the idea of joints to represent human movement is introduced [41], as in Figure 7 [42]. The primary function of Labonation is to describe the points and their path through space.

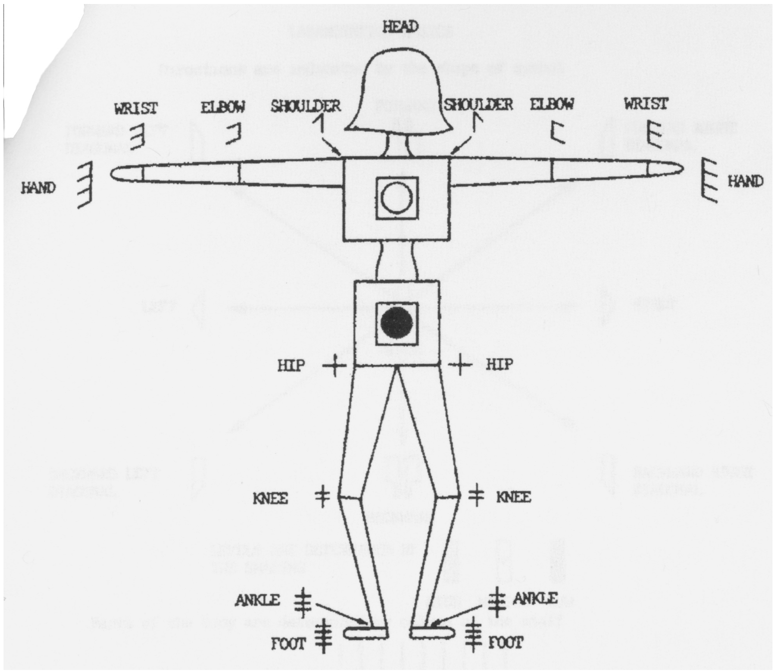

Fig. 7. Abstraction of the human body into joints [42].

Labanotation is based on an abstraction of the structure of the human body like in the figure above. The principle data elements of this abstraction are the individual joints and extremities of the body, with additional articulation of the torso region into "joints". The essential task of Labonotation is to describe the position and trajectories of a set of points in space. The Labanotation views the body as a set of joints connected by limbs. This is probably the closest to our present day skeletons and bones in 3D [42].

Skinning geometry effectively continues to be one of the more challenging and time consuming aspects of character articulation character setup. An anatomically precise geometric model must also move with an equal degree of realism or we do not accept the illusion. Underneath the skin of a human model is the representation of the skeleton. The skeleton is an underlying articulated hierarchy which provides the foundation for controlling the motion of the character. The muscle layer is then added on top of and attached to the skeleton hierarchy.

In recent years, there have been a variety of new skinning methods introduced to model accurately character skin deformations [43, 44, 45, 21, 46]. These remain beyond the scope for a crowd system for obvious reasons of complexity and huge system resources these methods demand. These methods works perfectly fine on hero characters but they are too complex and cumbersome to implement on a crowd of characters.

## B. Crowds in Movies

Since the very beginning of cinema, film-makers have created large crowds scenes in movies, right from the the 1925 silent film, *Ben-Hur* to the more recent films like, *Lord of the Rings* and *Troy.* For over 75 years, their style and techniques have evolved and changed with technology, from employing 300,000 extras for *Gandhi* to having over 100,000 digital characters fighting each other in *Lord of the Rings.*

One of the very first movies to successfully implement and utilize a digital crowd effectively was *Forrest Gump.* In *Forest Gump*, crowds were created using 2D image manipulation, done very effectively. A portion of the intended crowd was shot and then this was "cloned" and copied many times over and adjusted for perspective and image distortion.

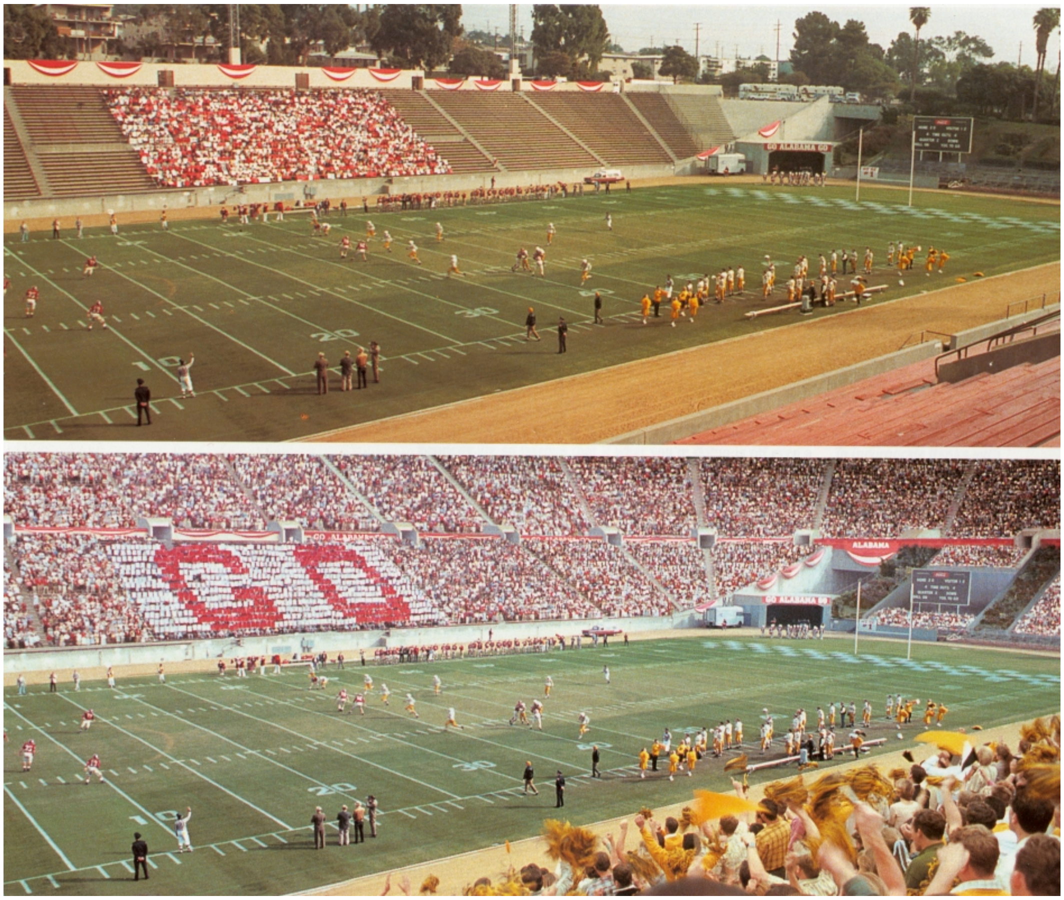

Fig. 8. 2D digital crowd example #1. Effective 2-dimensional image manipulation [47].

In Figure 8, Industrial Light and Magic also expanded the size of the stadium adding multiple tiers, and filled them with digital duplications of the original crowd. As seen in Figure 9 to add believability, extra detail was carefully added, like reflections on the water surface and distortion to adjust for perspective. This helps root the crowd on the background plate.

Another breakthrough film in creating digital crowds is *Titanic*. For many wide shots of the *Titanic*, the passengers are actually computer generated models. The digital extras and stunt-people for *Titanic* were animated via a complex combination of

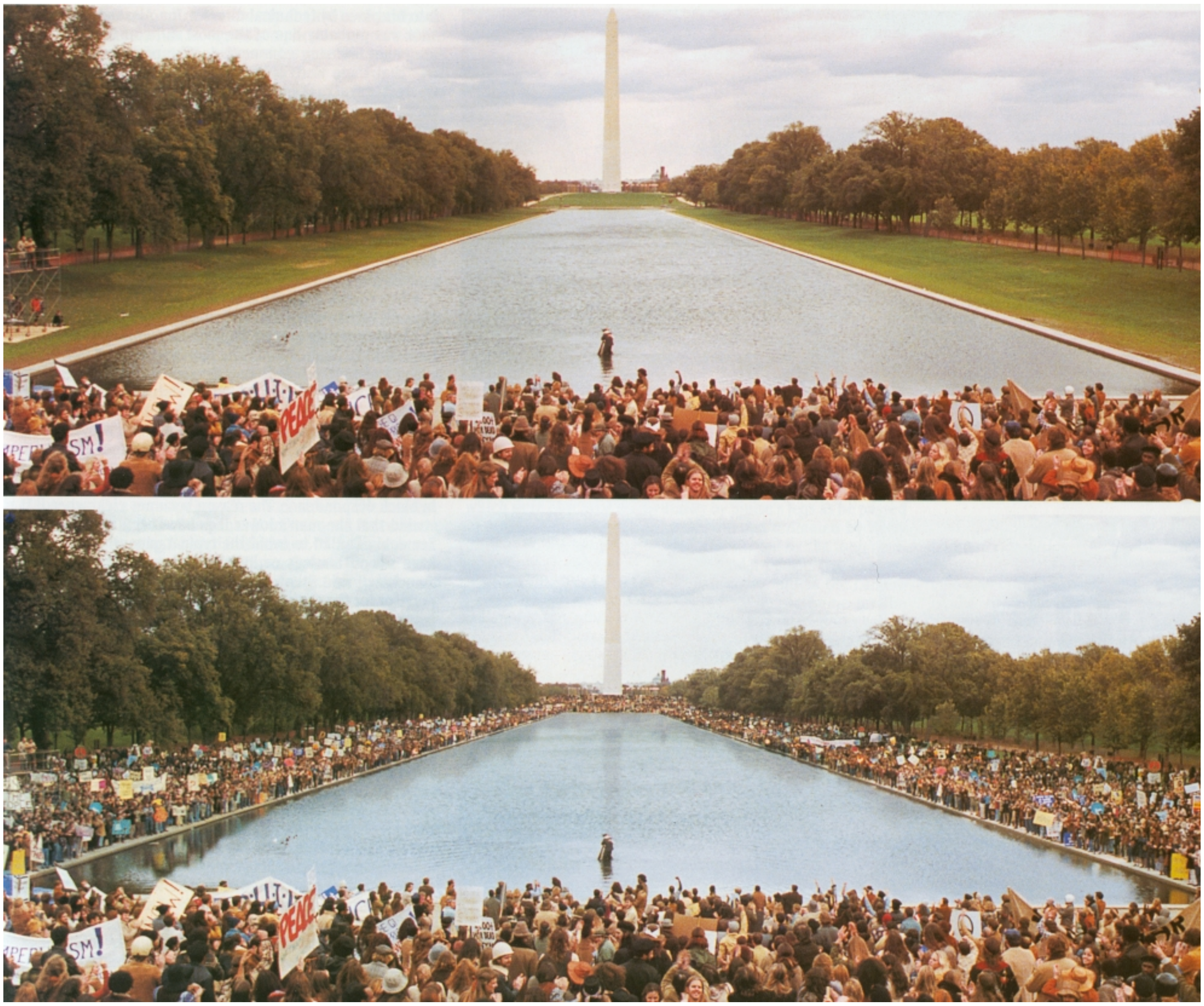

Fig. 9. 2D digital crowd example #2. Effective 2-dimensional image manipulation [47].

motion capture, freehand animation and 'roto-capture', where animators key-framed CG models using footage of actors performing an action as reference. The position of the decks had to be meticulously tracked in 3D and manipulated in 2D, so that CG passengers would actually appear to be standing on the decks and leaning on the railings throughout complicated camera moves.

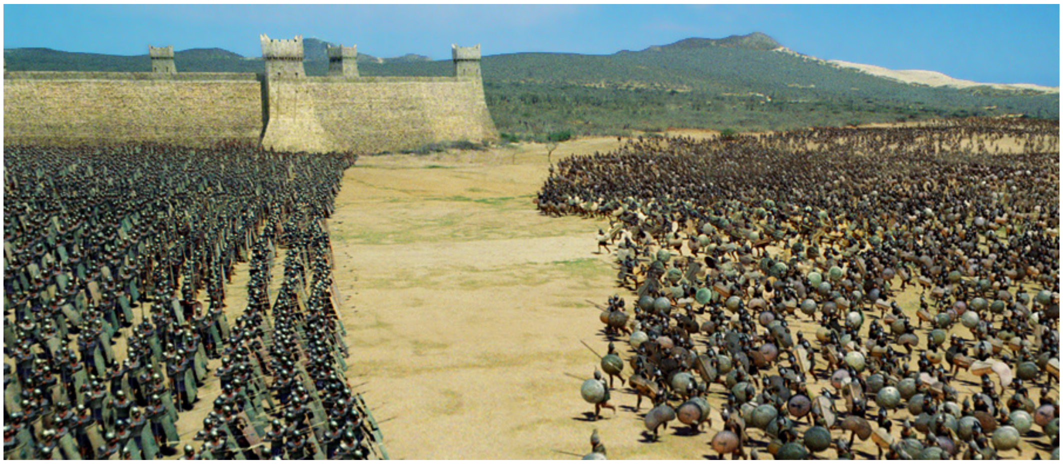

Fig. 10. Digital crowds, up to 150,000 troops fighting each other in *Troy* [48].

Crowds in movies are not limited to humanoids. For instance, in *Harry Potter*, Mill Films, London, created an army of spiders for *The Chamber of Secrets*. Almost all contemporary movies use computer generated crowds. The recent epic movies that use computer generated crowd include *Star Wars I, II* and *III*; *Lord of the Rings* trilogy, *Alexander* and *Troy* (as in Fig 10). While the creation of a big crowd sequences is not limited to epic, fantasy or live action movies, other computer generated movies like *Antz*, *Shrek*, *A Bugs Life*, have sequences that have hundreds of digital characters that interact with each other.

# CHAPTER IV

# METHODOLOGY

In 3D computer graphics, skeletons have been primarily used for animation. The skin (or 3D model) is bound to the skeleton through the process called smooth skinning or enveloping. It is the rotation of the skeletons that causes the skin to deform, and the interpolation of the rotations, on the skeleton that produces animation.

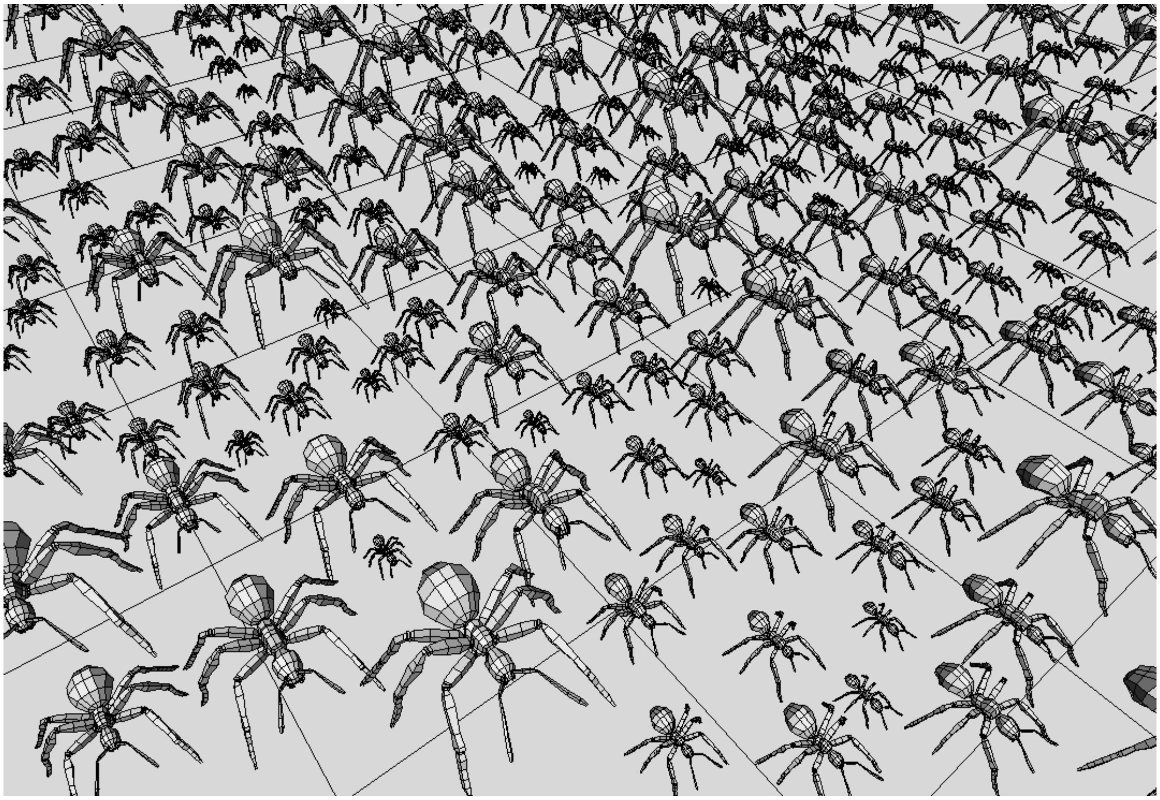

Fig. 11. Ants. A colony of ants created using my thesis. Model by Lu Liu.

Since the skin is already bound to the skeleton, it is very obvious that any other transformation (like translation, scale and shear) on this skeleton will directly affect the skin in an intuitive manner. By manipulating the transformations on this skeleton, we can create a wide variety of characters, that look anatomically different from one another in the crowd. This method is capable of generating a crowd with varying level of detail, from a bare bone crowd (a low-detail digital crowd, with limited animation capability) to a more complex crowd in which, each and every character is articulated to be able to produce realistic animation and has complex facial animation. This thesis explores a way to create and model a 3D digital crowd of characters using traditional articulation tools.

The main advantage of this methodology lies in the fact that the topology is maintained between the original base character and the newly created digital crowd characters. This means all the production work done on the base character model, can be effectively and efficiently reused. This includes, all the point-weighting of the generic character during articulation, the creation of blendshapes used for facial animation, the UV-mapping for surfacing the character, and any procedural or hand-painted textures maps etc, are all retained and can be reused on the newly created crowd character. This approach saves a lot of time, resources, production man-hours, is very economical and production friendly. This anatomical approach helps us create an organic and realistic looking crowd, where individual members are all anatomically unique and different from one another. The new characters that are created, inherit the exact same naming conventions from the base character, with the only exception of having a prefix that uniquely identifies them, from the other characters in the crowd (and also eliminates namespace conflicts in Maya.). This makes task of animating individual crowd characters, using either Motion Capture Data, procedural animation or animation cycles very easy. Figure 11 shows a simple, articulated colony of ants, created by the method explained in this section.

Modeling a large crowd of digital characters is an extremely labor intensive process and because of strict budget and time constraints. Articulating each and every individual digital character in this crowd for animation is a daunting and practically impossible task in a production environment like feature films or video games. This thesis proposes a simple, effective and practical way to generate a crowd of articulated characters. Creation of this crowd of digital characters would include procedural modeling of individual characters and articulating them so each and every character can be individually animated.

## A. Introduction

The generation of a fully articulated digital crowd being an important problem facing the 3D entertainment industry, has to address key issues that will help this approach, or a modified version thereof, to be adopted within a production environment. Some of the main issues are:

- Easily adapted to any generic character, from humans to insects to animals.
- The relative ease of generation of a male and female characters in the crowd.
- Fully flexible and automated approach (preferably independent of the production pipeline).
- Capable of editing, modifying or deletion of any number of specific individual characters in the crowd.
- Anatomically correct and consistent solution.
- Easily adapted to generate child characters.
- Articulated and capable of applying motion-capture data. Unique naming convention to avoid possible namespace conflicts.

The methodology of generation a digital crowd of characters, explained in this chapter is robust in handling many types of characters, from insects like spiders and ants, to humanoid characters to inanimate characters like droids and robots. It can easily be adapted with any of the different modeling approaches (Polygonal Modeling, NURBS, Subdivision Surfaces).

The crowd modeling process, starts with the character development of a generic character. This multi-step creation process (loosely defined as crowd pipeline), starts with the (1) Concept Art and Character Design (2) Character Modeling (3) Texturing/Surfacing and UV Layout (4) Articulation of the generic Character (5) Crowd Creation and (6) Articulating the digital Crowd.

## B. Crowd Modeling: Ideas, Concepts and Inspiration

It is well known in the entertainment industry, that the joint[1] placements are very critical to attain good mesh deformation. To achieve good realistic deformation, the joint placement needs to be as close to the true anatomical bone/skeletal structure of the character in question.

Careful observation of the human anatomy was the main inspiration that forms the basic foundation of this thesis. The fundamental anatomical differences that were observed in humans and later in animals, can be very easily adapted and modified to generate a very large digital crowd of characters. This approach is highly efficient, production friendly and works well within the time and budget constraints in a production environment. This methodology is capable of producing a wide variety of characters, that look completely apart from one another in the crowd. More importantly, this particular approach is anatomically correct and produces consistent and

[1] Joints are also called bones or skeletons in different software, and I use these terms interchangeably.

realistic results.

Consider, for example the varying arm-lengths of, a pro-basketball player and an average person. We expect the athlete to have longer arms than the average person. Anatomically, the athlete's longer arms are due to a longer underlying bone structure that support the extra muscle mass, connective tissues and skin.

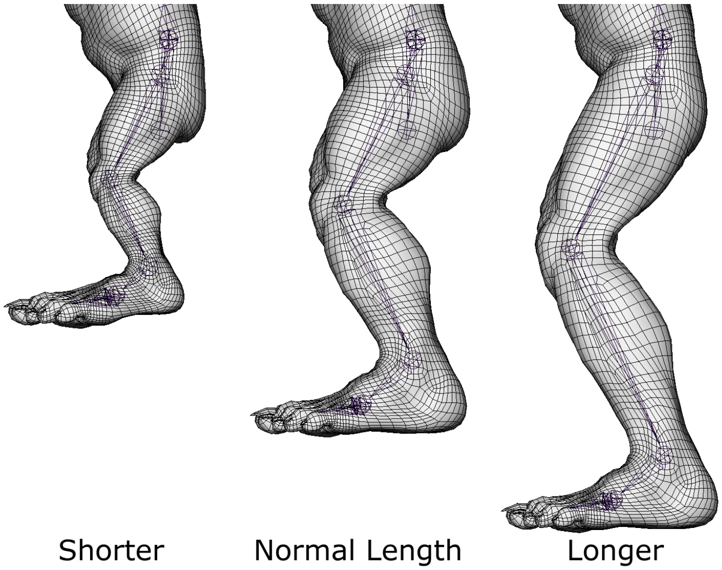

Fig. 12. Digital anatomy. A normal leg of a digital character is shown in the middle and by scaling the skeleton it is bound to, we can model a shorter leg (on the left) or model a longer leg (on the right).

Similarly, the thigh bones of a person weighing 350 lbs are more dense and thicker (in cross sectional area) when compared to a person of the same height who weighs 110 lbs. This increase in cross sectional area and in the physical property like density, can be attributed to the ability of the thigh bone to carry the additional weight, 240 lbs as in this case. The direct relationship between the bone structure (attributes like length, cross-sectional area) and the physical property (like density) to the anatomy and physical structure of the character, are important ideas that forms the fundamentals of this thesis.

The direct relationship between the bone structure and its material property, to the anatomy and physical structure of the character is fairly obvious from this example. The same principle can be extended in the creation of an organic digital crowd of characters. This is achieved by scaling the joint, to which the digital character is bound or skinned to.

To lengthen an arm or leg, just scale the specific joint by a factor greater than one (like 1.24), and to shorten to the same appendage, scale by a factor lesser than one (like 0.85), as illustrated by Figure 12. There is no obvious and intuitive way to model and manipulate physical material property like density of bone tissue in commercial 3D softwares. It is physically accurate to increase the cross-sectional area of the bone, keeping the same bone density. This would result in manipulating the thickness/girth of an arm or leg. To increase the girth of an arm or leg, simply scale the joint in the two axes perpendicular to the length, like in Figure 13.

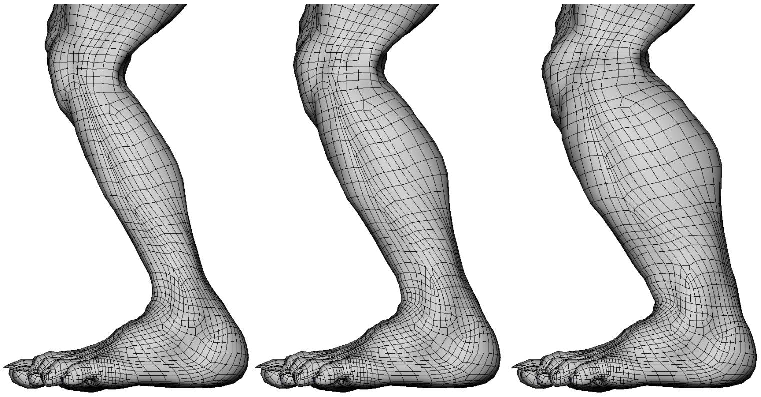

Fig. 13. Digital anatomy. A normal leg of a digital character is shown in the middle and by scaling the skeleton it is bound to, we can model a thinner leg (on the left) or model a bulkier leg (on the right).

## C. Drawbacks and Limitations

This method of crowd modeling can apply to a wide variety of characters, for instance Figure 11, shows a successful example of a huge colony of ants, created using the method mentioned in this section. However, this approach to creation of a crowd, works well within certain limitations.

When the same approach is adopted to a different type of character, for example for a humanoid character, it exposes certain limitations of this method. What we can create is a wide variety of digital humanoid characters, but their facial features all look alike. This method still works well overall, especially in areas like torso, arms, legs and fingers, but is found to be lacking in other areas. We cannot, for example manipulate the shape and size of the nose, chin, ears or chin of the character. These areas are not individually articulated, and it is beyond any reasonable scope to articulate the ears or nose of a crowd character. This is an important problem since we identify, differentiate and recognize human or humanoid characters from one another, based on their facial features. The problem is that there are very few joints for these non-articulated features mentioned above, and by scaling or translating these joints do not produce a wide range of facial features in the crowd.

In other words, we encounter this particular limitation when we want to modify/sculpt a certain part or detail on a character that cannot be animated. For example, the hand and legs of any character can be animated, and our technique works fine if we wanted to make the arms longer or shorter or thinner or thicker/fatter. On the other hand, the nose, finger nails or ears of a character, or a tusk of an elephant, typically do not have to be animated (in a crowd character at least), and hence using our method, it is difficult to change and modify their shape, size and appearance.

## D. Animation Skeleton and Modeling Skeleton

The obvious limitation of manipulating the skeleton to which the digital character model is bound to, to create new crowd characters can be addressed by introducing the concept of the **Modeling Skeleton**.

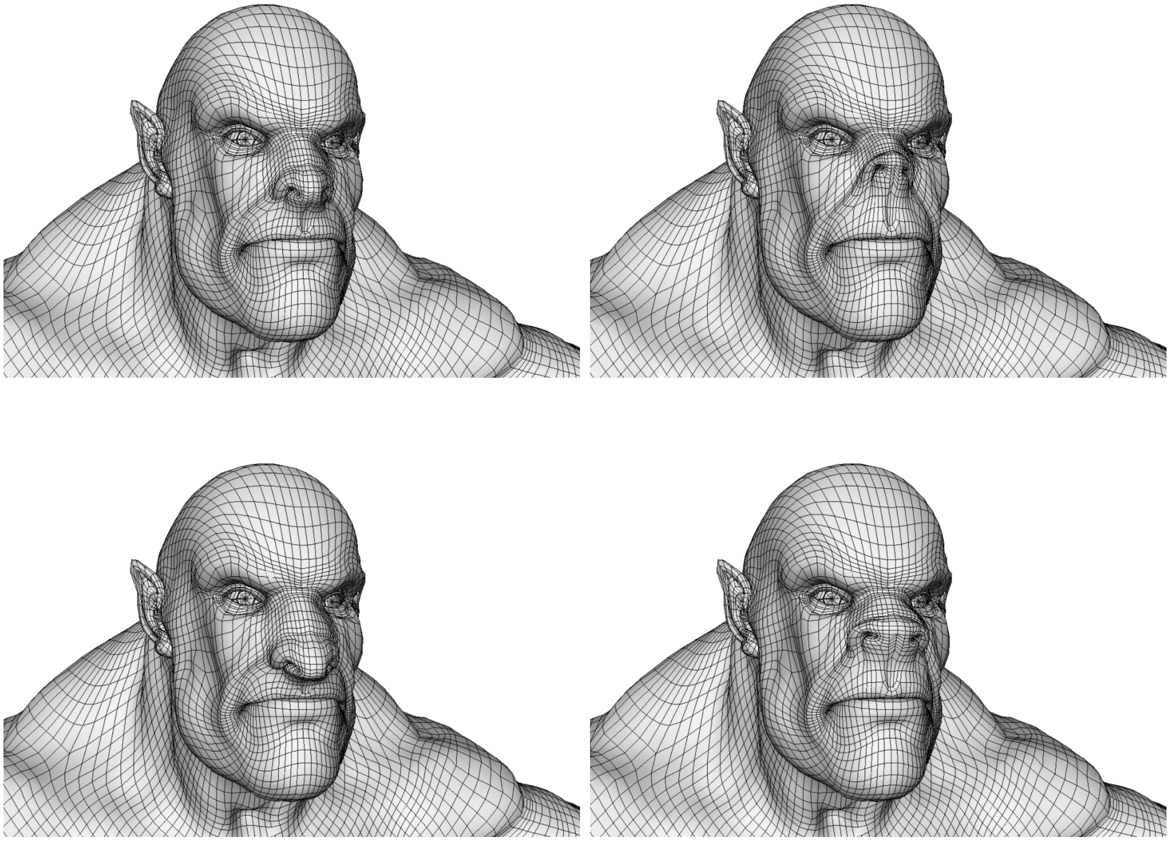

Fig. 14. Modeling skeleton: Using a modeling skeleton to sculpt localized detail for the crowd characters. As shown in this example, the shape, size and position of the nose can be manipulated by using a modeling skeleton.

To clearly identify and differentiate between the skeletons, let us classify them into **Animation Skeleton** and the **Modeling Skeleton** as explained below.

1. **Animation Skeleton**: A digital character needs to be articulated to facilitate animation. This is the basic skeleton the articulated character is bound to. We

shall call this skeleton the *Animation Skeleton*. The rotation of this skeleton produces the digital model to deform and produces animation.

2. **Modeling Skeleton**: To give us absolute control over a character's mesh, as in this example, say the nose of a character, we create additional secondary skeleton, articulate and point weight it locally and then manipulate to create a variety of nose shapes and sizes. Let us call this skeleton as the *Modeling Skeleton*, as its primarily used to tweak the digital model rather than being used for animating the digital character.

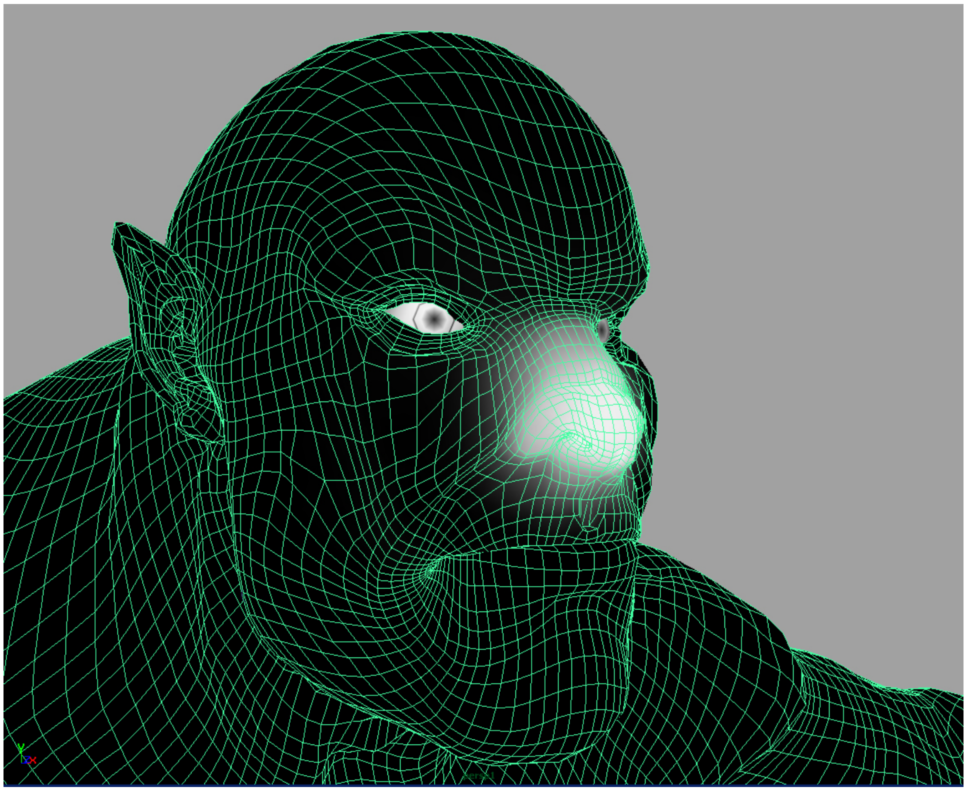

Fig. 15. Point weighting of the modeling skeleton, that is used to manipulate the size, shape and structure of the nose, as seen in Fig 14.

To build a modeling skeleton, we create a new set of skeletons that is completely different than animation skeleton. For instance on the character's face, we create the modeling skeleton, to give us the ability to manipulate and sculpt, localized features like the nose or ear.

Care should be taken to create the modeling skeleton, as the placement and position of this skeleton would depend on the specific characteristics we want to manipulate and exaggerate like length and width of the nose or finger nail etc. We then carefully articulate this skeleton, and point weight it with smooth uniform falloffs, to guarantee good and realistic results.

There are no restrictions about the number of joints, that can be created to sculpt the face or any other specific body part (like finger nails etc). The modeling skeleton guarantees good deformation if the skeleton is properly articulated and point weighted with smooth falloffs.

The modeling skeleton is not used for animation in anyway or manner. It is strictly used as a modeling tool, and acts like a deformer, sculpting, deforming and modifying geometry. It is used for modeling local details so we can change certain features on characters, as the demands of the production dictates. Using the modeling skeleton, we can change a very specific local detail on the mesh (ie character model) without affecting other areas of the animation skeleton, but creating a wide range of variety. This skeleton can be procedurally modified, using scripts and the variety of character models can be created automatically.

### 1. Conflict between Modeling Skeleton and Animation Skeleton

When a modeling skeleton is added on to a carefully articulated model, we expect to see a redistribution of the point weighting of the animation skeleton. The point weighting of the modeling skeleton directly affects the point weighting of the animation skeleton. This redistribution of point weights on the animation skeleton is highly undesirable as it would cause problems further down the pipeline during animation. It is very important that the modeling skeleton is tightly integrated into the crowd pipeline, so both the modeling and animation skeleton can be used together to generate digital crowd characters.

### 2. Adding Specific Localized Detail Back into Our Crowd Character

To overcome the particular problem of redistribution of weights when the modeling skeleton is added to the animation skeleton, we adopt a two step layered approach as explained in this section.

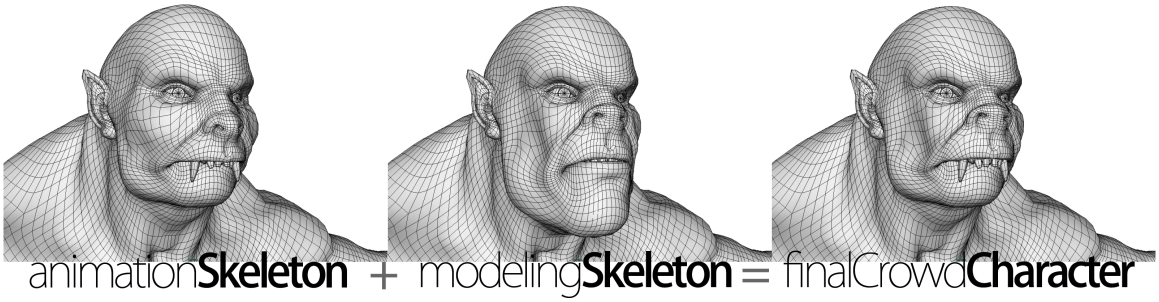

Fig. 16. Using the animation skeleton and modeling skeleton to create a final crowd character. The animation skeleton produces major changes on the character and the modeling skeleton produces localized area detail and both of them together create the final crowd character.

Instead of adding the modeling skeleton to the articulated base mesh, we start by creating a modeling skeleton to a duplicate copy of the character mesh (without the animation skeleton) and articulating it very carefully, so the manipulation of this modeling skeleton creates a wide variety of models. This method works well to eliminate the conflict between the modeling and animation skeletons. We then use this new model, and add it as a new morph target (Blendshapes in Maya) to the character model we initially started with.

This method works seamlessly and is best demonstrated in Figure 16. By manipulating just the animation skeleton, we are quite limited by variety of characters generated. The introduction of the modeling skeleton to sculpt additional local detail, dramatically increases the number of unique characters we are able to create as evident from Figure 14, and Figure 15, shows the sample point weighting for this specific example. The introduction of the modeling skeleton effectively increases our ability to manipulate the almost any part of the mesh of our crowd character. This allows the smooth interaction between the animation skeleton and the modeling skeleton and integrates back into our crowd generation pipeline.

### 3. Facial Animation for a Crowd Character

One of our main goals of this thesis was to make the crowd character a secondary character, and not an obscure background character. This would usually require the crowd characters to have some degree of acting capability and dialogue delivery (lip-sync). It would be beyond the scope of any production environment for a crowd to have complex facial animation based on a muscle system or a system based on hundreds of morph targets.

We have created a very organic crowd of digital characters, where each character in the crowd is individually articulated. It is very easy to modify the crowd pipeline so that each and every character in this crowd has facial animation.

The method explained in this section, will effectively transfer all the facial animation done on the base generic character and apply it seamlessly to each and every crowd character we generate automatically. This system gives the flexibility of making a simple or complex facial setup or a hybrid, where the characters closer to the camera, have a slightly complex facial animation than the characters further away from the camera. Figure 17 and 18 shows a detailed example of morph targets transferred

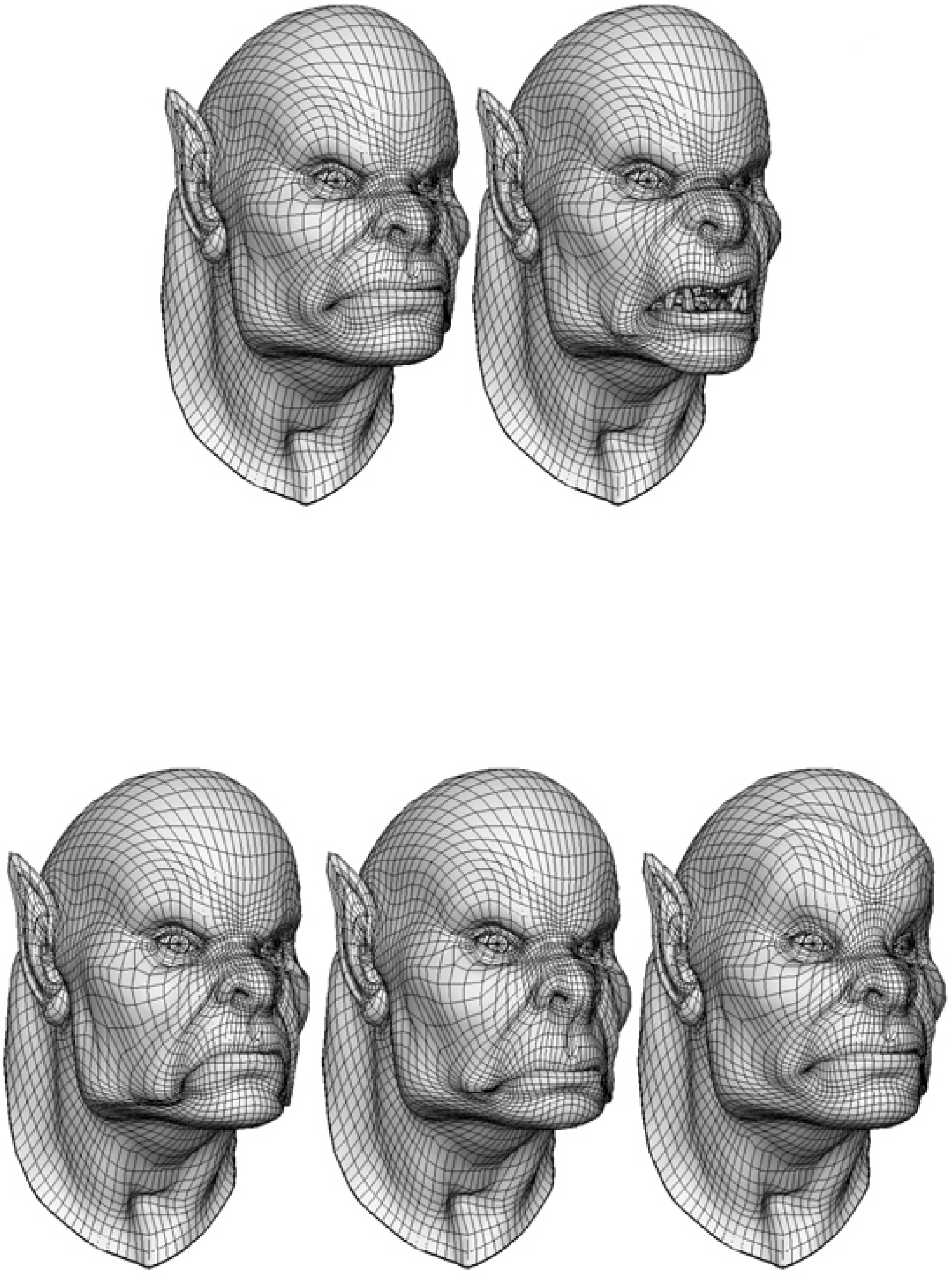

Fig. 17. Facial animation setup for a character. Example of morph targets for facial animation of a crowd character.

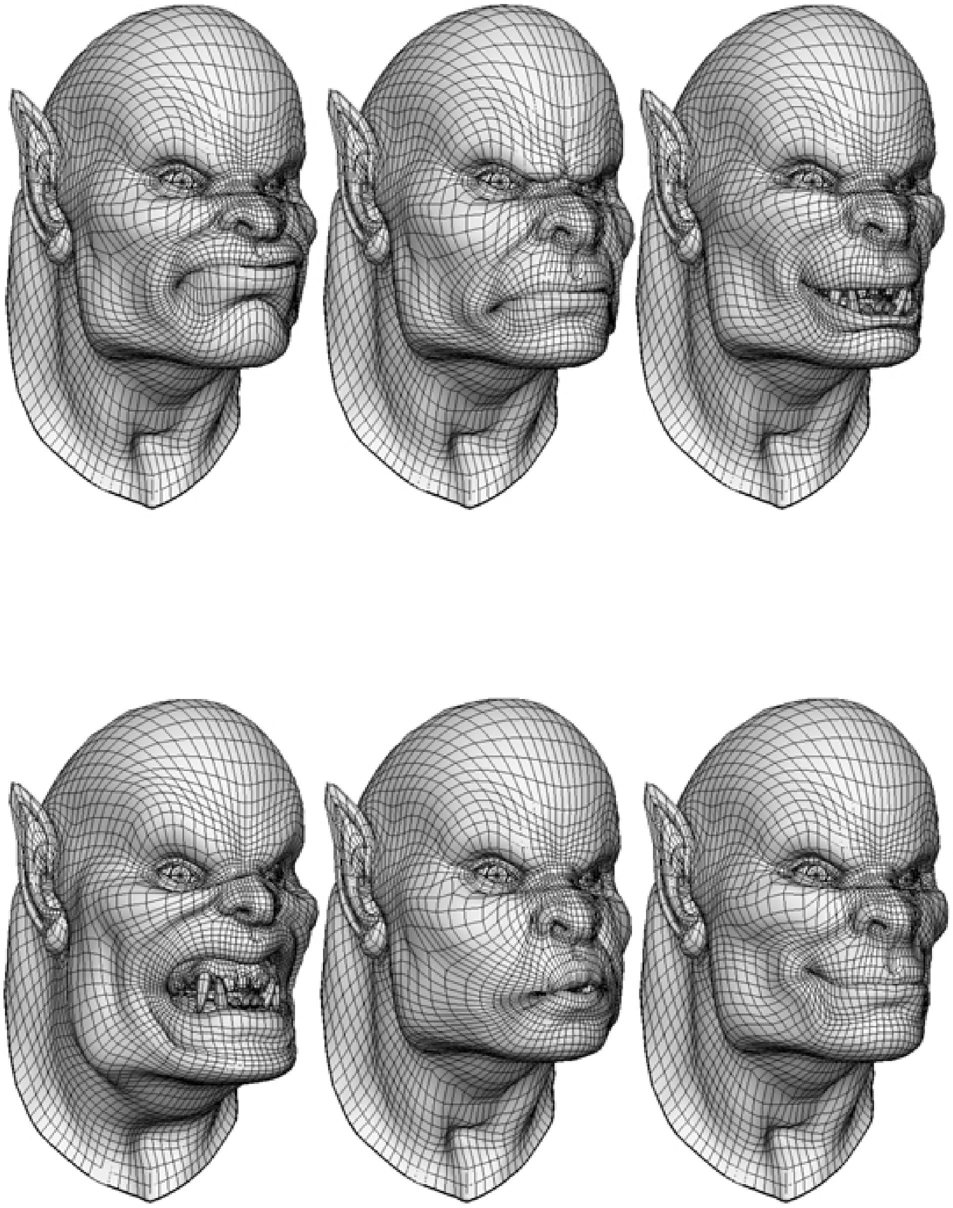

Fig. 18. Facial animation setup for crowd character. An example showing the morph targets set up for facial animation. These blendshapes were actually modeled for the generic character or base character but been transferred onto the crowd character to facilitate facial animation. This saves a lot of production time and each and every character in the crowd has facial animation.

to a crowd character to help in facial animation.

Using articulation tools for modeling is a much effective tool for modeling, than the traditional method of pushing and pulling vertices. The skeleton acts like a deformer used for modeling. The modeling skeleton method explained above can be further broken down to include a series of modeling skeletons. This is best explained through an example. We could possibly have multiple modeling skeletons to model and modify individual localized detail. For example, one modeling skeleton can be used to manipulate nose while another could manipulate the ears. Both these modeling skeletons have to be carefully articulated, and are used to modify and manipulate the size shape and position of the nose and ear respectively. This helps us to create a diverse variety of characters making our crowd feel real and organic. This process of generating individual nose or ear shapes can be automated and randomized and can be mixed and match. A simple system can be implemented where the random numbers are generated within a boundary of extremes to guarantee acceptable solutions.

# CHAPTER V

# IMPLEMENTATION AND RESULTS

To implement the idea of crowd modeling described above, I had to develop tools needed to procedurally create character models, articulate and animate them. I had to first decide on a software package that would satisfy certain basic requirements as listed below. It had to meet the industry standards and an-off-the-shelf, highend 3D software. The software had to be user-friendly, customizable and script-based, to help improve workflow. Upon careful consideration, I eventually chose Alias's Maya, over other commerically available software for its functionality and my familiarity with the software. Maya is completely scriptable with MEL ( Maya's embedded Language ) and allows us to write scripts to automate the process and streamline the work-flow.

To successfully present the idea presented in this thesis, I had to test this approach on a variety of different characters, from insects like ants and bees to humans and humanoids. The results had to demonstrate the versatility of the anatomical approch to crowd modeling, and how easliy it can be adapted to suit the production demands. The ease with which a bare bone crowd and a digital crowd with full facial animation, are created will demonstrate the scalability of this anatomical approach.

## A. Getting Started

- **Assets**: Before I started to implement the ideas and concepts presented in this thesis, I had to decide on the type of characters I would use to demonstrate the crowd modeling results. The characters models had to be of professional quality, as the topology of the model often dictates how the mesh deforms. I decided to use the beast model from the Maya Masters Classes by Jeff Unay [1], Paul Thuriot [2] and Erick Miller [3]. The beast was modeled by Jeff Unay from Weta Digital, New Zealand (from the studio that brought *The Lord of The Rings Trilogy* to life).

I also tested the ideas and concepts of crowd modeling on a generic ant and bee model, modeled by Lu Liu and Radhika Thirunarayanan.

- **Software**: Maya 6.5 for Linux.
- **Custom Scripts**: Over the course of this thesis, I have written and developed a lot of custom MEL scripts to automate the process of crowd modeling.
- **Operating system**: Linux. Mac OS X, Win XP.
- **Graphic Softwares**: Adobe PhotoShop CS, Gimp 2.2.

## B. Generic or Base Character

To create a crowd of digital characters, we start with the generic base character model. The entire quality of the crowd is dependent on this base mesh. Care should be taken with the topology of the mesh. Unay's beast model was an obvious choice to start testing with the crowd modeling idea as it was created by an industry professional. I worked with the beast model extensively to do my numerous little tests and experiments, proof of concept of ideas and to implement solutions. It was a good clean model with proper edgeflow and I was pleasantly surprised how nicely it withstood a wide variety of tests and experiments. It had a good topology, had adequate detail to guarantee good deformation and was a production quality model with good detail and adequate resolution.

To start the crowd modeling process, the base character needs to be carefully articulated for animation. This is an important step and extreme care should be taken to articulate the generic base character. The automatic procedural articulation of the entire crowd of digital characters depends on the articulation of the generic base character. The skeleton of the base generic character needs to have proper and consistent naming convention, as it would help in the procedural animation of the crowd later in the process and it is also a good practice to name the joints in a logical fashion. This is particularly helpful when writing scripts to model the geometry procedurally and also while using procedural animation or for writing scripts for transferring Motion Capture data to the individual crowd characters. The system I have developed uses the same name for the skeletons and joints, but it is padded with a prefix to make the character unique and identifiable.

Now I shall briefly explain the the implementation of this thesis, and for a better understanding about its inner workings and specifics. I shall try to keep this section from being too software (Maya) specific, but again since I implemented it in Alias's Maya 6.5, I am bound to use some technical Maya specific terminology.

## C. Crowd Generation

In this thesis I propose, a fast easy and efficient a way to generate a fully articulated digital crowd of characters. It is a fairly simple and straightforward process, and depends on the specific production needs and requirements. It also depends on the type of character in question (human/humaniod or insects like ants/spiders etc).

Animation is the result of a series of transformations (typically translation and rotation) on the articulation skeleton of any 3D character. Other transformations (like scaling and shear) of this skeleton causes favorable and intuitive deformation on the skin/3D model and this property can be used to effectively used as a modeling tool. If the mesh is carefully point weighted, this tool has the added advantage of producing clean and smooth falloffs. This is the fundamental idea of my thesis.

### 1. Animation Skeleton and Modeling Skeleton

The concept of an animation skeleton and modeling skeleton was introduced in the previous chapter. In this modeling approach, both the animation and the modeling skeleton are used for modeling and deforming the character mesh. The main difference between the two is that, the animation skeleton, is used to deform major areas on the mesh like the arm or leg of a character, while the modeling skeleton, is used to model to create additional localized detail.

The first, and possibly the most important step in crowd modeling is to have a very good understanding of the anatomy of the character in question and identifying ways to create a wide variety of characters (like dominating features and landmarks, like the size, shape, length of the horn of an Unicorn). In some characters, manipulation of the animation skeleton alone would provide a wide range of characters. These characters are mostly simple and most common examples are ants, spiders, bees etc. These are simple characters that do not have recognizable features to distinguish one character from another, but their main character is the crowd itself, of which they are a part, like in Figure 11.

The main features or characteristics, that we would typically like to manipulate to create a crowd of characters, are features like the length and girth of individual appendages, body/torso, abdomen area, head, antennae etc, and if the character is human/humanoid, then have different facial features that makes them unique and identifiable.

The length of individual appendages can be modified if the corresponding Animation Skeleton is scaled along its axis and the girth of the specific appendage can be modified by scaling on the other two axis perpendicular to the axis that runs along the skeleton. This process can be automated with the relative ease, as if the extremes are known, we can generate the length automatically through a random function in MEL.

In some characters, just manipulating this animation skeleton does not create a wide range of characters. For example, in Unay's beast character, just manipulating the animation skeleton provides us with a wide range of realistic body types, but when it comes to the face, since there are not too many joints (for animation) to manipulate in the face, every character almost looks the same. Since we do not want to model a clone army, I introduce the idea of a *modeling skeleton*, as explained in detail in the previous chapter. The modeling skeleton, give us the ability to control and manipulate each and individual (local) feature of the character. The modeling skeleton needs to be articulated carefully to give us intuitive results.

In Maya, a character mesh, cannot have more than one skin-Cluster node, meaning it cannot be bound to more than one skeleton chain, this is a small limitation of using the animation and modeling skeleton in the same mesh. If we create the modeling skeleton as an influence, it reorders the point weights on the animation skeleton, and it causes weird results during animation. We resolve this by using a two step process is using the modeling skeleton and the animation skeleton in parallel. Instead of creating the modeling skeleton on the same mesh as the animation skeleton, we create it on a duplicate copy of the mesh and then point-weight the modeling skeleton carefully, as per the needs and demands of the schedule. Then we tweak the modeling skeleton to fine tune details like size and shape of the nose, ears , finger nails etc. Then we add this as a blendshape to the original character mesh and we now see the we have successfully transfered local modeling detail to our main character mesh. Most often, *the deformation order* needs to be changed to see the result on the original mesh. *The blendshape node has to be above the skinCluster node in the deformation order*, for it to work.

We can now manipulate the animation skeleton to create new and different body shapes (thin, fat, tall, skinny, thin legs, arms etc) and we can also tweak the modeling skeleton to finesse the local detail. Once the crowd character has been finalized, this deformed mesh is like a template and we can bake out final crowd character,

In Maya, it is typical that the animation skeletons have uniform scales of (1,1,1) on X,Y and Z axis and (0,0,0) for their X,Y and Z rotations. This is good practice and helps the animators during the animation phase, as they can quicky pose and repose the character to the default bindpose with ease, when they "zero-out" the rotations on the skeletons.

To finalize and publish our new crowd character model, I wrote three MEL scripts to perform certain routine tasks, on all our crowd characters. The scripts are :

1. **Hierarchy.mel**: This script traverses through the modified animation skeleton's hierarchy and recreates it using the same naming conventions but with a unique prefix, to avoid a name-space conflict. This naming convention also helps us to animate individual characters in the crowd without any difficulty. The new skeleton thus created has (1,1,1) for scale, and (0,0,0) for rotation, on their corresponding X,Y and Z axis.

2. **exportSkinWeights.mel**: Since we just created a new clean skeleton, we would like to use this as the animation skeleton. I wrote a script, that will export the skin weights from the old skeleton to a text file.

3. **importSkinWeights.mel**: Since the new skeleton would be worthless without the weights information, I wrote this script to read the text file and assign the weights to the new skeleton in a similar fashion, in other words binding the new skeleton to the crowd character.

At this point, we have successfully created our crowd model. This process of creating the crowd starting from a base character is completely flexible and automatized in varying degrees, and we have three options to choose how we generate our crowd characters. They are:

- **Manual System:** The process of generating crowd characters, can be manual, so as to have complete control over the crowd characters. Mostly employed when, when the camera gets very close to the crowd.
- **Automated System:** This method can be easily automated, to generate any number of crowd characters.
- A **Hybrid System:** where crowd characters are generated randomly and then can be manually handpicked to suit the specific requirements of the project.

## D. Crowd Modeling: Results

Facial Animation for crowd characters: Figures 19, 20 and 21 , demonstrates the results of a successful integration of facial animation into a crowd of articulated characters. The generic character is on the left along with four other crowd characters.

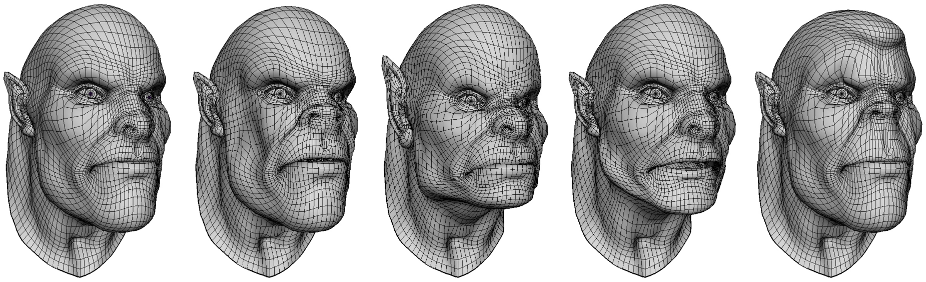

Fig. 19. Facial animation for crowd characters, set 1. This examples demonstrates that ability to implement a facial animation system for our crowd characters. Base/Generic character in on the left along with four crowd characters.

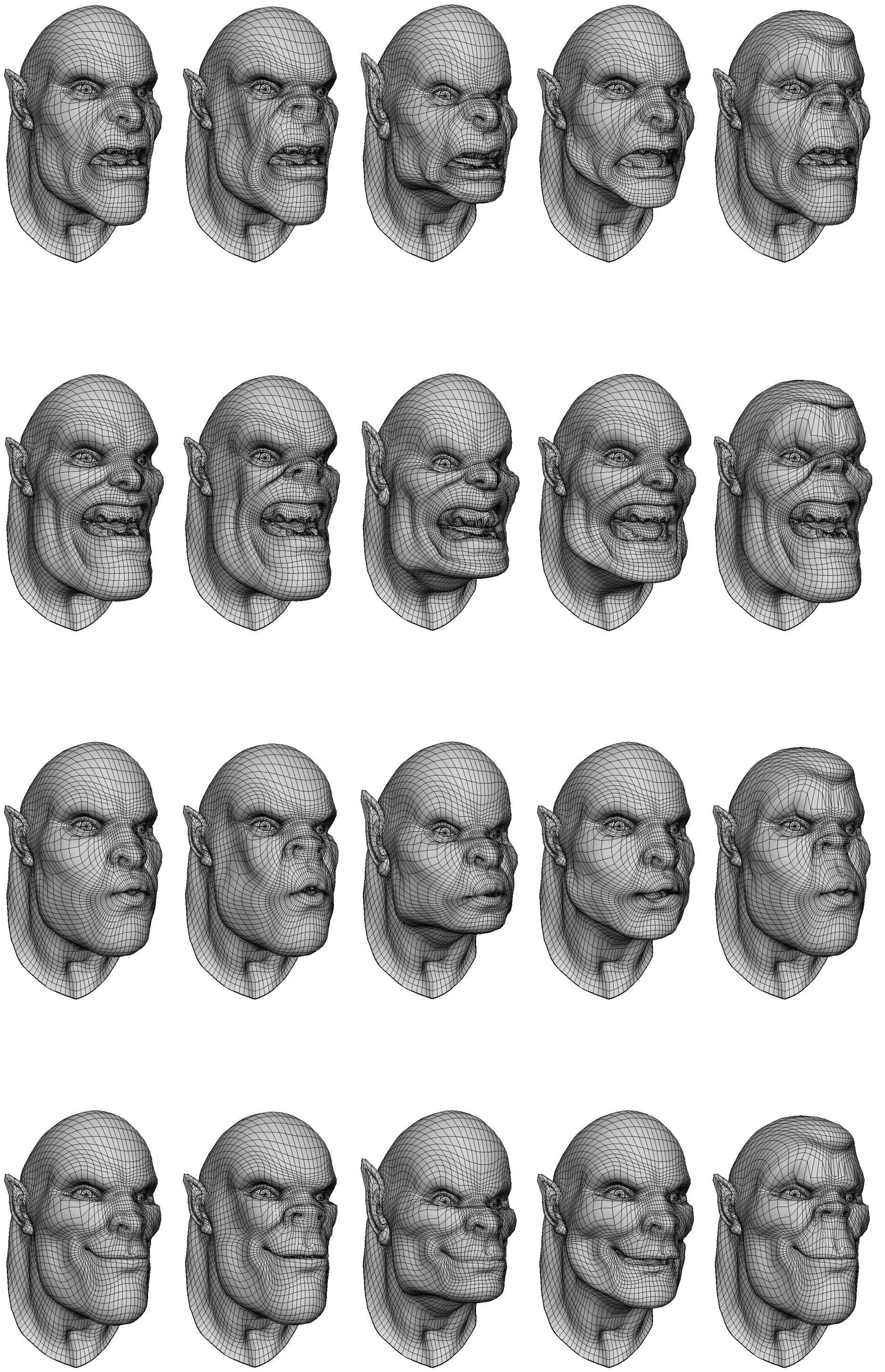

Fig. 20. Facial animation for crowd characters, set 2.

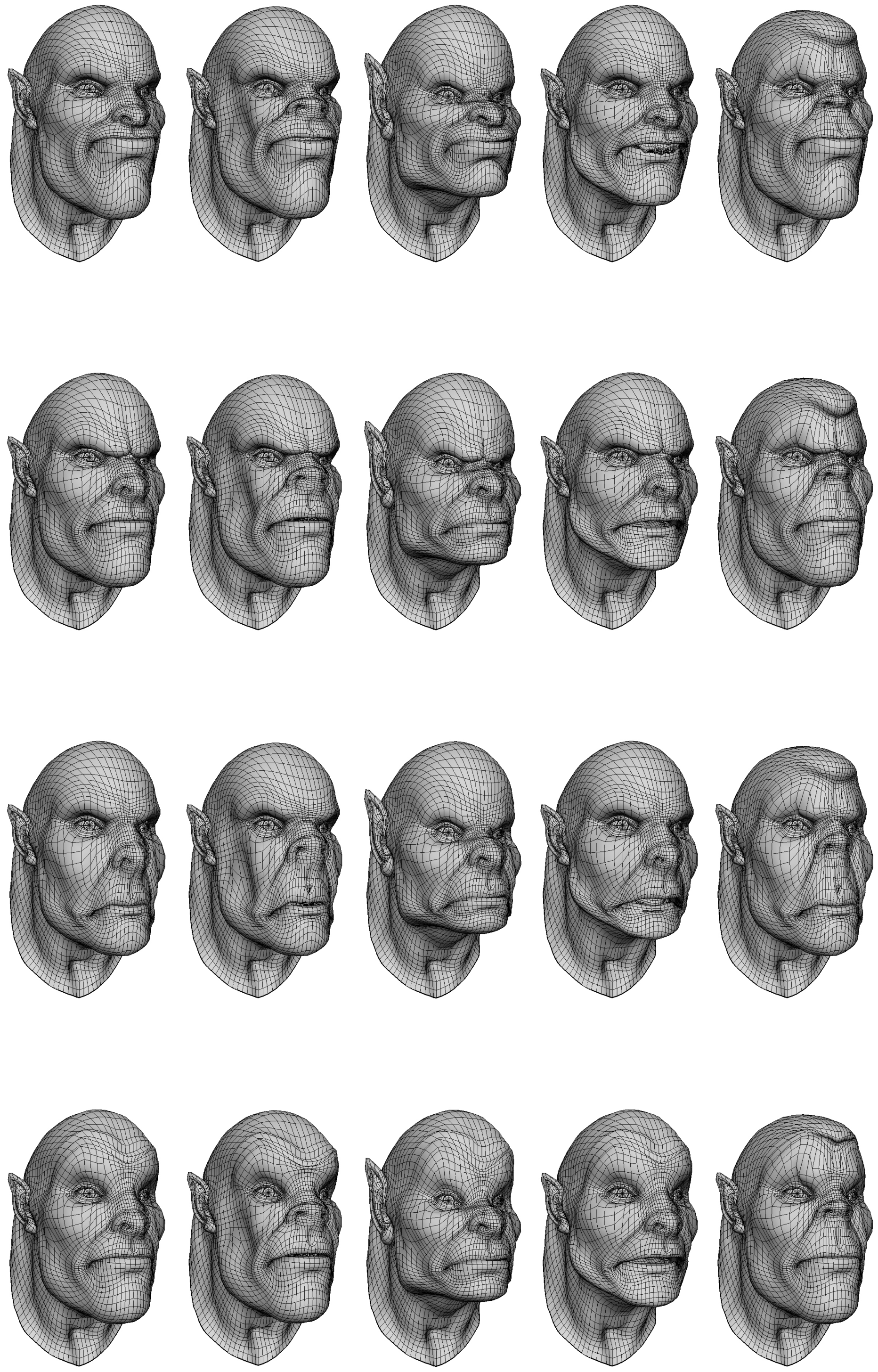

Fig. 21. Facial animation for crowd characters, set 3.

### E. A Crowd of Articulated Characters:

Modeling skeleton, was used to create subtle differences in the face to create localized detail on the character model, as in Figure 22 and 23. The result, produce a wide variety of characters that look completely different from one another, which is very important when modeling a crowd of humanoid characters.

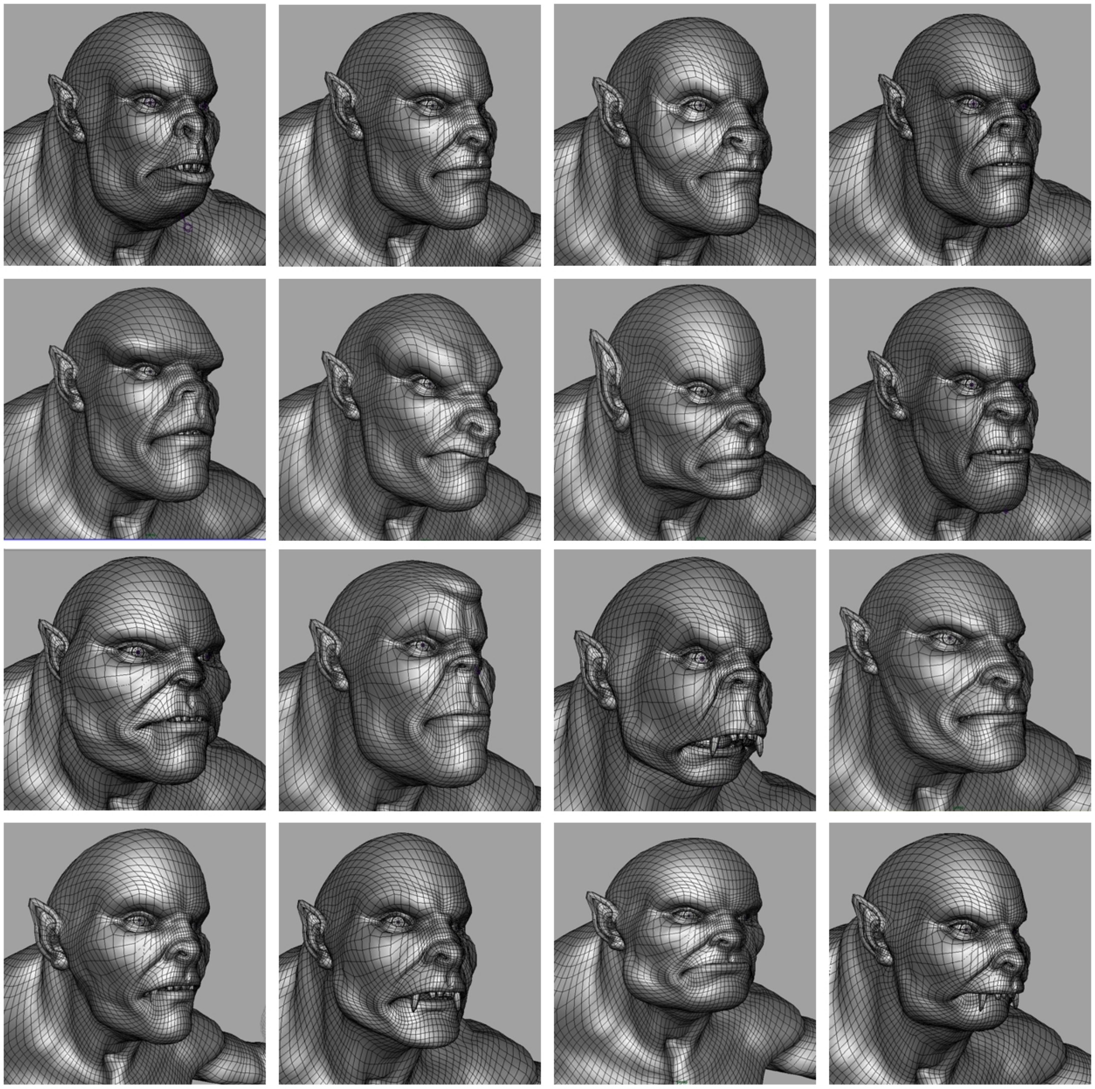

Fig. 22. Modeling skeleton. The effective use of modeling skeleton is demonstrated in this example.

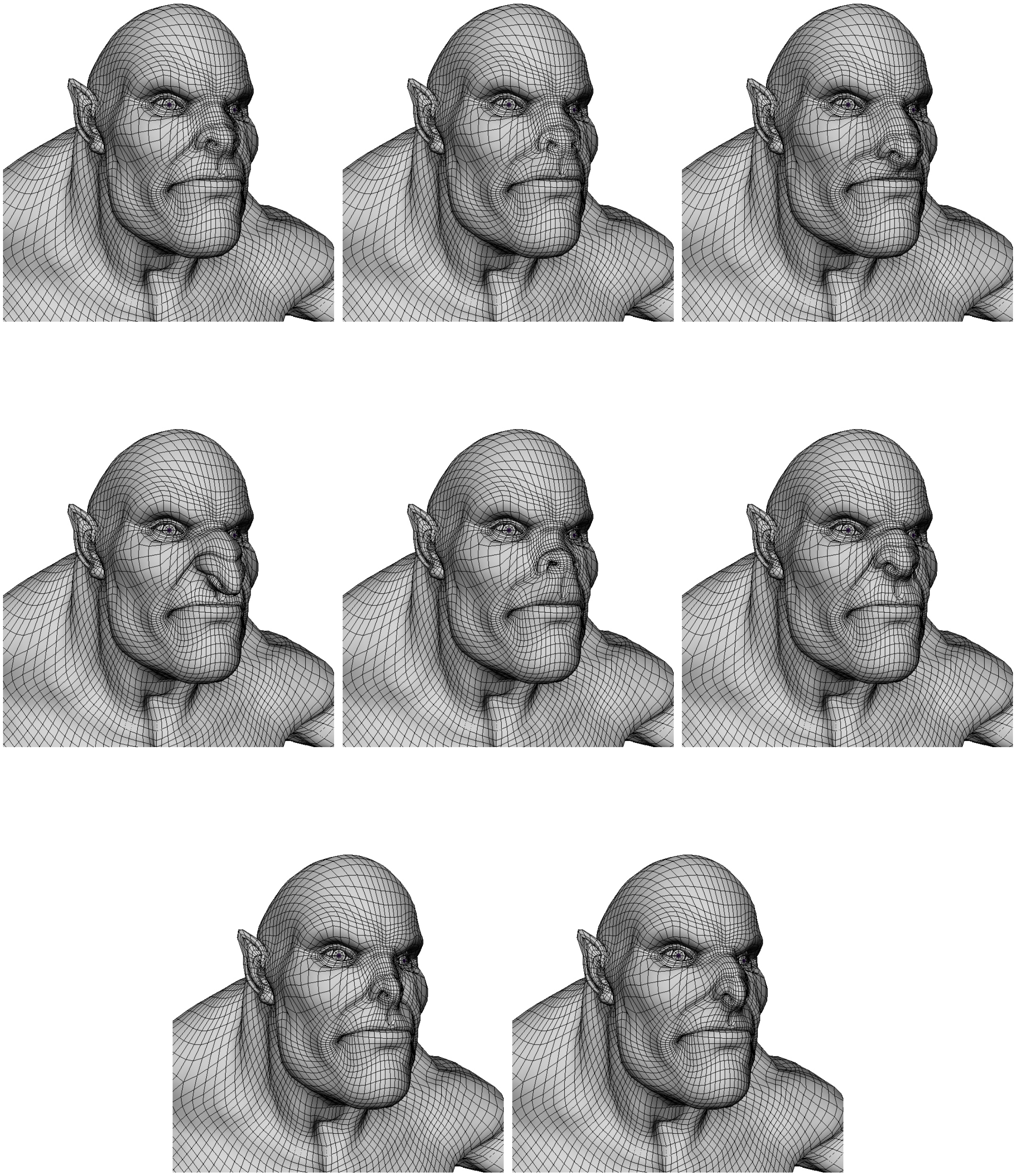

Fig. 23. Modeling skeleton. Using the modeling skeleton, localized detail can be added to the character to create a wide variety of nose shapes. They can then be added seamlessly to the crowd modeling pipeline as a morph targets.

## F. A Swarm of Digital Bees

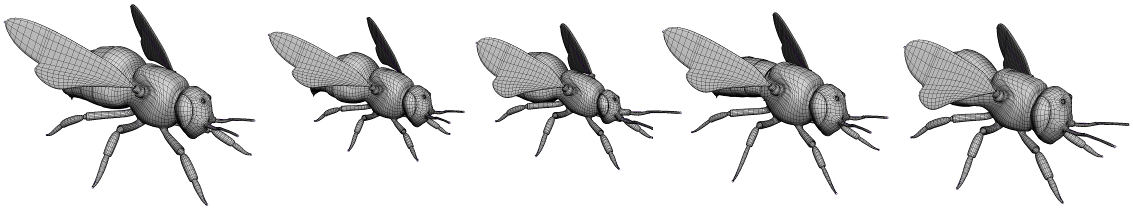

Fig. 24. The same anatomical approach to crowd modeling can be extended to create a swarm of digital bees.

## G. A Colony of Ants

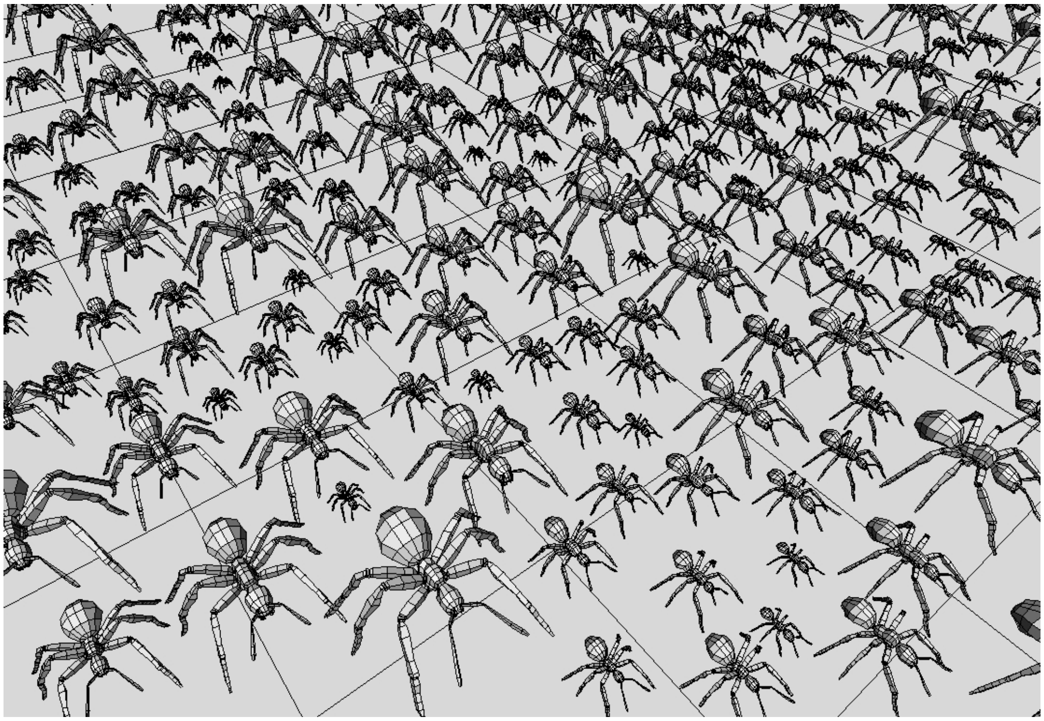

Fig. 25. The colony of ants modeled using this thesis.

# CHAPTER VI

# CONCLUSION AND FUTURE WORK

The crowd modeling idea presented in this thesis is found to be quite effective on a wide range of characters and holds good under different situations and circumstances. The range of characters produced by this technique, honestly surprised me and then I realized the power and potential of the system. It is easily adaptable to almost any kind of pipeline, and can be modified to seamlessly fit any production pipeline. This method is modular and is totally flexible, and demonstrates that articulation can be an effective modeling tool, if used correctly.

This method has an added advantage, in that it can be modified to:

- Create individual characters by hand, carefully sculpting and creating them so as to have full control on how they look, and that they can be very close to the camera.
- Create individual characters based on ”characters already create manually”, and using some random functions with standard deviation and variance, to create the new crowd elements. In other words, it can be a system that learns from previous examples to create new models on the fly.

## A. Future Work

Crowd modeling is still at a nascent stage in the history of computer graphics. As the computing resources increases, and filmmakers expect more realistic and more from realistic crowds, crowds with a wide range of behavior, and acting capability, this field is going to continue to grow and expand.

My thesis is only a small step towards creating anatomically correct crowd of characters from a few base character. The realistic crowds, that I have been able to create, along with the state of the art AI-software, namely Massive [49], should lead us to realistic crowds and believeable animation from every character in the crowd. This also could take the camera deep into the crowd, now that we treat the crowd characters as a secondary character. With time, the method/idea mentioned above, or its deviant, will be the standard method that companies would employ creating secondary characters in movies. It is fast efficient and work that goes into it is effectively re-used multiple times.

With crowd modeling becoming more and more important and significant, I would expect research to continue in this area. New challenges will force technical directors to come up with innovative and effective ways to solve the particular issues and problems. Some of the improvements as I see them now are listed in the following section.

- **Level of Detail**: A system based on LOD (Level of Detail) would surely be one of the hot topics of future research and study in crowd modeling. Characters away from the camera, should automatically be lower resolution models, but at the same time its should also be articulated for animation. The lower resolution models can be **normal mapped**, so they look like high resolution models, closer to camera. Ideally, this low-res swapping should happen at render time to give us the necessary interactivity when opening and handling heavy crowd scene files. This would mean bigger and larger armies, but effectively no overhead in respect to system requirements and file management.

- **Smart Instances**: In our crowd model, all the characters share the same topology. Since it is created from the same base model, they can be instances of the same geometry instead of duplicate copies. The instances can have a vector offset for every vertex, so every character would look different.

- **Superficial Details**: "Superficial" details like facial hair and other attributes like color, length, hair type etc can be added to the crowd characters to create a wider range of characters. Elements like caps, tunics/dresses, jewelery, shoes and appendages like battle axes, swords and shields can be added to make it look organic. These extra appendages can also be modeled using articulation, but that will be a separate study of its own.

- **iMation: interactiveAnimation**: While working on the topic of crowd modeling, I thought of an easy way to animate the crowd characters. I call it, **iMation**:interactiveAnimation. It is a very simple idea of moving the character on screen (like in maya's interface) in realtime and the animation is saved on playback. I wrote a simple MEL script to accomplish this. I have demonstrated complex animation of a fish swimming with this method. the animation data is on one object and in 2 or 3 channels. Its very efficient to get complex and impressive results. Also it makes the task of animating a crowd of characters much easier. I would like to investigate how to employ this method for different types of characters and for different gaits and walk cycles.

- **Procedural Texture Modification**: Since the crowd characters are created from the basic generic character, they all share the same topology, blendshapes and UV layout etc. Coming up with a system that will handle procedural modification of the skin textures for crowd characters, we can create an organic looking crowd. It will be really impressive to create an procedural eye shader for crowd, so different character have different looking eyes.

- **Video Games**: It would be very interesting to use a low resolution model and implement a similar crowd modeling system, to be used in a gaming engine and see the results real time. The gaming industry, would also appreciate the same ideas of LOD (Level of Detail) and instancing.

- **preVisualization Tool**: Almost all movies nowadays seem to spend time and money on preViz (ie preVisualization). This is to plan as much as possible about specific details like camera moves, on-screen action etc, before shooting the live action plates. This avoids costly reshoots, and production delays and also keeps the director of the movie and the visual effects supervisor on the same page throught production. The above idea of using my thesis idea and incorporating it into a gaming engine, can help the technical director, to pre-visualize the crowd scenes and choreograph actions, with the director of the show and the visual effects supervisor.

# REFERENCES

[1] J. Unay, “Maya MasterClasses Hyper-Real Modeling”, Aug 11th 2004, Los Angeles, SIGGRAPH 2004.

[2] P. Thuriot, “Maya MastersClasses Hyper-Real Body Rigging”, Aug 11th 2004, Los Angeles, SIGGRAPH 2004.

[3] E. Miller, “Maya MastersClasses Hyper-Real Facial Rigging”, Aug 11th 2004, Los Angeles, SIGGRAPH 2004.

[4] T. Dirks, “The Birth of a Nation(1915)”

[ http://www.filmsite.org/birt.html ]

[5] “American Cinematographer”

[ http://www.theasc.com/magazine/dec97/titanic/eect/pg1.htm ] - (Date accessed: 09 Aug 2005)

[6] Barbara Robertson, “After Effects”. Computer Graphics World

[ http://go.cadwire.net/?5444,3,1 ] - (Date accessed: 09 Aug 2005)

[7] D. Thalmann, C. Hery, S. Lippman, H. Ono, S. Regelous and D. Sutton. “Crowd and group animation”, Course #33, Aug 11th 2004, Los Angeles, Proc. SIGGRAPH 2004.

[8] “Shrek” & “Shrek 2”, PDI/Dreamworks, Redwood city, California, USA. Motion Picture.

[9] B. Robertson, “When Harry Met Dobby”, *Cinefex Issue* #93 April 2003.

[10] J. Fordham, “Middle-earth Strikes Back”, *Cinefex Issue* #92 January 2003.

[11] "The Incredibles", Pixar Animation Studios, Emeryville, California, USA. Motion Picture. Special Features on DVD Disc2, 2005.

[12] J. Cantor and P. Valencia, *"3D Short Film Production"*, Course Technology PTR; 1st Edition, 2004.

[13] M.C. Vaz, *"The Art of Finding Nemo"*, Chronicle Books LLC, March 2003.

[14] M.C. Vaz, *"The Art of Incredibles"*, Chronicle Books LLC, September 2004.

[15] A. Amidi, *"The Art of Robots"*, Chronicle Books LLC, March 2005.

[16] B. Curless and M. Levoy. "A volumetric method of building complex models from range images", Proc. SIGGRAPH '96 pg.303-312 1996.

[17] M. Levoy K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller et al. "The Digital Michelangelo Project: 3D scanning of large statues" Proc. SIGGRAPH 2000 pg.131-144 2000.

[18] V. Krishnamurthy and M. Levoy. "Fitting smooth surfaces to dense polygon meshes", Proc. SIGGRAPH 96, pp 313-324, 1996.

[19] A. Lee, H. Moreton and H. Hoppe, "Displaced subdivision surfaces", SIGGRAPH 2000, pg.85-94 2000.

[20] H. Seo and N. Magnenat Thalmann. "An automatic modeling of human bodies from sizing parameters", Proc. SIGGRAPH 03. pg19-26, 2003.

[21] B. Allen, B. Curless and Z. Popoviz. "Articulated body deformation from range scan datas", Proc. SIGGRAPH 02 pp.612-619, 2002.

[22] L.Dekker, "3D Human body modeling from range data", Ph.D. dissertation, Dept. of Computer Science, University College, London 2000.

[23] X.Ju and J.P. Siebert "Conforming generic animatable models to 3D scanned data", Proc. 6th Numerisation 3D/Scanning 2001 Congress, Paris France 2001.

[24] I.A. Kakadiaris and D. Metaxas. "3D Human body model acquisition from multiple views", Proc. 5th International Conference on Computer Vision pp.618-623, Boston, MA 1995.

[25] B. Allen, B. Curless and Z. Popovic. "The space of human body shapes: reconstruction and parameterization from range scans", Proc. SIGGRAPH 03, pp.587-594, 2003.

[26] A. Ford, "Planet Unreal",

[ http://www.planetunreal.com/fordy/gradientssmeshtutorials/uvw.html ] - (Date accesssed: 09 Aug 2005)

[27] F. Scheepers, R. Parent, W. Carlson, S. May, "Anatomy based modeling of human musculature", Proc. SIGGRAPH 97 pg163-172 SIGGRAPH 1997.

[28] J. Wilhelms and A.V. Gelder. "Anatomically based modeling", Proc. SIGGRAPH 97 pg 173-180 1997.

[29] E. Catmull, "A system for computer generated movies", Proc. ACM Annual Conference pp 422-431, August 1972.

[30] N. Burtnyk and M. Wein, "Interactive skeleton techniques for enhancing motion dynamics in key frame animation." *Comm. ACM*, vol.19, no.10, pp 564-569, October 1976.

[31] N. Magnenet Thalmann, R. Laperroere, D. Thalmann, "Joint dependent local deformation for hand animation and object grasping". Proc. of Graphics Interface 98 pg.26-33, 1988.

[32] K. Komatsu, “Human skin model capable of natural shape variation”, The Visual Computer 98, vol.4, no.3, pg265-271 1988.

[33] E. Catmull. “The problems of computer assisted animation”. Proc. SIGGRAPH ‘78, Computer Graphics, Vol.12, No.3 pp.348-353, August 1978.

[34] F. Gracer and M.W.Blasgen, “KARMA: a system for storyboard animation”. Proc. Ninth Annual UAIDE meeting 70, pp 210-255, 1970.

[35] N. Burtnyk and M. Wein, “Computer generated key frame animation”. J. Soc. Motion Picture and Television Engineers 80, pp.149-153, 1971.

[36] N. Burtnyk and M. Wein. “Computer animation of free form images”. Computer Graphics 75, vol.9, pp 78-80, 1975.

[37] W. Reeves, “Inbetweening for computer animation utilizing moving point constraints”, Proc. SIGGRAPH 81 Computer Graphics vol.15, no.3, pp 263-270. August 1981.

[38] D. Kochanek and R. Bartels, “Interpolating splines with local tension, continuity and bais control” Proc. SIGGRAPH 84, Computer Graphics, vol. 18, no. 3, pp.33-41, July 1984.

[39] M. Levoy, “A color animation system based on the multi-plane technique” Proc. SIGGRAPH ’77. Computer Graphics, vol.11, no.2, pp64-71, July 1977.

[40] John Lasseter, “Principles of traditional animation Applied to 3D computer animation”, Computer Graphics, vol 21. no.4 1987.

[41] N.I. Badler and S.W. Smoliar. “Digital representation of human movement”, ACM Comput. Surveys 11(1), pg.19-38, 1979.

[42] A. Hutchinson, "Labanotation", Theatre Arts Books, New York 1970.

[43] J.P.Lewis, M.Cordner and N.Fong, "Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation", Proc. SIGGRAPH 2000, pp165-172, 2000.

[44] P. Sloan, C.F. Rose, M.Cohen, "Shape by example", Proc. of 2001 symposium on Interactive 3D Graphics pg.135-143, 2001.

[45] P.Kry, D.L.James,D.K.Pai. "EigenSkin: Real time large deformation character skinning in hardware", Proc. SIGGRAPH 02, pg.153-160, 2002.

[46] A. Mohr and M.Gleicher. "Building efficient, accurate character skins from examples", Proc. SIGGRAPH 2000, pg. 562-538, 2000.

[47] D. Shay, "Back to Titanic", Cinefex Issue #72 December 1997.

[48] Troy - Warner Bros Entertainment, Motion Picture 2004.

[49] Massive Software [ http://www.massivesoftware.com ]

# VITA

**Karthik Swaminathan**
18331, Roehampton Dr, Apt #422
Dallas, TX, 75252
carthick@gmail.com

**Education**

| | |
|---|---|
| M.S. in Visualization Sciences | Texas A&M University, December 2005 |
| Bachelor of Architecture | Madras University, India, May 1999 |

The typist for this thesis was Karthik Swaminathan.