Skin Deformation and Animation of Character Models Based on Static and Dynamic Ordinary Differential Equations

EHTZAZ CHAUDHRY

A thesis submitted in partial fulfilment of the requirements
of Bournemouth University for the degree of
Doctor of Philosophy

National Centre for Computer Animation
Bournemouth University

November, 2015

Copyright

Acknowledgements

My sincere appreciation and thanks go to my supervisors Dr. Lihua You and Professor Jian Jun Zhang for giving me their valuable advice and support through the course of my Ph.D. I would not have completed my research without their advice and support. I shall never forget the abundant time and effort Dr. Lihua You put into my thesis and the direction from Professor Jian Jun Zhang in my research. I also would like to thank the National Centre for Computer Animation, Bournemouth University and Ms. Jan Lewis for all her support with non-academic matters.

This research supported by grants of the UK Royal Society International Joint Projects[1] / NSFC 2010 and the Sino-UK Higher Education Research Partnership for Ph.D. Studies. In this research, I have gained practical knowledge during a four-month internship with Roxar, UK (Roxar is a subsidiary of Emerson Electric Co., a global diversified company). This internship project sponsored by EPSRC under the Industrial Mathematics Scheme[2]. The project evaluated the efficient use of Graphics Processing Units (GPUs) in oilfield reservoir simulation. This internship helped improve my understanding of

[1] http://skindeformation.bournemouth.ac.uk

[2] https://connect.innovateuk.org/web/mathsktn/project-index

advanced mathematics and solving computational problems. The internship case study and benchmarks can be found in the link below[3].

Last but not least, I would like to thank my parents and family for their support, which made it possible for me to undertake this research. Most importantly, I thank God for giving me wisdom and guidance throughout my life.

[3] http://www.mathscareers.org.uk/wp-content/uploads/2014/05/IP09-017-Roxar_Bournemouth_CaseStudy.pdf

Abstract

Animated characters play an important role in the field of computer animation, simulation and games. The basic criterion of good character animation is that the animated characters should appear realistic. This can be achieve through proper skin deformations for characters. Although various skin deformation approaches (Joint-based, Example-based, Physics-based, Curve-based and PDE-based) have been developed, the problem of generating realistic skin deformations efficiently with a small data set is a big challenge.

In order to address the limitations of skin deformation, this thesis presents a workflow consisting of three main steps. First, the research has developed a new statistical method to determine the positions of joints based on available X-ray images. Second, an effective method for transferring the deformations of the curves to the polygonal model with high accuracy has been developed. Lastly, the research has produced a simple and efficient method to animate skin deformations by introducing a curved-based surface manipulation method combined with physics and data-driven approaches. The novelty of this method depends on a new model of dynamic deformations and an efficient finite difference solution of the model.

The application examples indicate that the curve-based dynamic method developed in this thesis can achieve good realism and high computational efficiency with small data sets in the creation of skin deformations.

Thesis Outline

This thesis consists of seven chapters.

- **Chapter 1 - Introduction:** This chapter provides a brief introduction to the importance of skin deformation, followed by the motivation, objectives and framework of this research. The last section of the chapter highlights the contributions of this research followed by a list of publications.

- **Chapter 2 - Related Work:** Chapter 2 surveys various skin deformation techniques in character animation. It reviews the strengths and weaknesses of existing skin deformation methods, identifies some research challenges, and presents suggestions for future work.

- **Chapter 3 - Accurate Joint Determination:** In this chapter, the importance of correct joint determination is discussed. A new statistical method is used to determine the positions of joints based on available X-ray images. The statistical data obtained can be used in the determination of joint positions of human fingers.

- **Chapter 4 - Relationship between a Skin Surface and Curves Defining the Surface:** This chapter presents relationship between surface curves and the surface followed by a method for defining geometric models, which consists of transferring polygonal models into curve-defined surface models, and converting curve-defined models back to polygonal models.

- **Chapter 5 - Static Curve-Based Skin Deformations:** Chapter 5 outlines a new approach to curve-based skin deformation based on a mathematical model and a fast finite difference solution of static skin deformation. In order to achieve realistic skin deformations, an efficient finite difference solution of the mathematical model is proposed to animate skin deformation of character models. The last section of this chapter highlights the examples which demonstrate this approach.

- **Chapter 6 - Dynamic Curve-Based Skin Deformations:** In this chapter, a new skin deformation technique is presented. This technique is based on a new model of dynamic deformations and an efficient finite difference solution of the model. This method is based on combining data-driven methods and curve-based representations of 3D models, which has many advantages.

- **Chapter 7 - Conclusions: Summary and Future Work:** The final chapter summarises the contributions of this research and presents some suggestions for future research.

- **References**

Contents

List of Figures

List of Tables

Declaration

This thesis has been created by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated.

The X-ray images used in this thesis (Chapter 3: Accurate Joint Determination) were obtained from Southampton General Hospital (*Cardiothoracic Radiology*) department. No personal information has been collected with these images. With their permission, these images have been used for research and publication purpose. Anyone who consults it is understood to recognise that the copyright for these images rests with Southampton General Hospital (*Cardiothoracic Radiology*), which cannot derive from this thesis without their acknowledgement and permission.

Abbreviations

2D	Two Dimensional
3D	Three Dimensional
CPU	Central Processing Unit
CT	Computed Tomography
FEM	Finite Element Method
FVM	Finite Volume Method
GPU	Graphics Processing Unit
LBS	Linear Blend Skinning
MEL	Maya Embedded Language
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PSD	Pose Space Deformation
SSD	Skeleton Subspace Deformation
SSDR	Smooth Skinning Decomposition with Rigid Bones
WPSD	Weighted Pose-space Deformation

1 Introduction

1.1 Overview

Realistic and efficient character animation has become an essential part of computer animation, simulation and games. Improvements in technology have driven consumer demand for greater fidelity, and even with fantastical characters, the expectation is that they should appear real. Faces are supposed to express emotions through mouth and eyebrow movements, limbs need to bend at the joints, muscles to bulge when in use, and soft tissues such as fat to bounce and vibrate when the character walks. The basic criterion of good character animation depends both on the appearance and motion of the character. Greater realism creates greater empathy for the characters and greater immersion for gamers and viewers. This becomes even more important in 'serious games' or medical simulation, where the model must react with a plausible approximation of the real tissue.

1.2 Skin Deformation

Animated characters play an important role in the field of computer animation. The realism of an animated character can be achieved through proper skin deformation of characters. Skin deformation resulting from the movement of characters is one of the most interesting and challenging topics in computer animation. An effective and convincing skin deformation should cover all the

minor details while animating a character. Today, character animation is used in almost every video game.

Currently, character modelling and animation are two separate tasks. First, a skin surface for the virtual character is created. This surface is mapped onto the skeleton and the animator must spend time and effort to deform the skin surface realistically in relation to the motion of the skeleton. The effectiveness of an animated character depends on the realism of this relationship between skin and skeleton movement.

To achieve accuracy in skin deformation is one of the most challenging areas of computer animation given the variety of internal and external forces and factors that can affect a body, whether in motion or at rest. Conventional animation techniques are quite costly, time-consuming and have certain limitations on realism, believability and gameplay. Consequently there is a need to develop a fast, innovative and economical approach that can meet the future requirements of the animation industry. Due to its importance, many approaches to determining skin deformations have been developed, such as: *Joint-based, Example-based, Physics-based, Curve-based and PDE-based techniques.*

Joint-based techniques deform skin surfaces by transformations associated with the joints of the skeleton. This has been popular in the animation industry due to its intuitiveness and high computational efficiency. However, this technique relies on the perception and skills of animators. Example-based methods create new skin deformations from known skin shape examples. When there are sufficient skin shape examples, these methods can generate highly

realistic skin deformations. Physics-based methods consider the physics of skin and underlying tissues. The techniques use anatomy and elastic mechanics/biomechanics of skin deformation arising due to the movements of muscles and tendons. They can improve the realism of character animation. However, they involve heavy computational expense and are not popular in commercial animation. Curve-based methods use a curve network to define a skin surface. The curves in the network are deformed which are used to create a deformed skin surface. Partial differential equations (PDEs) techniques represent skin surfaces which involve two parametric variables. The solution of PDEs requires a set of boundary conditions. Partial differential equations techniques are popular in geometric modelling and computer graphics due to their significant advantage in obtaining smooth surfaces. Various skin deformation techniques in character animation have been discussed in Chapter 2 of this thesis.

1.3 Challenges in Skin Deformation

As discussed above, there is a deficiency in current character modelling and animation techniques. What is needed is a technique which integrates character modelling and animation into the same framework, reducing human labour and raising efficiency. In order to fill this gap, this research uses the skeleton to drive the skin deformations, curves to define skin surfaces, and physics-based curve deformations to improve realism and computational efficiency in creating skin shapes. This new technique can reduce the amount of labour required in animating character models, and further improve the efficiency of physics-based modelling.

1.4 Aims and Objectives of this Research

The aim of this research is to develop an efficient skeleton-driven and physics-based skin deformation technique for character animation through the introduction of curve-based surface modelling and ordinary differential equation-based shape deformation.

This can be achieved through the following objectives:

- To accurately determine joint positions of skeleton-driven skin deformations from X-ray images and statistical analysis
- To develop an efficient skeleton-driven and physics-based skin deformation technique. To achieve this, curve-based surface modelling and differential equation-based shape deformation will be used
- To investigate how to relate shape changes of curves to the surfaces defined by these curves
- To investigate the most efficient method for transforming deformable curves to the polygon model
- To investigate the resolution of differential equations and static and dynamic curve deformations
- To implement the proposed approach in C++

1.5 Framework of this Research

The framework of this research follows the process shown in Figure 1.1 and outlined below.

- Determine joint positions of skeleton-driven skin deformations
- Reconstruct real skin deformation models from photos
- Transfer polygonal models into curve defined models
- Develop static and dynamic deformation models and finite difference solutions of curve defined models
- Transfer the deformations of curve defined models to polygonal models

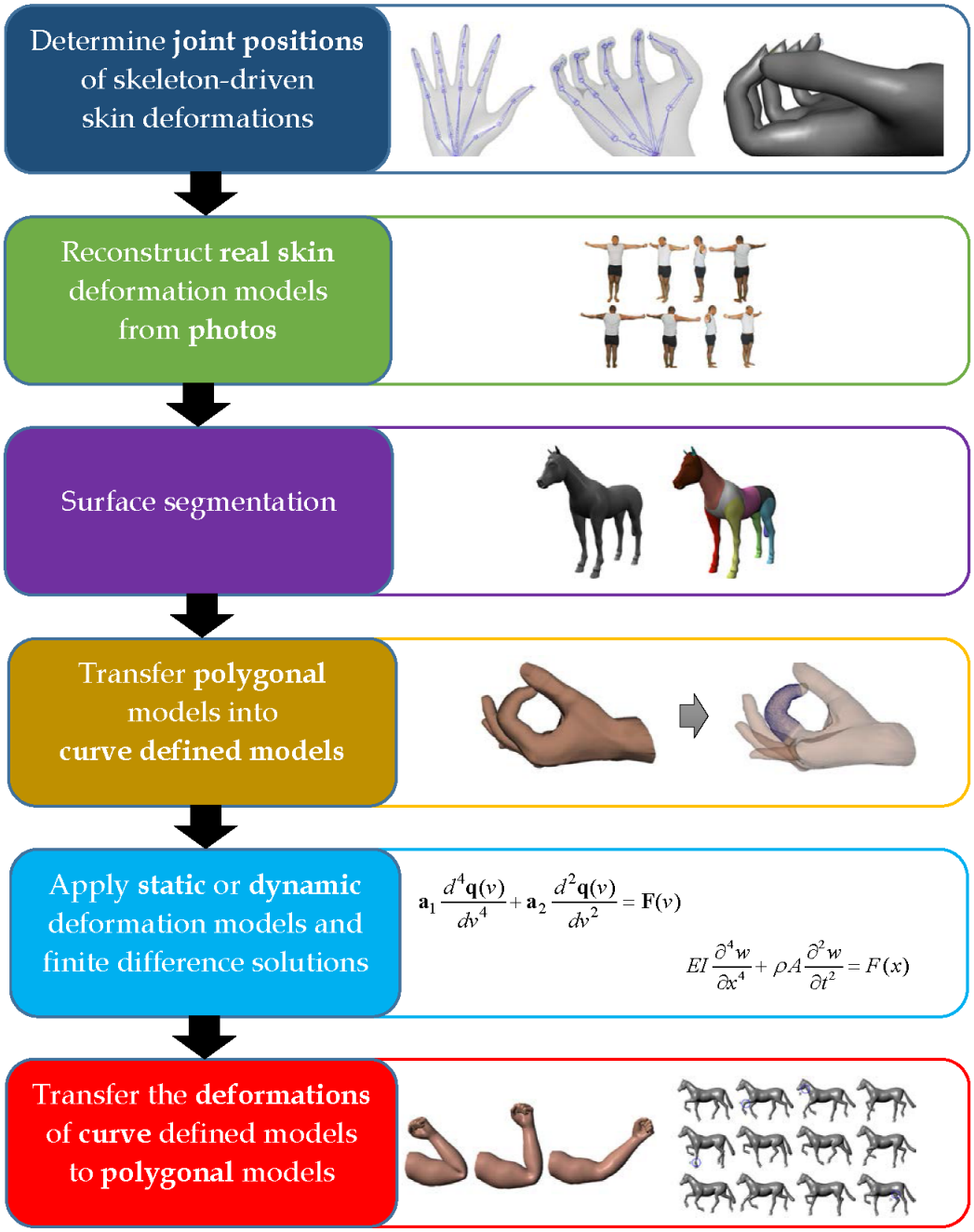


Figure 1.1: Framework of this thesis

1.6 Contributions

The main contributions of this thesis are as follows:

- A detailed review of existing skin deformation techniques.
- Presenting a new method to accurately determine joint positions of skeleton-driven skin deformations.
- Developing a mathematical model of static skin deformations and its finite difference solution: It animates skin deformations by introducing a curved based surface manipulation method, combined with physics and example-based approaches.
- Developing a mathematical model of dynamic skin deformations and its finite difference solution: It further considers dynamic effects leading to a determination of more realistic skin deformations.

1.7 Publications

The research related to this thesis has led to following publications. The first four publications (*1-4*) are directly related to my thesis.

1. **Chaudhry, E.,** Bian, S. J., Ugail, H., Jin, X., You, L. H., Zhang, J. J., 2015. Dynamic skin deformation using finite difference solutions for character animation[4]. In: Computers & Graphics Journal, volume 46, p. 294-305.

2. **Chaudhry, E.,** You, L. H., Jin, X., Yang, X. S., Zhang, J. J., 2013. Shape modeling for animated characters using ordinary differential equations. In: Computers & Graphics Journal, volume 37, no 6, p. 638-644.

3. **Chaudhry, E.,** You, L. H., Jin, X., Zhang, J. J., 2013. Statistical Analysis of Joint Determination for Skeleton Driven Animation of Human Hands. In: Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP), p. 123-126.

4. **Chaudhry, E.,** You, L. H., Zhang, J. J., 2010. Character Skin Deformation: A Survey. In: Computer Graphics, Imaging and Visualization (CGIV), 2010 Seventh International Conference, IEEE, p. 41-48.

[4] http://www.sciencedirect.com/science/article/pii/S0097849314001174

Further, I have contributed to six papers in the area of skin deformation. Publications (*5-10*) are not directly related to my thesis.

5. You, L., **Chaudhry, E.**, Jin, X., Yang, X., Zhang, J. J., 2013. Character Modeling using Physically Based Deformable Curves. In: Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP), p. 119-122.

6. You, L., Yang, X., Jin, X., **Chaudhry, E.**, Zhang, J. J., 2013. Determination of Force Fields for ODE-based and Skeleton Driven Character Animation. In: Proceedings of the International Conference on Computer Graphics Theory and Applications, (GRAPP), p. 115-118.

7. You, L., Ugail, H., You, Y., **Chaudhry, E.**, Zhang, J. J., 2010. Manipulation of Parametric Surfaces through a Simple Deformation Algorithm. In: Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP), p. 84-89.

8. You, L. H., Zhao, H. W., **Chaudhry, E**., You, X. Y., Zhang, J. J., 2009. Surface Modeling with Boundary Constraints and Partial Differential Equations. In: Proceedings of 2009. 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics 2009. IEEE, p. 306-311.

9. You, L. H., **Chaudhry, E.**, You, X. Y., Zhang, J. J., 2009. Physics and Example Based Skin Deformations for Character Animation. In: Proceedings of 2009 11th IEEE. International Conference on Computer-

Aided Design and Computer Graphics, CAD/Graphics 2009. IEEE, p. 306-311.

10. You, L. H., **Chaudhry, E.**, You, X. Y., Zhang, J. J., 2009. Equivalent Deformation Principle Based Skin Deformation of Character Animation. International Journal of CAD/CAM (ISSN: 1598-1800), 9(1), p. 61-69.

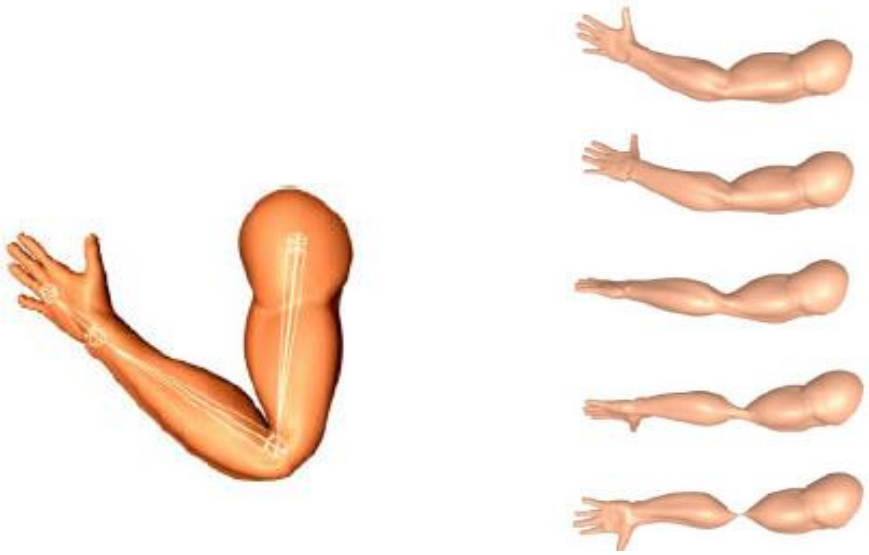
2 Related Work

Skin deformation plays a vital role in realistic character animation. Due to its importance, many skin deformation methods have been developed. Some of these techniques such as *Joint-based, Example-based, Physics-based, Curve- based and PDE-based* techniques are very popular in computer animation. This chapter gives an overview of the various skin deformation techniques in character animation.

2.1 Joint-Based Techniques

Joint-related approaches connect skin deformation to skeleton movement. These techniques are mostly applied in the computer animation industry and are more efficient but less realistic than physics-based and data-driven approaches. These approaches treat the skin as a shell that moves by an explicit function of the skeleton. Vertices of the skin are deformed by a weighted combination of the joint transformations of the character's skeleton Lewis et al. (2000) and Thalmann et al. (1988). Such methods are known as *smooth skinning*. These are easy to understand and intuitive to use. The tedious aspect is a proper assignment of the weights. In the production of character animation, the weights are painted by the animator who has full control over the process of animation. The smooth skinning approach suffers from some serious drawbacks such as the candy

wrapper effect, or collapsing-elbow effect due to non-consideration of volume preservation of soft tissues, as shown in Figure 2.1.



(a) Joint skeleton (b) Joint based twist

Figure 2.1: (a) Collapsing elbow problem in joint skeleton; (b) The candy wrapper problem in joint-based twist (Yang et al. 2006)

Skeleton subspace deformation (SSD) was first proposed in Thalmann et al. (1988). It deforms the skin surface by using a number of vertex weights, that tie the bones and mesh together for a smooth transformation of bones around the joints of a character's skeleton. The vertex weights are obtained from an authoring process. This method is called vertex blending or linear blend skinning (LBS). Many commercial animation software packages use this technique as it is fast and easy to use. Following this pioneering work, Lander et al. (1998, 1999) presented the basic principles of skeleton subspace deformation. The shrinkage problems during bending or twisting were discovered by Weber et al. (1988). In

order to tackle these problems, Wang and Phillips et al. (1988) proposed the multi-weight enveloping technique, Mohr and Gleicher et al. (2003) added additional joints, Yang et al. (2006) introduced curve skeleton skinning, and Kavan and ˇZ´ara et al. (2005) developed spherical blend skinning.

At present, SSD is the predominant approach to character skinning. It is used widely in games, virtual reality and other real-time applications, due to its low cost of computation and easy implementation. However, it suffers from both the candy wrapper effect and collapsing joint problem. Moreover, the determination of blending weights requires considerable skill during skin deformation. The artefact caused by SSD can be improved by using the curve skeleton skinning technique from Yang et al. (2006) or by inverting the skin operations and using the Powell optimisation approach from Xiao et al. (2006).

Among the various methods, classic linear skinning and dual quaternion skinning methods have been implemented in Autodesk Maya. These methods are very popular and efficient. As discussed above, the skinning method determines the deformed position of a vertex by a convex linear combination of individual vertex transformations. This method has the limitations of collapsing joints, candy wrapper effect, and volume loss caused by the linear interpolation of vertex positions. The dual quaternion skinning method of Kavan et al. (2013) provides a solution. It not only interpolates rotations but also blends the rotation centres.

2.2 Example-Based Techniques

Example-based approaches introduce examples of skin shapes to improve the realism of skin deformations. The more examples used, the more realistic the results. In example-based techniques, the artist models different shapes and poses of the characters, or reconstructs them from the data consisting of range scans of a human body in a variety of poses (Allen et al. 2002). New poses are interpolated from these poses (Lewis et al. 2007). A modified least squares fitting technique is used to compute the weights of the deformation and the subsequent skin movement to other animation poses. This approach relies heavily on key sample poses to derive a generalisation of deformation, and this becomes its major disadvantage since it is an expensive and time-consuming process. The example-based methods were developed as an alternative to remedy the artefacts of the other techniques and have had some success. For instance, with the blend shape method, the animator can control the exact appearance of the character. On the other hand, a large number of models have to be made in the pose space and stored for shape interpolation, making it rather expensive (Lewis et al. 2000).

Lewis et al. (2000) proposed the pose space deformation (PSD) technique by combining skeleton-related techniques with shape interpolation. This technique was extended to the weighted pose-space deformation (WPSD) by Rhee et al. (2006). In addition, Mohr and Gleicher et al. (2003) developed an automated method to create skin deformation from a set of examples. Allen et al. (2002) presented an example-based approach to determine joint-related skin deformations. Wang and Phillips et al. (2002) used a modified least-squares fitting technique to compute the weights of a deformation equation and

presented a multi-weight enveloping method to deform skin geometry. Kurihara and Miyata et al. (2004) reconstructed a human hand model from CT scans. Weber et al. (2007) used given deformation examples to develop a system for skeletal shape deformation. Park and Hodgins et al. (2008) created a database of dynamic skin deformations by recording the motion of the skin surface with many motion capture markers, and presented a data-driven technique to synthesise skin deformations by relating static deformations to the poses and dynamic deformations to the actions of muscles. Feng et al. (2008) extracted sparse control points and a skinned mesh with bones and bone influence weights from example meshes, proposed a learning method to capture connections between control points and bone deformations, and generated new deformations from the movements of control points.

Recently, Le and Deng et al. (2012) proposed an automated algorithm called Smooth Skinning Decomposition with Rigid Bones (SSDR) to extract the linear blending skinning from a set of examples. The algorithm can effectively approximate skin deformation by a low number of rigid bones and a sparse, convex bone-vertex weight map. In order to tackle the problem of geometric skinning techniques inability to mimic realistic deformations, and other methods using physical simulation or control volume not being able to deliver real-time feedback, Vaillant et al. (2013) proposed a purely geometric method handling of skin contact effects and muscular bulges in real-time. Neumann et al. (2013) presented a data-driven statistical model for muscle deformation of the human shoulder-arm region. By formulating a linear blend skinning for optimisation and solving it with an iterative rigging algorithm, Le and Deng introduced an

example-based rigging approach to obtain skeleton, joint positions, weights and corresponding bone transformations.

2.2.1 Pose Space Deformation (PSD)

Pose space deformation is used to rectify the limitations of SSD. It combines shape blending and SSD by formulating a scattered data interpolation to overcome problems in sculpted or otherwise obtained example poses (Xiao et al. 2006). Figure 2.2 shows a human arm interpolated model computed by the Powell optimisation method. This method shows that after repeated cycles of M (i.e., dimensionality of function f), the effort to achieve optimisation will be minimal.

Figure 2.2 : Human arm interpolated models computed by Powell optimisation approach (Xiao et al. 2006)

The inverse application of this method presents better performance and more consistent interpolation as shown in Figure 2.2. Pose space deformation can be performed as follows (Xiao et al. 2006).

- Set up the example pose
- In the inverse PSD approach, transfer the example models to the rest pose
- Solve a linear system to obtain weights for all example poses
- For real-time synthesis, obtain the delta in the rest pose by interpolating example poses at run time, and then add this delta to the original character surface and let SSD or any other skinning scheme complete the final transformation
- Inverse skinning will integrate SSD and shape interpolation more firmly

Powell's method for minimisation involves an effort to find each minimum of function f (d') in different directions until f (d') stops decreasing. The method proves that after repeated cycles of M (i.e., dimensionality of function f) line minimisations on conjugate directions, a minimum effort will be needed to achieve optimisation (Xiao et al. 2006). One advantage of Powell's classic method is that it does not need explicit computation of the functions gradient.

SSD - example pose

Example pose

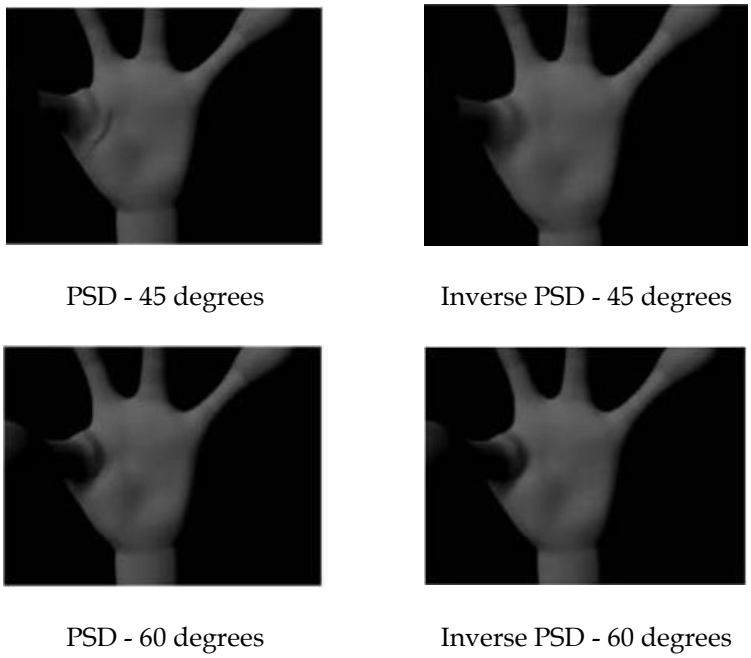


Figure 2.3: PSD and inverse PSD showing better performance (Xiao et al. 2006)

The Powell method minimises the problem. The inverse application gives better performance and more consistent interpolation, as shown in the comparison of PSD and inverse PSD in Figure 2.3. By formulating the inverse process as a minimisation problem, a unified model can be built not only for SSD but for other skinning schemes into which shape interpolation can be incorporated. It is better than SSD, but does not have the provision for re-use when modelling a joint muscle or cloth-like behaviour. PSD is a relatively simple algorithm which uniformly handles a variety of deformations, ranging from the deformation of an elbow to secondary animation (Xiao et al. 2006; Lewis et al. 2000).

2.3 Physics–Based Techniques

Physics-based techniques use anatomy and elastic mechanics or biomechanics of skin deformation originating from the movements of muscles and tendons. Since muscles are made of nonlinear, incompressible, anisotropic and hyperelastic material, existing physical laws still cannot describe the complicated muscle deformations accurately. Moreover, a physics-based approach also involves heavy computational expense.

The physics–based technique requires determination of contractile muscle forces and representation of changing muscle geometry during contraction. This can be achieved by the mass-spring system, the Finite Element Method (FEM) and the Finite Volume Method (FVM).

Wilhelms and Van Gelder et al. (1997) proposed an improved anatomically based approach by modelling muscles, bones and generalised tissues as triangle meshes or ellipsoids, treating muscles as deformable discretised cylinders, and relating skin motion to an elastic membrane model. Nedel and Thalmann et al. (1998, 2000) presented a human model with three layers, i.e., skeleton, muscle and skin, and introduced a mass-spring system with angular springs to physically simulate muscle deformations. Capell et al. (2003) presented a framework to predict skeleton driven dynamic deformations of elastically deformable characters. Venkataraman et al. (2005) combined a kinematic model with a variational model. Galoppo et al. (2007) proposed an algorithm to capture the dynamic skeleton-skin interplay through a novel formulation of elastic deformation in the pose space of the skinned surface. Lee et

al. (2009) introduced a biomechanical model of the human upper body, which can simulate the physics-based deformations of soft tissues.

Recently, Kim and Pollard et al. (2011) developed a fast, physically based simulation system for skeleton driven characters, consisting of a reduced deformable body model with nonlinear finite elements, a linear-time algorithm for skeleton dynamics, and explicit integration. Based on a novel discretisation of corotational elasticity over a hexahedral lattice, McAdams et al. (2011) presented a new algorithm to achieve near-interactive simulation of skeleton-driven, high-resolution elastic models. By introducing a general concept of joint-based deformers, Kavan and Sorkine et al. (2012) found a closed-form skinning method to approximate nonlinear elastic deformations very efficiently. Li et al. (2013) proposed a new approach to simulate thin hyperelastic skin. By optimising both of the control forces added to a linearised dynamic model and material properties, Li et al. (2014) presented a novel method for elastic animation editing with space-time constraints.

Von Funck et al. (2006) employed a divergence-free vector field to define deformations of solid shape which is volume and feature preserving and free of self-intersections. The finite element method is used to simulate elastic and elastoplastic fracturing materials (Debunne et al. 2006; Grinspun et al. 2006; Müller et al. 2002). It is also used for skeleton driven deformations and physically based rigging of deformable characters (Müller et al. 2006; Capell et al. 2006). The mass-spring system was initially used for facial modelling, skin, fat and muscle simulation, but now it is also used for interactive animation of structural deformable objects (Platt et al. 1981; Chadwick et al. 1989; Terzopoulos et al. 1990,

1991; Nedel et al. 1998). In order to simulate large mass-spring systems in real time, a GPU has been developed and implemented to stimulate the elongation and compression of spring on the graphics card and given deformed surfaces (Georgii et al. 2005).

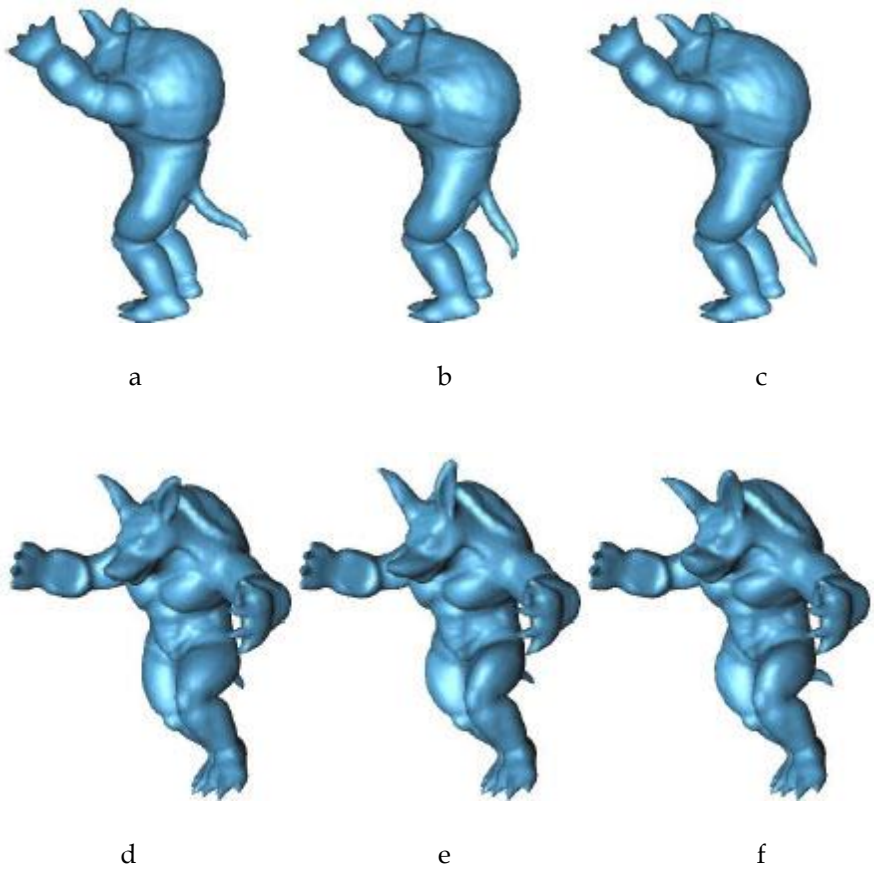


Figure 2.4 : Model based on a physical simulation (Shi et al. 2008)

In FEM, the surface vertices experience elastic forces not only from the surface neighbours, but also from the inside of the material. In order to simulate additional physically based secondary motion on any skeleton-driven animation, a graphics processing unit (GPU) based implementation of the calculated algorithm is used to animate characters in interactive environments such as video

games. This method is quite efficient and can simulate more detailed surfaces in real time, as shown in Figure 2.4. It is compatible with the existing skinning framework and traditional graphics hardware. However, this method cannot be applied to arbitrary mesh sequences. Moreover, it cannot precisely reproduce animation examples. It is also handicapped by explicit integration when applied to very stiff objects (Shi et al. 2008).

The finite volume method also approximates partial differential equations (PDEs) by algebraic equations. For integration of conserved variables in PDEs, volume integrals are converted to surface integrals using the divergence theorem. These terms are then evaluated as fluxes at the surface of each finite volume. A finite volume method based approach is used to simulate the deformable behaviour of skeletal muscles. In order to increase computational efficiency, a number of authors have been investigating adaptive simulation. Debunne et al. (1999) used a finite difference method with an octree for adaptive resolution. This was later improved by Debunne et al. (2000) using weighted finite difference integration techniques to approximate the Laplacian and the gradient of divergence operators. The finite difference method was introduced by Terzopoulos et al. (1999) as a tool to simulate elastically deformable models. The finite volume method provides a simple and geometrically intuitive way of integrating equations of motion with an interpretation that competes with the simplicity of the mass-spring system. However, unlike masses and springs, an arbitrary constitutive model can be incorporated into the finite volume method. The deformed configuration will consider dividing up the continuum into a number of discrete regions each surrounding a particular node.

2.3.1 Physics Based Volumetric Deformation

Volume-based approaches are based on more accurate models, which consider biomechanics and anatomical structures and have a potential to create more realistic skin deformations. Usually, these models consist of a skeleton, muscles, tendons and skin, etc. They obtain skin deformations through solving a volume deformation problem, not a surface deformation problem. Therefore, many volume-based approaches are physics-based, which consider the underlying physics of skin deformations and the effects of other tissues such muscles and tendons on skin deformations.

Volume-based approaches have been investigated by many researchers. How the musculature affects skin deformation of the torso and arm of a human figure was examined by Scheepers et al. (1997). Wilhelms et al. (1997) used a generalised cylinder comprising a certain number of cross sections with fixed vertices. The variation of muscle volume during deformation can be controlled by scaling each cross-section to preserve its area. Scheepers et al. (1997) provided a general muscle model that comprises tubular and bicubic patches. Moreover, they provided the scaling and tension parameters to simulate isometric contractions. The behaviour of various muscle models with flexion at the elbow joint is shown in Figure 2.5. In all these studies, the flexing and bulging of muscle are stimulated by binding the degree of freedom (i.e., scaling and translation or rotation) of each ellipsoid to the degree of freedom of the underlying skeleton joints. Despite their simplicity and attractiveness, the ellipsoid models have failed to depict most of the muscle shapes.

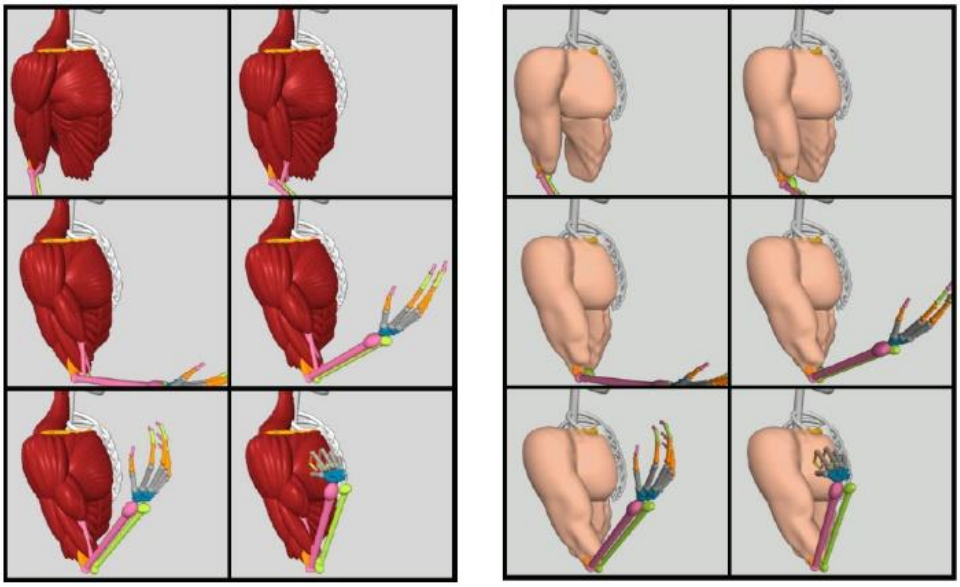


Figure 2.5: Behaviour of the various muscle models with flexion at the elbow joint, and application of a skin and fatty tissue model to muscles of the upper arm and torso (Scheepers et al. 1997)

The muscle layer is the main contributing factor in the surface form (Aubel et al. 1981). Indeed, muscles account for nearly half of the total mass of the body and almost completely fill the gap between the skeleton and the skin (Richer et al. 1997). For purposes of deformation, the muscle is divided into two layers; a skeleton and a surface mesh. Skeletal muscles produce the motion of the bones. Structurally, they consist of a contractile central part referred to as the belly, and often of tendinous extremities that connect the belly to the bones. The attachment to a stationary bone is called the origin, while the other end is termed the insertion. In the constitutive description, the belly consists of bundles of elastic contractile fibres. These bundles are in turn wrapped into a single envelope called fascia. The belly's fibres are responsible for producing the contraction of

the whole muscle. Tendons, which are not very elastic, act as transmitters (Fung et al. 1981).

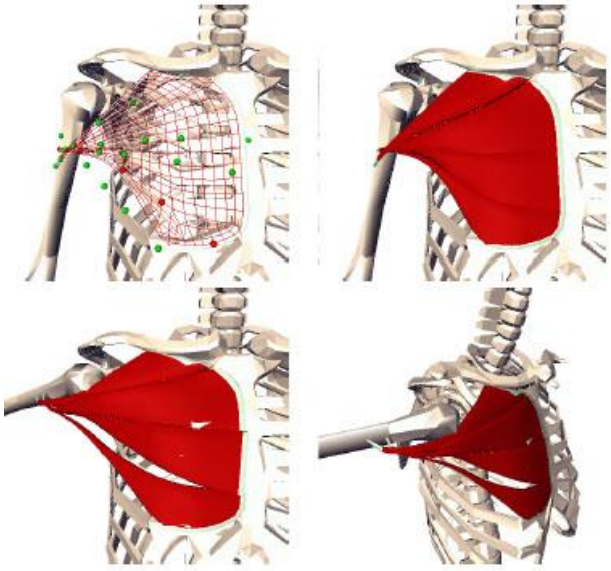


Figure 2.6: Deformation of the pectoralis (Aubel et al. 1981)

Figure 2.6 shows deformation of the pectoralis. Aubel et al. (1981) have studied the anatomy of the human body. There are approximate 700 skeletal muscles in the human body that differ greatly in size and shapes such as gastrocnemius, deltoid, quadriceps, brachialis, biceps brachii and latissimus dorsi. Due to this diversity, simple models such as the ellipsoid are not suitable for muscle modelling (Aubel et al. 1981). Terzopoulos et al. (1990) describe that all muscle tissue exhibits two fundamental properties. It can contract and shorten in length. After contraction, it relaxes and returns to its former length (Terzopoulos et al. 1990).

Jane and Allen (1997) incorporates muscles, bone and generalised tissue. A three-layer model was developed by Nedel and Thalmann (2000) to consider skeleton, muscle and skin. A hybrid model was presented by Guo and Wang

(2005) to create skin deformations from the deformation of chunks by combining a quasi-static linear deformation model and the finite method. Wojtan and Turk (2008) presented a new method which embedded surface meshes into the volumetric tetrahedral mesh which produces a fine fold over time due to periodic re-embedding of the plastic deformations and refinement of large or skinny triangles.

2.4 Curve–Based Techniques

Curve-based approaches use a curve network to define a 3D surface model, and transform a skin surface deformation problem into a curve deformation problem. Such a treatment changes a two-dimensional mathematical problem in a parametric space into a one-dimensional one. Therefore, it can greatly simplify the calculations and raise the efficiency of creating skin deformations. The realism of curve-based approaches is improved by considering the underlying physics of curve deformation and introducing example skin shapes. The curve-based techniques can be classified into *free form wire deformation, purely geometric sweep deformation, cross section method; physics based sweep deformation, static curve-based skin deformation, and dynamic curve-based skin deformation.*

2.4.1 Free Form Wire Deformation

A wire is a curve, the manipulation of which deforms the surface of an associated object near the curve. Wire reflects the geometry of an object and provides a rough geometric representation of an object that create through a sketch, as shown in Figure 2.7. The domain curves define the parameter of deformation of an object. The combination of wires and domain curve provides a new approach

to outline the shape of intended volume in space. Wires can control the variations of geometric representations of the same object, and can be reused on different objects with similar deformable features (Singh et al. 1998). There are two stages in wire deformation. In the first stage, the object is bound to a set of wires, and in the second stage, the manipulation of wires affects the deformation of an object.

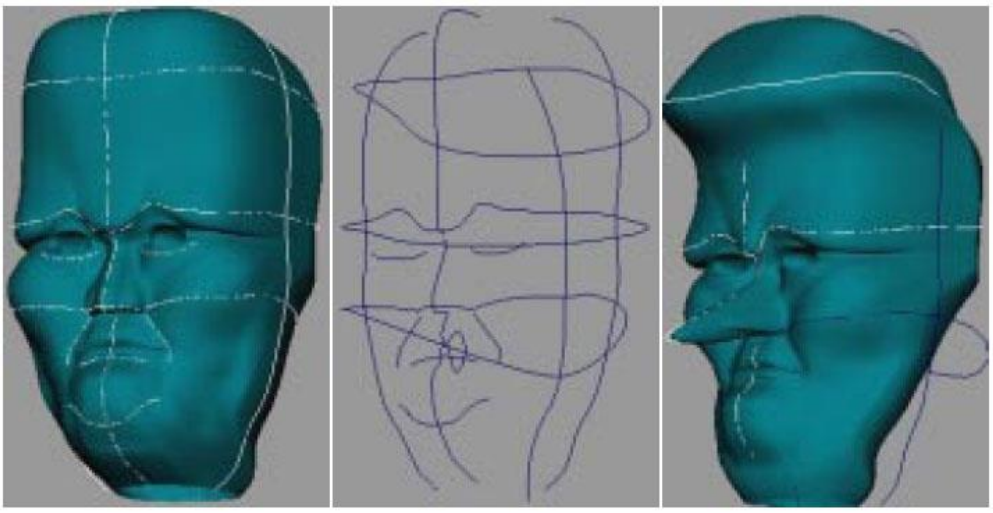
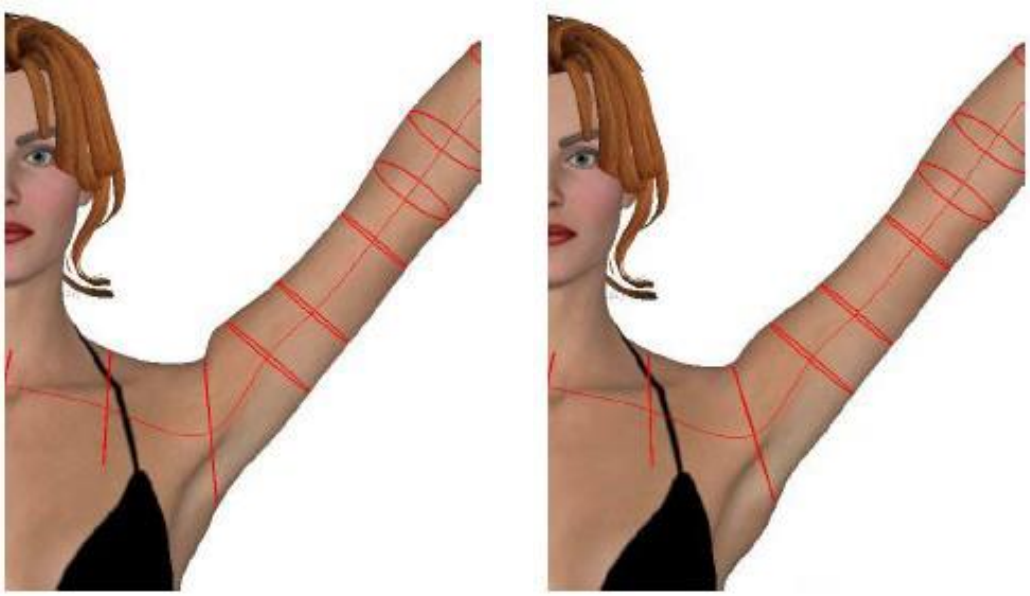


Figure 2.7: Wires: A geometric deformation technique (Singh et al. 1998)

2.4.2 Purely Geometric Sweep Deformation

This technique very nearly generates the parts of a human body, with sweep surfaces that are defined by a moving ellipse. The image shown in Figure 2.8 indicates how to achieve different shapes of the shoulder by editing one of the ellipses used to define the sweep surfaces (Hyun and Yoon et al. 2005).



(a) (b)

Figure 2.8: Editing sweep deformations: (a) an automatically generated shoulder and (b) the result of editing one of its key ellipses (Hyun and Yoon et al. 2005)

2.4.3 Physics-Based Sweep Deformation

This approach introduces physics of curve deformations into the creation of sweep surfaces. In the work of Yoon et al. (2008), a spring-mass system is used to determine curve deformations. The approach set out by You et al. (2007) involves satisfying the boundary constraints, including the position and tangent conditions at the boundaries of a swept surface. This allows different surfaces to modelled and smoothly assembled, leading to the construction of complex objects. This approach is quite effective in automatically rebuilding existing complex models.

2.4.4 Cross Section Method

This method combines the advantages of polygonal and parametric surface approaches, and produces quite realistic body deformations along with surface model display at several levels. The complexity of free-form surface construction and manipulation can often be reduced with the cross-sectional design method (Shen et al. 1994). This approach is used to model and deform human skin. It is especially applicable to human limbs, which exhibit a cylindrical topology. The underlying skeleton provides the natural axes upon which some cross-sections can be formed. To extract cross-section contours from an original polygon body database, the neutral polygon body is first mapped onto the skeleton in initial posture.

2.4.5 Static Curve-Based Skin Deformation

The shape of a surface can be described by some curves on the surface. When the surface moves or deforms, these curves also change their shape. This principle is used to animate surface curves (You et al. 2009), and may be achieved by finding the relationship between surface curves and the surface described by the curves. This is done by obtaining the surface from a polygon model and then transferring the deformation of the surface curves to the polygon model, as shown in Figure 2.9(c)-(g).

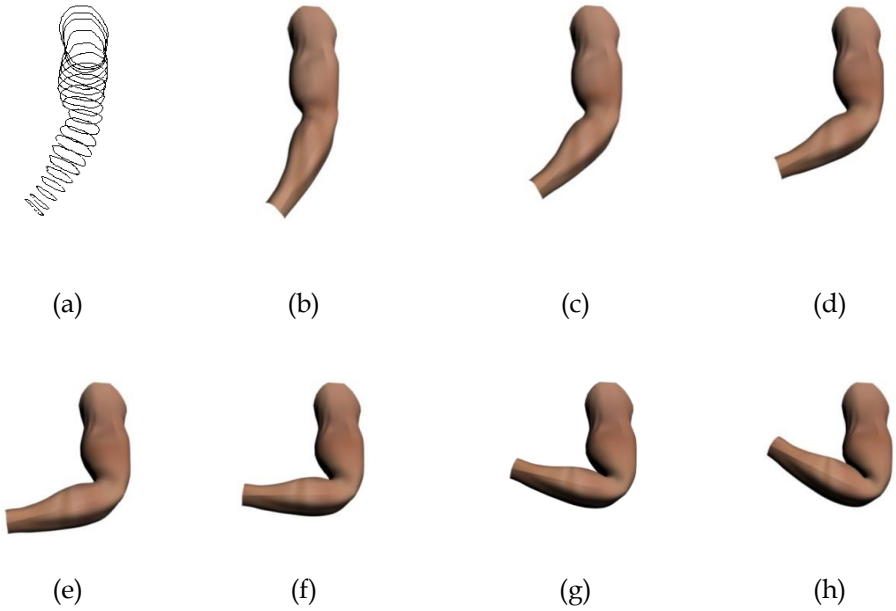


Figure 2.9: Animation of a human arm (You et al. 2009)

Once the surface curves describing a polygon model are obtained, the surface curves can be deformed, and the deformations of these curves transferred to a polygon model to achieve a new shape. Thus, a polygon model can be animated by animating the surface curves describing the model.

2.4.6 Dynamic Curve-Based Skin Deformation

For dynamic deformations, dynamic effects are introduced which consider how velocity and acceleration of a moving skin surface affect its shape. The ordinary differential equation dealing with dynamic deformations is time-dependent (You et al. 2008).

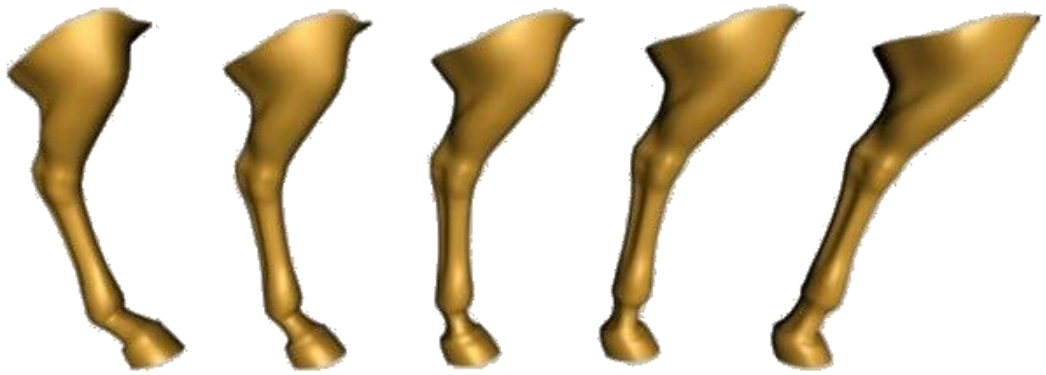


Figure 2.10: Horse leg animation and skin deformation from initial pose at t=0 to t=1 (You et al. 2008)

The solution of the time-dependent ordinary differential equation is used to determine dynamic deformations of these curves. When the surface model moves or deforms, the characteristic curves on the model also change their position and shape. These curve deformations are used to control the corresponding skin deformation, as shown in Figure 2.10.

2.5 PDE-based Deformation Techniques

Partial differential equations (PDEs) techniques are widely used in geometric modelling and computer graphics to represent skin surfaces which involve two parametric variables. The solution of PDEs requires a set of boundary conditions. When physics-based methods are applied to determine shape changes of skin surfaces, complicated partial differential equations are involved which have to be solved by inefficient numerical methods. Partial differential equations-based geomatic modelling techniques were first proposed by Bloor and Wilson et al. (1990 and 1995). Their method was based on the resolution of Fourth order biharmonic PDEs with one vector-valued parameter to create free-form surfaces,

which requires boundary curves as input. This approach generates a surface derived from its boundaries, which are controlled by a relatively small set of parameters compared with the spline-based and subdivision methods. Lowe et al. (1990) presented a method to use PDEs for achieving smooth blending surfaces that satisfy given design conditions, such as geometric constraints, physical criteria and engineering requirements, which can be incorporated into the shape design process. Ugil et al. (1999) have applied the PDE method to interactive design with boundary conditions. This method was defined using curves in 3D space. Figure 2.11 illustrates the edge of the surface patch determine the overall shape of the surface.

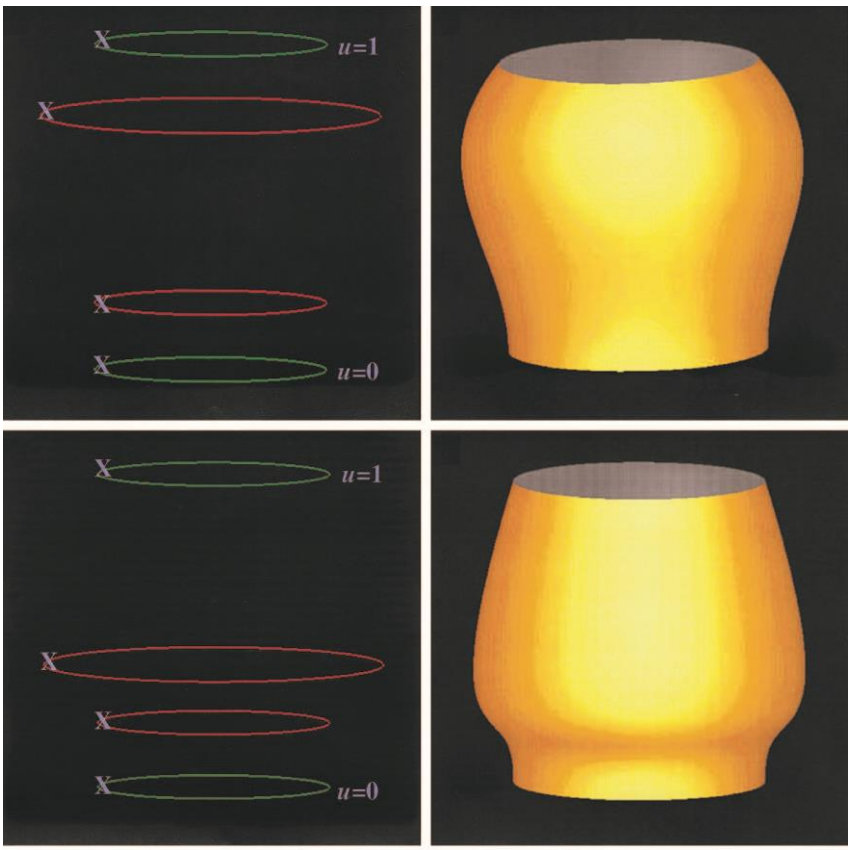
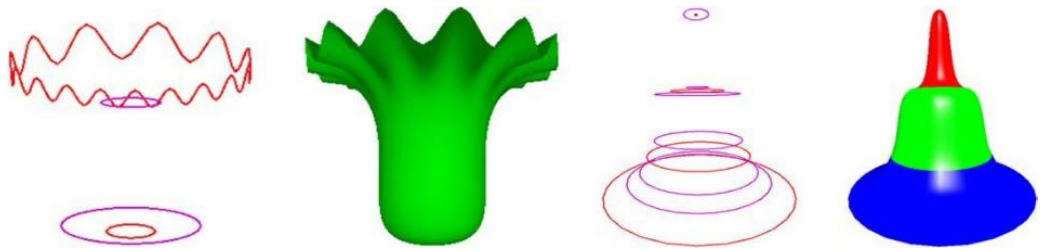


Figure 2.11: The effect of change in the size and direction of the derivative vector on the shape of the PDE surface (Ugail et al. 1999)

Dekanski et al. (1995) applied the PDE method to generate a propeller blade geometry. Du and Qin et al (2000 and 2005) developed a technique for direct manipulation and interactive sculpting based on PDEs and equation of motion. PDE surfaces permit geometric objects to be defined and governed by a set of partial differential equations (Du and Qin et al., 2005). Figure 2.12 shows a PDE surface with local control. These techniques are used to interactively define and change boundary conditions to construct PDE surfaces, and also consider geometric and physical constraints in dynamic PDE-based surface design. Mimis et al. (2001) discussed shape parameterisation of a two-stroke engine and optimisation of scavenging properties of the engine.



a b c d

Figure 2.12: Example of a PDE surface from boundary conditions: (a) boundary conditions, where boundary curves are in red and derivative curves in purple; (b) the surface subject to (a); (c) three sets of boundary and derivative curves for connected PDE surface; (d) the connecting result (Du and Qin et al. 2005)

Zhang and You et al. (2002 and 2004) applied a PDE-based approach in vase design and surface blending with three vector-valued shape parameters. As shown in Figure 2.13 their work demonstrated that various shapes can be

produced by changing shape parameters, force functions and boundary conditions of the proposed PDE.

Figure 2.13: Vase design using PDE (You and Zhang et al. 2002)

Ugail and Wilson et al. (2003) presented an automatic design optimisation method based on PDE formulation, enabling efficient shape definition and shape parametrisation. This method has demonstrated the capability for setting up automatic design optimisation problems. You and Zhang et al. (2003) extended PDE surface modelling from static problems to dynamic ones. Monterde and Ugail et al. (2006) proposed a general 4th-order PDE method to create Bézier

surfaces from boundary information; these surfaces are a special case of Hermite interpolation.

Partial differential equations (PDEs) are equations that *involve two or more independent variables, an unknown function (dependent on those variables), and partial derivatives of the unknown function with respect to the independent variables*. Partial differential equations can be used to describe skin surfaces involving two parametric variables. When physics-based methods are introduced to determine shape changes of skin surfaces, complicated partial differential equations are involved which have to be solved by inefficient numerical methods. In contrast, ordinary differential equations (ODEs) *involve derivatives of an unknown quantity with respect to a single variable*. The parametric representation of curves only involves one parametric variable. When physics-based methods are used to determine shape changes of curves, simple ordinary differential equations are formulated, which can be solved efficiently.

2.6 Cage-based Deformation Techniques

Cage-based deformation techniques use a cage that controls the deformation which can be applied to any closed triangular surface mesh. The cage-based deformation technique was first introduced by Floater and Ju et al. (2003, 2005). Their work are based on an interpolation method for surfaces using Mean Value Coordinates. Joshi et al. (2007) developed another method to set the coordinates, successfully avoiding some of the drawbacks of Mean Value Coordinates. This method was further improved by Lipman et al. (2007), who proposed a positive

coordinates approach using GPU-visibility render for analysing the volume inside the cage to address negative coordinates.

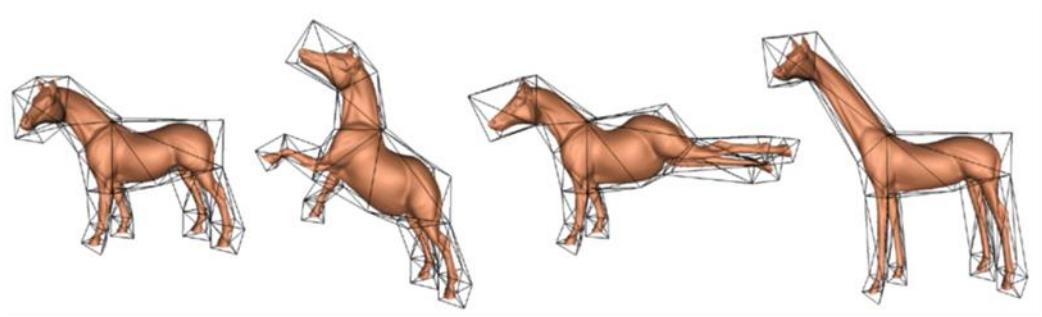


Figure 2.14 : Mean value coordinates for closed triangular meshes (Ju et al. 2005)

Joshi et al. (2005) introduced different cage-based approaches called harmonic coordinates, which do not produce negative coordinates and do not possess local extrema. Figure 2.14 shows Joshi et al. (2005) proposed technique of Mean value coordinates for closed triangular meshes. The advantage of this approach is that it produces a more local deformation. However, it requires more computation time. Lipman et al. (2008) proposed a new method referred to as green coordinates, which uses face normals. This method suggests that more data should be used to be able to relate the cage to the object. A free form deformation method was introduced by Sederberg and Parry et al. (1986), which became popular due to the smooth and intuitive control over the character using limited parameters, or free-form lattice control points. The technique embeds an object in space that is then deformed by moving control points. This model does not need to satisfy other geometric constraints apart from being inside the control lattice. The method has some limitations, for example, it cannot handle complex deformations, such as a character articulated with several limbs. In such cases it

is difficult or even impossible to implement this method. The free form deformation method allows the user to edit the control point and deform the object inside the lattice. However, it very difficult to fit a lattice perfectly into a character shape and to combine several cages (Joshi et al., 2005).

2.7 Summary

In this chapter, different skin deformation methods used in character animation were reviewed and the limitations of each method have been analysed. Each technique has advantages and disadvantages. The curve based methods are the most efficient, but require further investigation with respect to achieving realistic skin deformations. Joint-based methods are generally time-consuming for the animator, and can affect the level of realism. The volume-based methods, on the other hand, use a muscle based system to achieve realistic skin deformation. The physics-based method was introduced to improve realism and incorporate the effect of physical properties like elasticity, gravity and pressure, etc. It also allows for deformations of muscle shape. The 3D nature of the issue of elasticity is reduced to 2D for fusiform muscles and flat muscles.

The curve-controlled techniques described in this thesis are related to this research, but different from the above approaches. The existing work on curve controlled techniques is based on analytical approaches which are not effective in dealing with complicated skin deformations involving concentrated forces, or line distribution forces in a local region. In contrast, the finite difference solutions of static and dynamic ordinary differential equations developed in this thesis can effectively tackle these problems. The new methods developed in this thesis

investigate accurate determination of joints, which has not been examined previously, and apply rigid-body transformations to exclude shape and position changes caused by translations and joint rotations. They also relate sculpting forces and skin deformations to joint rotations, combine the physics-based method with data-driven approaches to achieve realistic skin deformations with only two skin shape examples, and use dynamic curve deformations to achieve high efficiency in dynamic skin deformations.

3 Accurate Joint Determination

3.1 Overview

Skeleton-driven character animation is most frequently applied in computer animation since various commercial animation packages use this technique. Skeleton-driven skin deformation is essential for realistic character animation, as the realism of an animated character depends on the appearance and motion of the character. Current approaches to skeleton-driven character animation have not addressed this issue. In this research, a statistical method is proposed to determine the correct joint positions by way of data analysis of different X-ray joint images. Skeleton-driven character animation involves the following steps. First, a skin surface for the virtual character is created. Then this surface is mapped onto the skeleton. The animator spends a lot of time and effort to deform the skin surface realistically in relation to the motion of the skeleton. The realism of an animated character depends on the accuracy of the relationship between skin and skeleton movement. Most character animation is skeleton-driven, and the quality of such animation depends on correct joint positions. Currently, a joint determination is a manual process where animators place joints onto a 3D model without any reference data. Hence this manual process may not produce correct joint positions, leading to an unrealistic skin deformation. Obtained joint positions have successfully been applied to deform a 3D hand model. The

application example demonstrates that joint positions determined by the proposed method result in more realistic skin deformation.

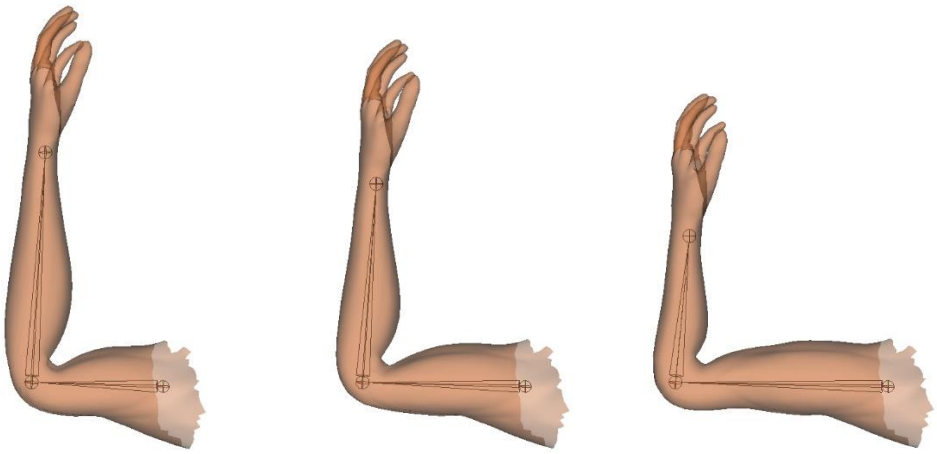
Although various methods of skeleton-driven character animation have been developed, these methods require animators to build the skeleton of the skin surface and determine joint positions manually. Such a manual placement of joints depends on the perception of the animator. Different animators may place the same joint at somewhat different positions. This research demonstrates that joint positions greatly affect the appearance of skin deformation. Therefore, correct determination of joint positions plays a very important role in creating good quality skin deformation. Down-sampling and the use of a database of skeleton templates were mentioned in some existing references; however they are either inaccurate or not available. Therefore, determining the correct joint position is still an unsolved problem.

In order to understand how joint positions influence skin deformation, a test was performed to place the same joint of a human arm at three different positions, as shown in Figure 3.1. These three positions are referred to as top joint, centre joint, and bottom joint, and are shown in Figure 3.1(a), 3.1(b) and 3.1(c), respectively. Using the default rigging method in Maya, the human arm has been rotated at 90 degrees and the skin deformations obtained have been depicted from the side in Figure 3.2. Figures 3.2(a), 3.2(b) and 3.2(c), correspond to Figures 3.1(a), 3.1(b) and 3.1(c), respectively. The images in Figure 3.2 indicate that different positions of a same joint cause different skin deformations. The bulge of the forearm from the top joint is the most obvious, and that from the bottom joint is less obvious. Big differences in the skin deformations of the upper

arm between the three different positions can be clearly observed. Finally, in Figure 3.3 the human arm has been rotated at 120 degrees to obtain skin deformations, where Figures 3.3(a), 3.3(b) and 3.3(c) correspond to Figures 3.1(a), 3.1(b) and 3.1(c), respectively.

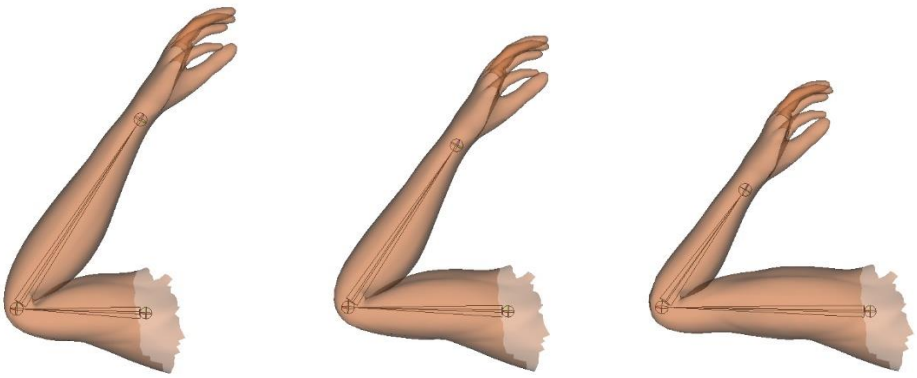
a b c

Figure 3.1 : Top view of three different positions of the same joint of a human arm before rotation a) joint placed at the top position (top joint), b) Joint placed at the centre position (centre joint), c) Joint placed at the bottom position (bottom joint)



a b c

Figure 3.2 : Skin deformation from the side view after 90 degree rotation a) top joint, b) centre joint, c) bottom joint



a b c

Figure 3.3 : Skin deformation from the side view after 120 degree rotation a) top joint, b) centre joint, c) bottom joint

Once again, very large differences in skin deformations can be seen in the three images. The deformed skin surfaces from the top joint contact each other at the region of the elbow joint. However, those of the centre and bottom joints require more joint rotation to move to the contact state. In addition, the three

images also show different shapes of the forearm and upper arm caused by different positions of the top, centre and bottom joints.

From the deformed skin shapes, it can be concluded that different positions of the same joint will cause quite different skin deformations. Correct determination of joint positions greatly contributes to realistic skin deformation.

In order to address this issue, this research has developed a method which presents the relative mean, maximum and minimum positions together with the relative range of joints from statistical analysis of available X-ray images. These data can be used to determine joint position correctly.

3.2 Related Work

Four methods can be used to determine the joints for skeleton-driven skin deformations. The first is the rigging method, the second is the down sampling method, the third is the use of a database of skeleton templates, and the fourth is to estimate joint positions from different poses. The four methods are reviewed in this section.

3.2.1 Rigging method

The rigging method is the only one currently in practical use for skeleton-driven character animation. With this method, the joint positions are completely determined by animators. In many commercially available animation packages, such as Maya, users manually determine the positions of joints. The process relies on the perception and skills of the animator to place the skeleton joints inside the skin surface of character models and then to specifying which parts of

the surface are attached to which bone. More importantly, such a method may produce incorrect joint positions. Different animators may place the same joint at somewhat different positions, which will cause different skin deformations and affect the realism of character animation.

3.2.2 Down sampling method

The down sampling method is used for transforming a curve skeleton into a bone skeleton where the curve-skeleton is obtained through automatic skeleton extraction or construction from 3D skin meshes. Skeleton extraction or construction is used to generate a skeleton. A skeleton can be identified as a curve-skeleton and a bone-skeleton. The curve-skeleton is usually a curve, which is related to the medial axis. The bone-skeleton is a straight line between two adjacent joints and which is used in character animation. The bone skeleton can be readily obtained from the curve-skeleton by down-sampling.

The down sampling method down-samples the curve-skeletons into compatible bone-skeletons based on the bending angles in the different poses. It calculates the bending angle between the adjacent bones connecting one node (Au et al. 2008). For this method (Au et al. 2008), if the maximum bending angle across all examples is less than a threshold, the node is removed and the two adjacent nodes of the node are connected directly. This operation is repeated for the whole curve-skeleton and transforms the curve-skeleton into a bone-skeleton. The advantage of this method is its suitability for all types of models. Its disadvantages are inaccurate determination of joint positions, and failure to

identify the joint where the two bones connected to the joint have no bending rotation, such as a fully extended human arm.

3.2.3 Database of skeleton templates

A database of skeleton templates was mentioned, but not used in the paper by Pan et al. (2009). This method requires a large enough database of skeleton templates capable of covering different types of characters. It makes use of a database to find a skeleton template matching the character model in question and uses the skeleton template to change the curve-skeleton into a bone-skeleton. This method is relatively straightforward and intuitive. However, building such a large database is not an easy task.

3.2.4 Estimating joint positions from different poses

Kurihara and Miyata et al. (2004) presented a method based on measurement of a real human hand in five different poses using computed tomography (CT). From the different reference poses, joint centre and joint angles of each scan were estimated. Then, PSD was used to deform the 3D models.

This method is able to determine the correct joint positions for one person whose X-ray images at different poses are known. However, it has two weaknesses. First, the X-ray images of a same human part at different poses are not always available, and X-rays are harmful to human health. It is difficult to obtain these images from the available sources, such as the internet or hospitals. Second, the method only considers a single individual. Different people have different positions of the same joint. This method does not consider the difference of the positions of the same joint for different people. When animators create a

character model, they still do not know how to determine correct positions of the joints of their model.

The proposed method is related to but different from the method which estimates joint positions from different poses of a single human part. The method is based on a statistical analysis of the existing X-ray images available from the internet and hospitals. It considers the different joints of different people and presents the relative mean, maximum and minimum positions, as well as the relative range of the same joint for different people. When animators create a character model, they can choose from the data or the position between maximum and minimum positions to determine the joint positions of their built model.

3.3 Statistical Analysis of Joint Determination

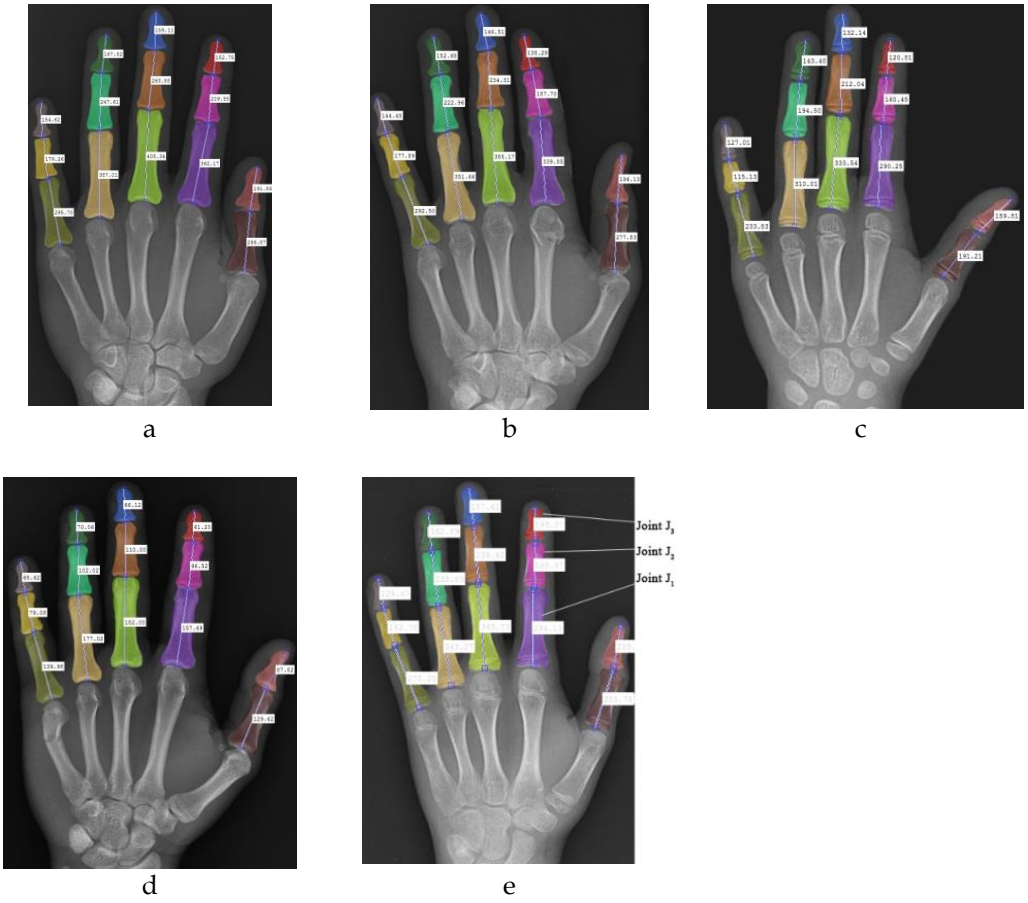
The idea behind the proposed method is to explore the statistical data from available X-ray images of joints, and use these data to guide the placement of joints in character models.

There are many X-ray images available on the internet. An internet search was conducted and the relevant images downloaded. In addition, the Cardiothoracic Radiology department of Southampton General Hospital, UK was contacted and many useful X-ray images were obtained. Figure 3.4 shows the obtained X-ray images of the fingers of a human right hand from five different persons. In what follows, obtaining statistical data from the five images is demonstrated. First, a brief introduction to statistics is given. Statistics is a branch of mathematics which deals with the effective management and analysis of data.

Data can be described in terms of central tendency and variability. Central tendency is an average value of any distribution of data that best represents the middle. Variability describes how spread out or closely clustered a set of data is.

The measures used for central tendency include: mean, median, mode, and midrange. The mean is an average value of a set of data. The median is the number in the middle when a set of data is ranked. The mode is the value that occurs most often in a set of data. And the midrange is the midpoint between the highest and lowest values. In this chapter, the mean is used to describe the central tendency of the data.

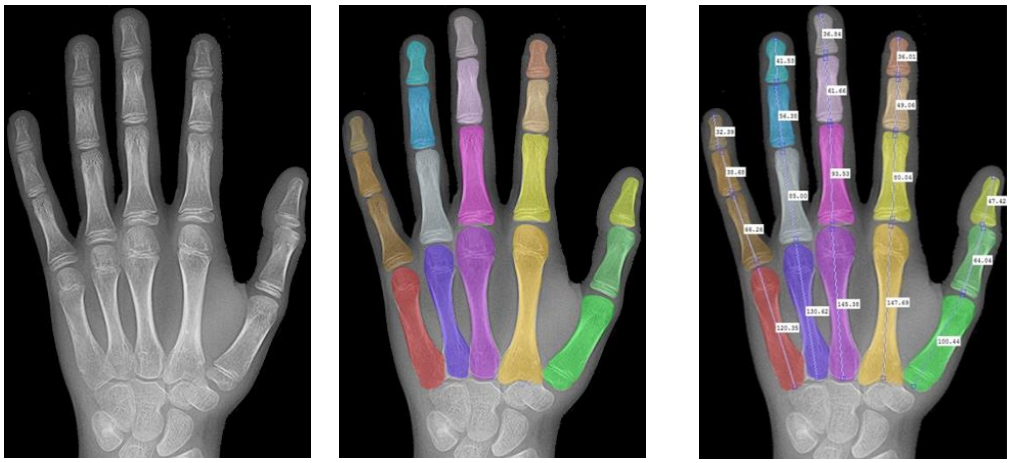
The measures used for variability have range, variance, and standard deviation. The range of a set of data is the difference between the highest and lowest values in the set. The variance describes how the data are spread around the mean. Large variance means the data are widely spread around the mean. And the standard deviation shows how much variation or dispersion exists for the mean. Since the range relates to the highest and lowest values of a set of data, it is suitable for determination of the maximum and minimum positions of joints. Therefore, the range together with the maximum and minimum values is used.



a b c

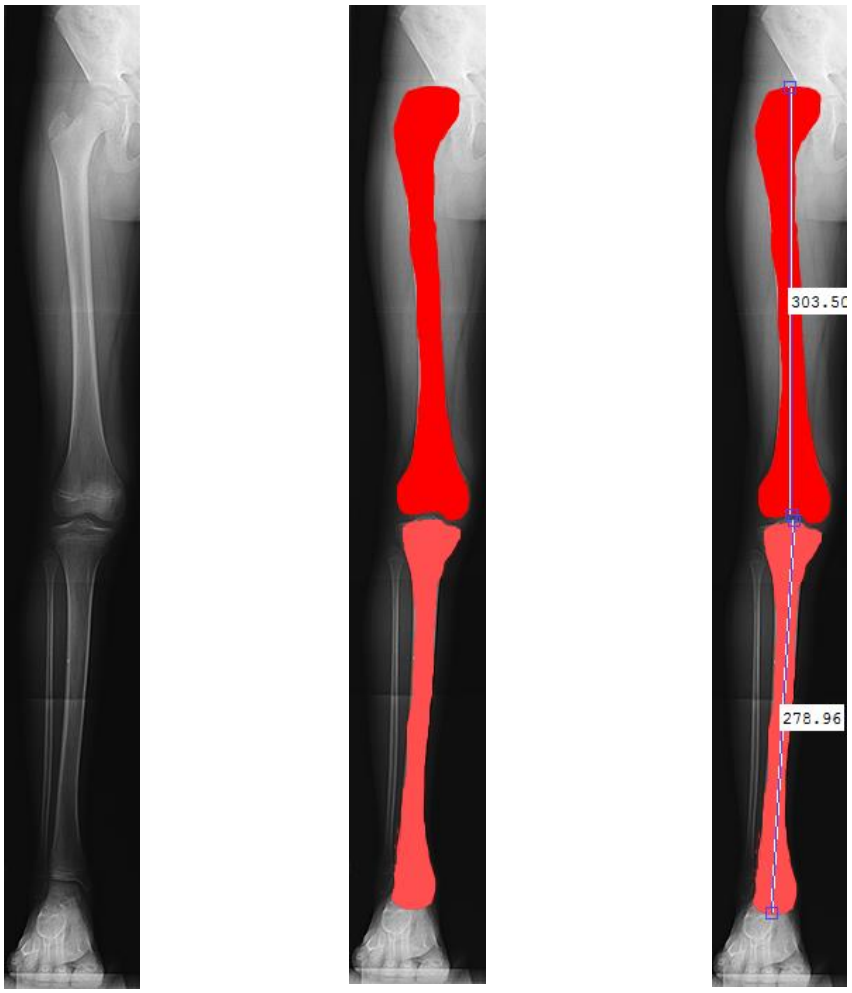
d e

Figure 3.4: X-ray images of the fingers of a human right hand from five different persons



a b c

Figure 3.5 : (a) X-ray of a human hand (b) X-ray of a human hand with colour codes and (c) Computation of length



a b c

Figure 3.6 : (a) X-ray of a human leg , (b) X-ray of a human leg with colour codes and (c) Computation of length

The X-ray images of a human hand were first manually colour coded, as shown in Figure 3.5 (b) and 3.6 (b). Second, the lengths between the joints were calculated automatically using the image processing tool in Matlab, as shown in Figure 3.5 (c) and 3.6 (c). For each of grooming, middle, ring and little fingers, the length between the first joint and the second joint from the root of the fingers is

marked as J_1, that between the second and third joints was marked as J_2, and the length beyond the third joint was marked as J_3. For the thumb, the length between the first joint and the second joint from the root of the thumb is marked as J_1, and that beyond the second joint was marked as J_2. The obtained values of J_1, J_2 and J_3 for all the fingers shown in Figure 3.4 are given in Table 3.1. In the table, the numbers indicate the length between two adjacent joints of the same finger. For example, the numbers 339.96, 188.83 and 169.97 for the grooming finger of Figure 3.4(a) indicate $J_1 = 339.96$, $J_2 = 188.83$, and $J_3 = 169.97$. The total length J of each of the fingers is the sum of J_1, J_2 and J_3, i.e.,

$$J = \sum_{i=1}^{I} J_i \tag{3.1}$$

where $I = 2$ for the thumb, and $I = 3$ for all other fingers.

Then the relative value of each of J_1 and J_2 was determined. Since the position of the first joint will be determined by its position relative to the wrist joint, only the second and third joints which are determined by $\tilde{J}_1$ and $\tilde{J}_2$.were considered. The relative values $\tilde{J}_1$ and $\tilde{J}_2$ of J_1 and J_2 are

$$\tilde{J}_1 = \frac{J_1}{J}$$
$$\tilde{J}_2 = \frac{J_2}{J} \tag{3.2}$$

Taking the grooming finger in Figure 3.4(a) as an example, the total length of the finger is:

$$J = J_1 + J_2 + J_3$$
$$= 339.96 + 188.83 + 169.97 = 698.76 \quad (3.3)$$

and the relative values $\tilde{J}_1$ and $\tilde{J}_2$ of J_1 and J_2 of the grooming finger are:

$$\tilde{J}_1 = \frac{J_1}{J} = \frac{339.96}{698.76} = 0.4865$$
$$\tilde{J}_2 = \frac{J_2}{J} = \frac{188.83}{698.76} = 0.2702 \quad (3.4)$$

One of the advantages using the relative values $\tilde{J}_1$ and $\tilde{J}_2$ of J_1 and J_2 is that the results can be easily extended to other models. For example, one built grooming finger model has a total length of 1000. According to the relative values $\tilde{J}_1$ and $\tilde{J}_2$ of J_1 and J_2 given in Equation (3.4), J_1 and J_2 of the built grooming finger model are found to be $0.4865 \text{ x } 1000 = 486.5$ and $0.2702 \text{ x } 1000 = 270.2$, respectively.

With the same method, the obtained total length J, and relative values $\tilde{J}_1$ and $\tilde{J}_2$ of J_1 and J_2 of the grooming finger are 724.41, 0.4728, and 0.2695 for Figure 3.4(b), 739.78, 0.4862, and 0.2792 for Figure 3.4(c), 826.53, 0.5047, and 0.2731 for Figure 3.4(d), and 793.68, 0.4728, and 0.2708 for Figure 3.4(e).

If there are M X-ray grooming finger images, the relative mean values $\bar{J}_1$ and $\bar{J}_2$ can be determined from the relative values $\tilde{J}_1$ and $\tilde{J}_2$ of the M X-ray grooming finger images through the following equation

$$\overline{J}_1 = \frac{1}{M}\sum_{m=1}^{M} \tilde{J}_{1m}$$
$$\overline{J}_2 = \frac{1}{M}\sum_{m=1}^{M} \tilde{J}_{2m} \tag{3.5}$$

where $\tilde{J}_{1m}$ and $\tilde{J}_{2m}$ are $\tilde{J}_1$ and $\tilde{J}_2$ of the m^{th} X-ray grooming finger image.

According to Equation (3.5) and the five X-ray grooming finger images given in Figure 3.4, $M = 5$ and the relative mean values $\overline{J}_1$ and $\overline{J}_2$ of the grooming finger are

$$\overline{J}_1 = \frac{1}{M}\sum_{m=1}^{5} \tilde{J}_{1m} = \frac{1}{5}(0.4865 + 0.4728 + 0.4862 + 0.5047 + 0.4728) = 0.4946$$
$$\overline{J}_2 = \frac{1}{M}\sum_{m=1}^{5} \tilde{J}_{2m} = 0.2726 \tag{3.6}$$

The maximum and minimum relative values are also given among all the values of $\tilde{J}_k$ of the M X-ray images through the following equation

$$\tilde{J}_{k\max} = \max\left\{\tilde{J}_{k1} \quad \tilde{J}_{k2} \quad \tilde{J}_{k3} \quad \cdots \quad \tilde{J}_{kM}\right\}$$
$$\tilde{J}_{k\min} = \min\left\{\tilde{J}_{k1} \quad \tilde{J}_{k2} \quad \tilde{J}_{k3} \quad \cdots \quad \tilde{J}_{kM}\right\} \tag{3.7}$$

where $k = 1$ is for the thumb, and $k = 1$ and $k = 2$ are for all other fingers.

According to Equation (3.7) and the five grooming finger images given in Figure 3.4, the maximum and minimum relative values $\tilde{J}_{1\max}$ and $\tilde{J}_{1\min}$ for the five X-ray grooming finger images are

$$\tilde{J}_{1\max} = \max\left\{\tilde{J}_{11} \quad \tilde{J}_{12} \quad \tilde{J}_{13} \quad \cdots \quad \tilde{J}_{15}\right\}$$
$$= \max\{0.4865 \quad 0.4728 \quad 0.4862 \quad 0.5047 \quad 0.4728\}$$
$$= 0.5047$$

$$\tilde{J}_{1\min} = \min\left\{\tilde{J}_{11} \quad \tilde{J}_{12} \quad \tilde{J}_{13} \quad \cdots \quad \tilde{J}_{15}\right\}$$
$$= \min\{0.4865 \quad 0.4728 \quad 0.4862 \quad 0.5047 \quad 0.4728\}$$
$$= 0.4728 \tag{3.8}$$

According to Equation (3.9), the relative range for J_1 of the five X-ray grooming finger images is

$$R_1 = \tilde{J}_{1\max} - \tilde{J}_{1\min}$$
$$= 0.5047\text{-}0.4728 = 0.0319 \tag{3.9}$$

Using the above method, the relative mean values and maximum and minimum relative values were calculated together with the relative range of J_1 and J_2 of grooming, middle, ring and little fingers and those for J_1 of the thumb from the five images in Figure 3.4. The resulting statistical data are listed in Table 3.1.

In the following section, additional examples are given to show how the data in Table 3.1 are used to determine joint positions and how these positions are used to create more realistic skin deformations for skeleton-driven character animation.

Table 3.1: Computation of joint lengths of five models

	Grooming			Middle			Ring		
	J_1	J_2	J_3	J_1	J_2	J_3	J_1	J_2	J_3
Figure 4a	339.96	188.83	169.67	375.28	225.60	184.37	358.40	226.47	191.09
Figure 4b	342.47	195.24	186.70	389.42	239.15	197.31	372.98	226.72	212.12
Figure 4c	359.67	206.58	173.53	394.47	260.15	195.31	366.78	247.01	200.40
Figure 4d	417.16	225.70	183.67	446.49	261.72	209.62	442.49	267.65	227.06
Figure 4e	375.25	214.96	203.47	419.47	271.77	223.35	382.13	247.86	219.75

	Little			Thumb	
	J_1	J_2	J_3	J_1	J_2
Figure 4a	284.66	174.72	169.11	270.83	230.95
Figure 4b	291.24	166.72	170.56	259.53	251.34
Figure 4c	289.73	172.52	180.18	264.05	231.33
Figure 4d	334.51	180.32	196.25	301.67	264.72
Figure 4e	294.10	168.67	195.09	285.76	239.81

Once a finger model is built and its total length J is known, the data in Table 3.2 can be used to determine the mean, maximum, minimum and range. Taking J_1 of a built grooming finger model as example, if the total length of the model is $J = 1000$, the mean value of J_1 is $\overline{J}_1 \times 1000 = 0.4946 \times 1000 = 494.6$, the maximum value of J_1 is $J_{1\max} = \tilde{J}_{1\max} \times 1000 = 0.5047 \times 1000 = 504.7$, the minimum value of J_1 is $J_{1\min} = \tilde{J}_{1\min} \times 1000 = 0.0.4728 \times 1000 = 427.8$, and the range of J_1 is $R_1 \times 1000 = 0.0319 \times 1000 = 31.9$, which is the same as $\tilde{J}_{1\max} - \tilde{J}_{1\min} = 504.7 - 472.8 = 31.9$.

Table 3.2: Relative mean, maximum, minimum and range values

	$i=1$				$i=2$			
	Mean $\bar{J}_i$	Max value $\tilde{J}_{i\max}$	Min value $\tilde{J}_{i\min}$	Range R_i	Mean $\bar{J}_i$	Max value $\tilde{J}_{i\max}$	Min value $\tilde{J}_{i\min}$	Range R_i
Grooming finger	0.4946	0.0547	0.4728	0.0319	0.2726	0.3231	0.2703	0.0528
Middle finger	0.4681	0.4864	0.4586	0.0278	0.3135	0.3333	0.2933	0.0400
Ring finger	0.4937	0.0502	0.4618	0.1084	0.2918	0.3449	0.2918	0.0531
Little finger	0.4754	0.5322	0.4529	0.0793	0.2779	0.2869	0.2652	0.0217
Thumb	0.5507	0.6011	0.5172	0.0839				

3.4 Effects of Joint Position on Skin Deformation

In this section, the statistical data in Table 3.2 is used to demonstrate the determination of joints of any built character model and how these data are used to improve the skin deformations.

A hand model with a manually built skeleton found on the internet was used for this purpose. The positions of all joints from the manually built skeleton were measured and are given in Table 3.3. The total lengths J of the thumb, grooming, middle, ring and little fingers are 457.70, 723.60, 747.03, 614.43 and 575.88, respectively.

Table 3.3: Joint positions of manually built skeleton of a hand model

	J_1	J_2	J_3
Grooming finger	285.88	222.89	214.83
Middle finger	249.06	238.14	259.83
Ring finger	223.09	204.38	229.08
Little finger	187.49	186.96	201.43
Thumb	229.43	228.27	

Table 3.4: Joint positions of the same hand model determined by the relative mean

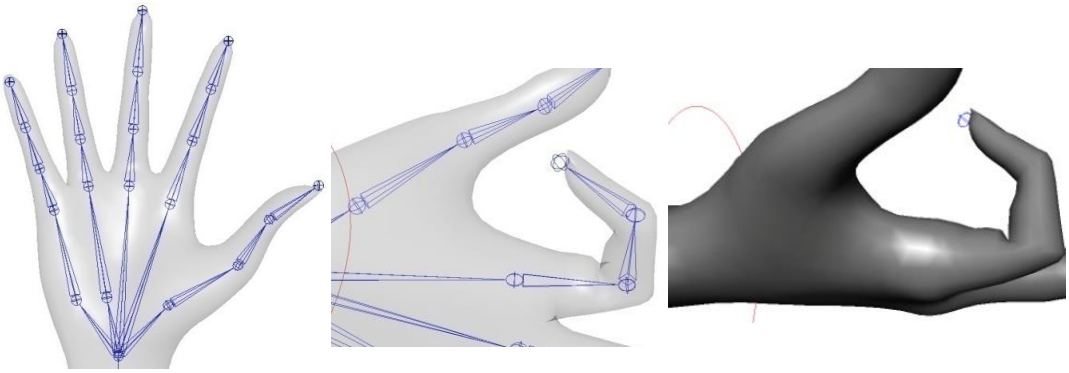
	J_1	J_2	J_3
Grooming finger	357.89	197.25	168.46
Middle finger	369.48	203.64	173.91
Ring finger	303.89	167.49	143.05
Little finger	284.83	156.98	134.07
Thumb	226.37	231.32	

According to these total lengths and the statistical data given in Table 3.2, the mean values J_1 and J_2 of the fingers can be calculated.

Taking the grooming finger as an example, the mean values J_1 and J_2 of the fingers were obtained and which are $J_1 = \bar{J}_1 \times J = 0.4946 \times 723.60 = 357.89$ and $J_2 = \bar{J}_2 \times J = 0.2726 \times 723.60 = 197.25$, respectively. J_3 can be determined by $J_3 = J - J_1 - J_2 = 723.60 - 357.89 - 197.25 = 168.46$.

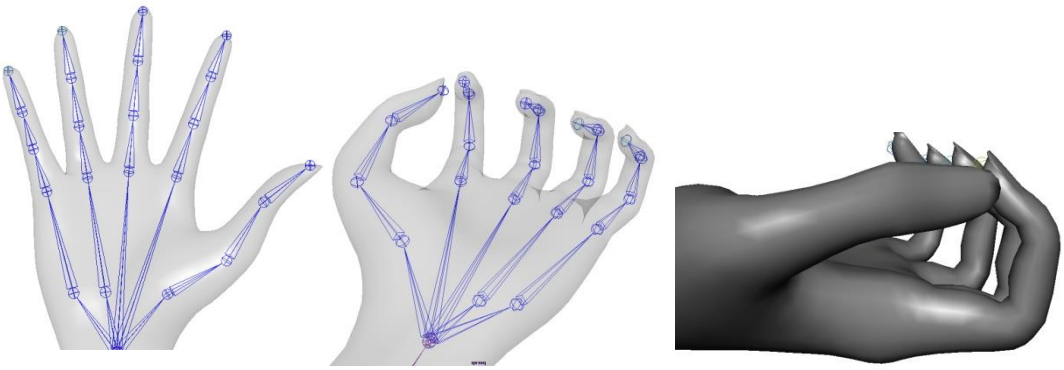
Using the same method, the mean values of J_1, J_2 and J_3 of all the fingers were determined and listed in Table 3.4. Comparing Tables 3.3 and 3.4, for the built hand model, the joint positions of the manually built skeleton found

from the internet were found to be very different from those determined with the statistical method given in this chapter.



(a) (b) (c)

Figure 3.7: (a) Manually built skeleton of human hand model, (b) Human finger animated, and (c) Problem of limb crossover which gives un-realistic skin deformation



(a) (b) (c)

Figure 3.8: (a) Human hand model rigged with correct joint positions, (b) Human fingers animated by the statistical method, and (c) Correct joint positions with realistic skin deformation

Figures 3.7 and 3.8 show a comparison of the human hand model. The skeleton of the human hand model in Figure 3.7 (a) is manually built, in Figure 3.8(a) the skeleton is built using the correct joint positions of the skeleton as

determined by the statistical method given in this chapter. In comparison, it can be seen clearly from Figures 3.7(b) and 3.8(b) that different skin deformations were generated by different joint positions. The skin deformation created by the joints determined with the statistical method as shown in Figure 3.8(c) results in a more realistic appearance than that created by the manually specified joints as shown in Figure 3.7(c).

3.5 Summary

In this chapter, a statistical method to determine the positions of joints based on available X-ray images and statistics has been presented. The statistical data of the joint positions of human fingers has been obtained. This data can be used in the determination of joint positions of human fingers. The proposed method can also be extended to the determination of joint positions of various articulated character models.

The proposed method has been applied to obtained statistical data to determine the joint positions of a hand model found from internet, and a comparison of skin deformations has been made between the manually specified joints and the calculated joints using the statistical method proposed in this chapter. The comparison demonstrates that the joints determined by the statistical method proposed in this chapter can create more realistic skin deformations.

4 Relationship between a Skin Surface and Curves Defining the Surface

4.1 Overview

For curve based surface modelling, the shape of a surface can be described by the curves on the surface. When the surface moves or deforms, these surface curves also change their shape. Therefore, the surface can be animated by these surface curves. In order to use surface curves to animate a surface, the relationship between surface curves and the surface described by these curves must be found. This entails obtaining the surface curves describing a surface from a polygon model, and transferring the deformations of the surface curves to the polygon model.

4.2 Related Work

The relationship between a surface and the curves defining the surface entails obtaining the curves describing a surface from a polygon model, and transferring the deformations of the curves to the polygon model. Some methods have been proposed in the existing references to determine curves on a surface and relate the curves to the surface (You et al. 2008, 2009) and extract the skeleton of a polygon model automatically (Au et al. 2008). The polygonal model can be decomposed into parts according to the extracted skeleton or using certain mesh

segmentation methods. The shape changes of each part can be used to determine the distribution of curves on the polygonal model.

4.3 Defining Geometric Models

Geometric models can be defined by three types of surfaces in Autodesk Maya: *polygons, subdivision surfaces and patch modelling*.

Polygonal modelling is a technique in which objects are represented by polygons. Polygons are made up of vertices, the points in a three-dimensional space. Two vertices connected by a straight line become an edge. A triangle can be defined by three edges, which is the simplest polygon. Another common polygon is a quad, which can be created out of two triangles. Both triangles and quads are used to build three-dimensional models in Autodesk Maya. It is easy to control and work with the polygon shape of surfaces both in modelling and texturing.

Subdivision surfaces create a complicated three-dimensional model from a simple polygon primitive. Two algorithms, interpolation and approximation are used to transform the simple polygon primitive into a high-resolution model with detail.

Patch modelling divides a complex model into many simple surface patches. Each patch is created separately. Then these surface patches are stitched together to represent the model. Currently used patch modelling includes Bezier, B-spline and NURBS. Among them, NURBS surfaces are the most popular.

Three-dimensional character models can be manually created with the above three modelling approaches. They can be also constructed from captured data, such as photos or video streams. Compared to manually created character models, reconstructed character models are more realistic. In this thesis, reconstructed human models are used to provide example skin shapes and ground truth models for comparison. Here I introduce how to reconstruct real deformed skin shapes from photos.

There has been a lot of research on surface reconstruction, especially on facial reconstruction. Different techniques have been developed for various applications. In film and game production, facial markers (Huang et al. 2011), camera arrays (Beeler et al. 2011), and structured light projectors (Zhang et al. 2004) are used to obtain 3D facial geometry of high fidelity. For ordinary users, video-based facial tracking and animation provide a more practical solution. The related techniques include: feature displacements in expression change (Chai et al. 2003), physically-based deformable mesh models (Decarlo et al. 2003), data-driven face models (Pighin et al. 2003), cascaded pose regression (Cao et al. 2003), Constrained Local Model (Saragih et al. 2003), and supervised descent method (Xiong et al. 2003).

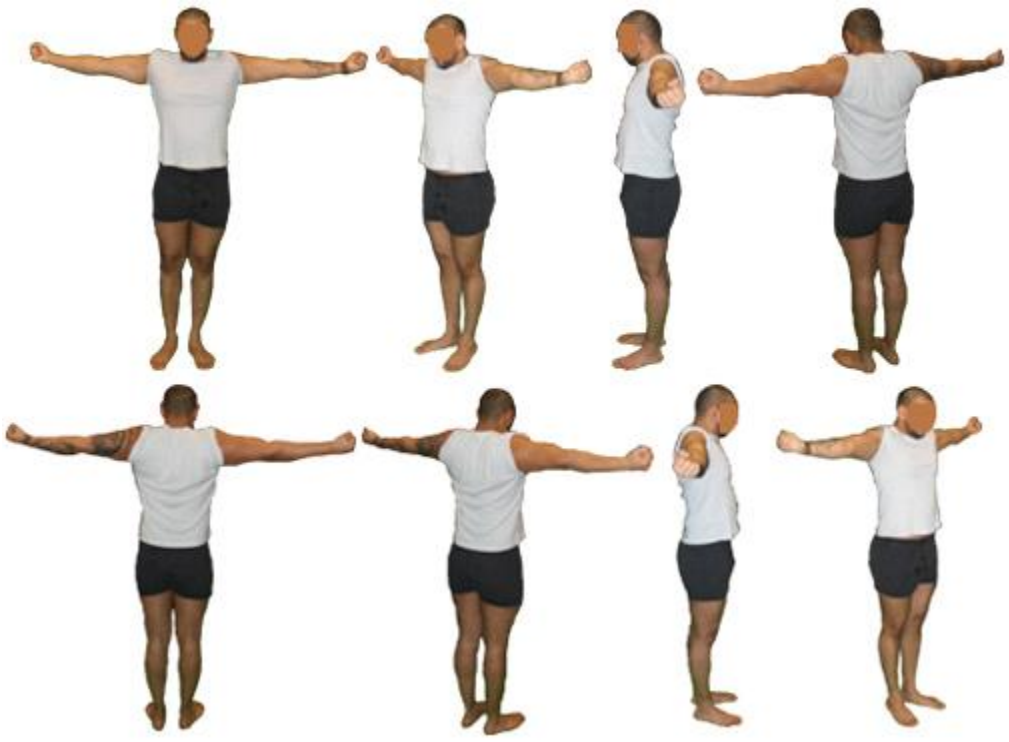
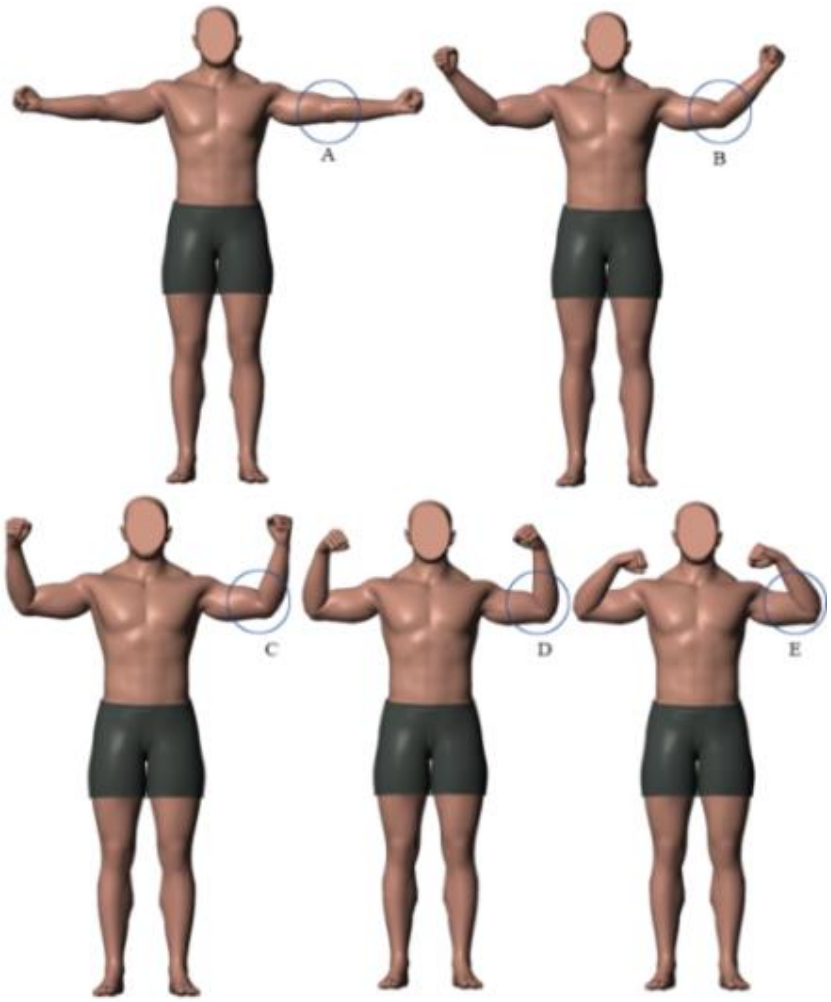


Figure 4.1: Photos of a human male arm at the rest pose

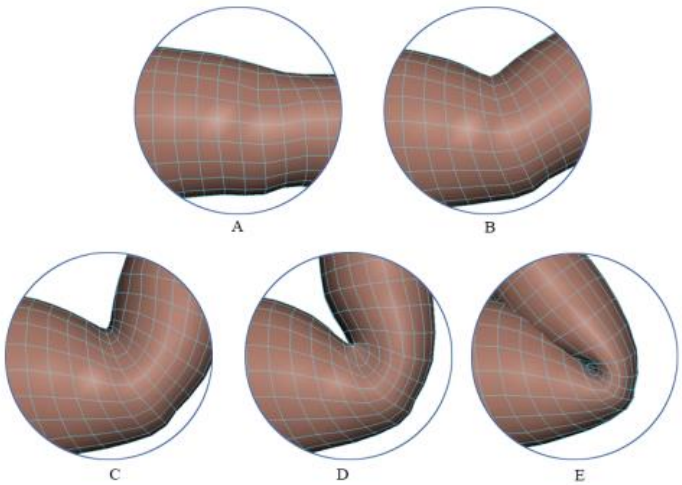
In this thesis, the real skin deformation shapes have been reconstructed from photos taken of human male arm movement. The movement has been divided into numerous poses. At each pose, 20 photos from different views have been taken. Figure 4.1 shows eight photos selected from the 20 photos taken at the rest (initial) pose of the human male arm movement.

Figure 4.2: Skin shapes at five different poses

Figure 4.2 shows photos of skin deformation shapes of the human male arm movement at five different poses. All photos were then uploaded to Autodesk 123D Catch to reconstruct 3D skin deformation models. The reconstructed models were input into Autodesk Maya for further processing. Figure 4.3(a) shows the reconstructed skin deformation models at five different poses obtained from Autodesk Maya. Figure 4.3(b) gives the close-up images of the human elbow in Figure 4.3(a).



a



b

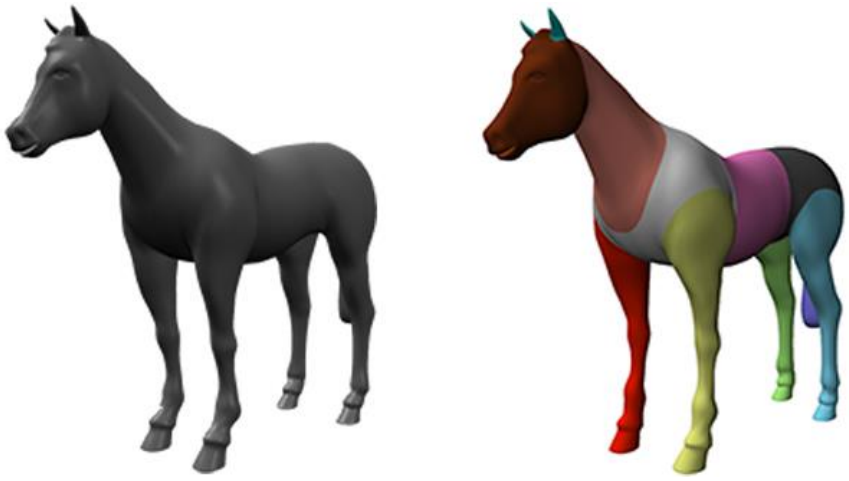
Figure 4.3: Reconstructed skin deformation models

4.3.1 Transferring polygonal models into curve defined surface models

Each of the reconstructed models is a polygon model. It is first transferred into a curve network, i.e., a curve defined model. In order to use the curve network to represent the deformations at the same surface positions of a polygon model but with different poses, surface vertices were used to define the curve network, and to obtain deformations of the curve network at different poses as explained below.

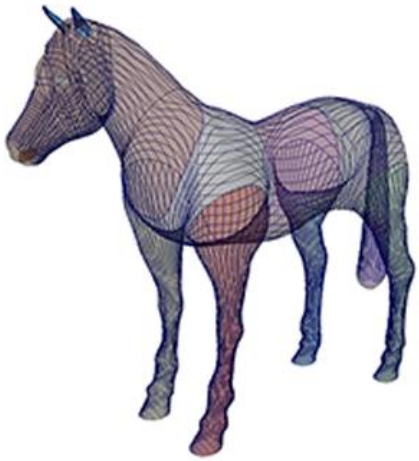
Transferring a polygon model into a curve network is achieved through three simple stages: dividing a polygon model into different parts, obtaining vertex indices of the curves to be extracted, and extracting the coordinate values of the curve vertices from the vertex indices.

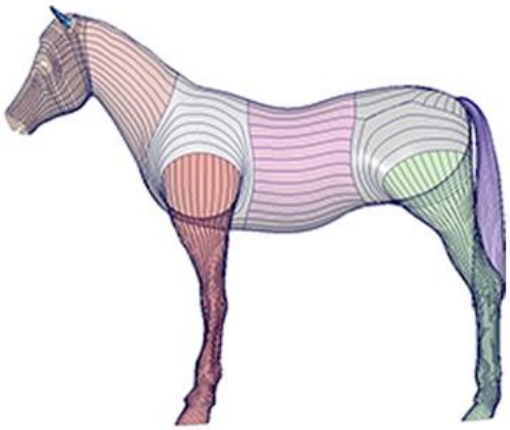
First, a polygon model is divided into different parts through manual operations or surface segmentation. In doing so, the horse model shown in Figure 4.4(a) is divided into different parts, and then highlighted with different colours in Figure 4.4(b).



a b

c





d e

Figure 4.4: Transforming a polygon horse model into a curve network

Second, a Maya Embedded Language (MEL) script is developed to obtain the vertex indices. The process involves the following steps:

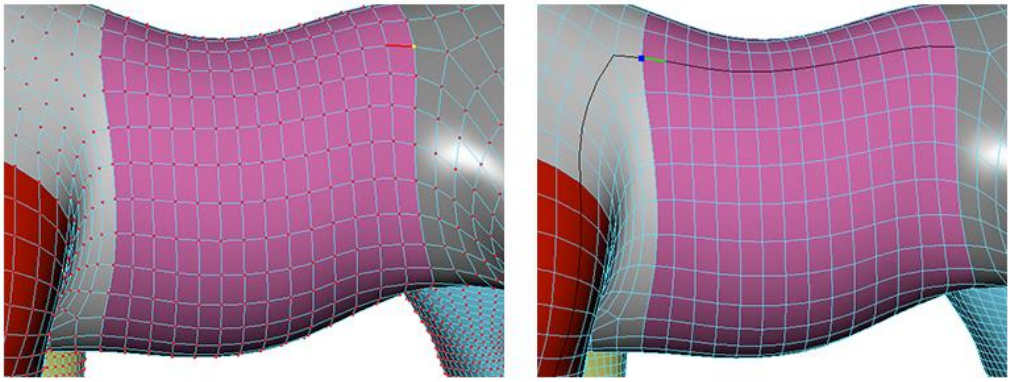
1) Manually specify a vertex in yellow, which is the starting vertex of a curve to be extracted, and an edge in red from the vertex, as shown in Figure 4.5(a)

2) Carry out the Maya Select Edge Loop MEL command to extend the curve in black, as indicated in Figure 4.5(b)

3) If the curve has passed its ending vertex in blue, select the edge in green before the ending vertex, as demonstrated in Figure 4.5(b) and use the Maya Stop Edge Loop MEL command to complete the extraction of the curve, as depicted in Figure 4.5(c)

4) If a black curve to be extracted goes the wrong direction at the vertex in pink, select the correct edge in red as in Figure 4.5(d), and conduct the Maya Change Edge Loop MEL command to change the direction of the curve to the correct one, shown in Figure 4.5(e)

With these steps, the vertex indices of all curves from the vertices of the horse model indicated in Figure 4.4(c) are obtained and the polygon horse model in Figure 4.4(a) is transformed into a curve network, i.e., a curve defined model as demonstrated in Figures 4.4(d) and 4.4(e). Third, after obtaining the vertex indices of all curves, the coordinate values of all curve vertices from the vertex indices are automatically extracted at different poses by using the developed Maya Plug-in.



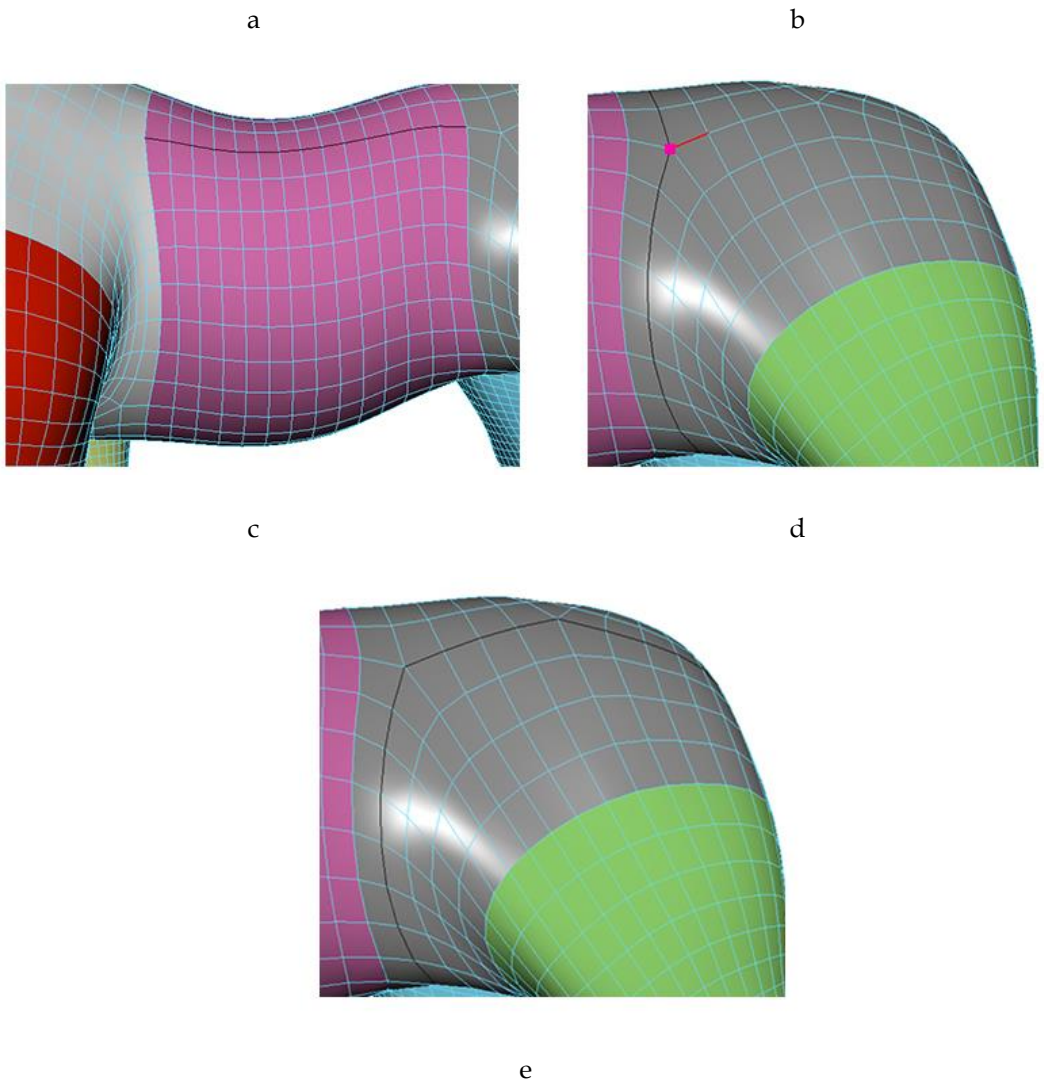
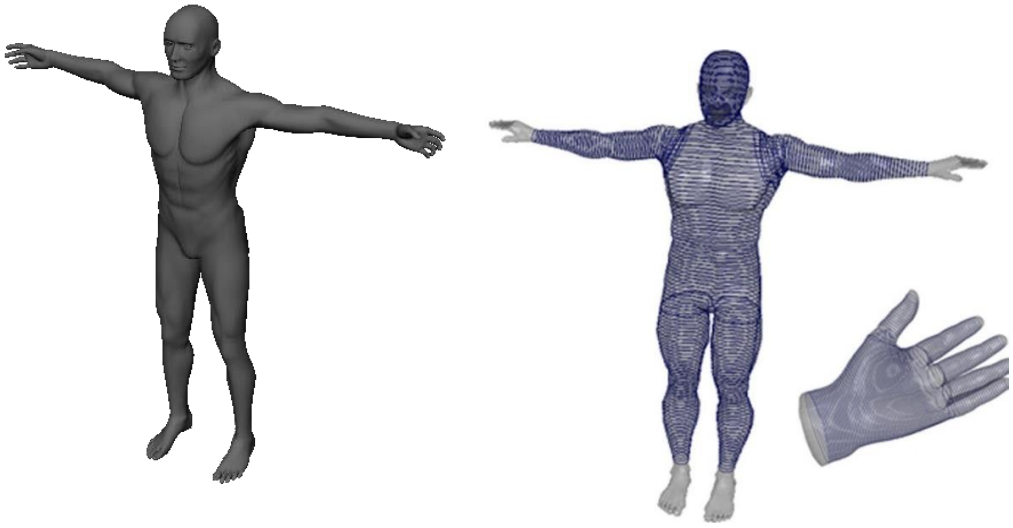


Figure 4.5: Curve extraction with MEL commands

This method is efficient. The horse model in Figure 4.4(a) has 30,134 vertices. Using the above curve creation method, the curves defining a human arm, a male model, a hand with fingers, and a horse model are obtained and depicted in Figure 4.6 and Figure 4.4.

a

b



c d

Figure 4.6: Curve defined models

4.3.2 Transferring deformations of curve defined models to polygonal models

After the curves describing a polygon model are obtained, these curves can be deformed and the deformations of these curves transferred to the polygon model to achieve the new shape of the polygon model. Through this method, a polygon model can be animated by animating the curves describing the model shown in Figure 4.7.

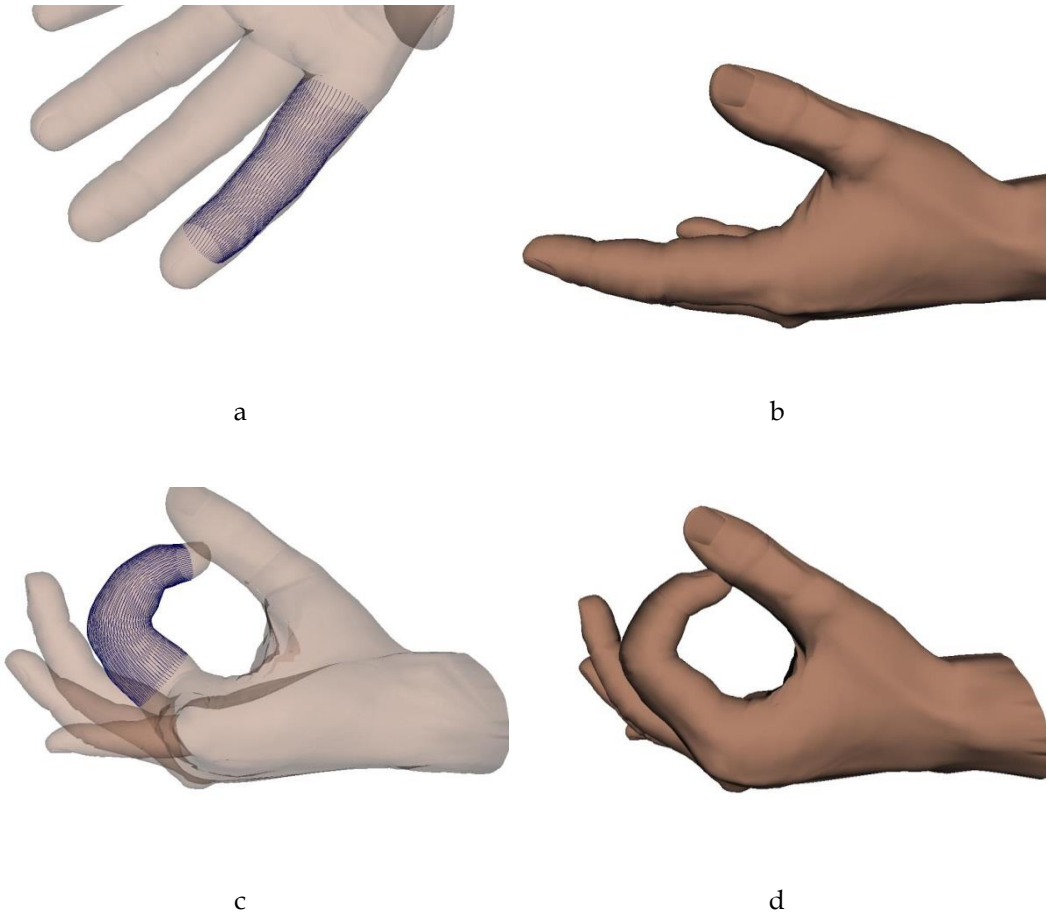


Figure 4.7: Process of transferring deformations of the curve defined model back to the polygonal model

There are two methods which can be used for this purpose, as proposed by You et al. (2009). One is to create a new model from the curves through the skinning method, and the other is to relate all surface points of the polygon model to these curves. Here, the latter is adopted. The basic idea is to find the position of a surface point on a curve or between two adjacent curves, and use this position to determine the displacement of the surface point.

Before carrying out the static or dynamic deformation calculations described in Chapter 5 and Chapter 6 below, rigid-body transformations are applied to the polygon model and extracted curves to exclude the influences caused by geometric rotations and translations.

With the static and dynamic deformation models and their finite difference solutions described in Chapter 5 and Chapter 6, the deformations of the extracted curves are calculated. These deformations are transferred back to the polygon model to achieve new skin shapes. Depending on whether the vertices of the polygon model are on the extracted curves or not, different methods will be used to determine their new positions.

If a vertex $\mathbf{V}$ of the polygon model is between the two vertices $\mathbf{p}_1$ and $\mathbf{p}_2$ of the curve, the new position $\mathbf{v}'$ of the vertex $\mathbf{V}$ after deformations can be obtained through $\overline{\mathbf{p}_1'\mathbf{v}'}/\overline{\mathbf{p}_1'\mathbf{p}_2'} = \overline{\mathbf{p}_1\mathbf{v}}/\overline{\mathbf{p}_1\mathbf{p}_2}$ where $\mathbf{V}$, $\mathbf{p}_1$, and $\mathbf{p}_2$ are positions before deformations, and $\mathbf{v}'$, $\mathbf{p}_1'$, and $\mathbf{p}_2'$ are positions after deformations.

If a vertex $\mathbf{v}$ of the polygon model is among the four vertices $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ of two adjacent extracted curves $\mathbf{c}_1$ and $\mathbf{c}_2$ where $\mathbf{p}_1$ and $\mathbf{p}_2$ are on $\mathbf{c}_1$

and $\mathbf{p}_3$ and $\mathbf{p}_4$ are on $\mathbf{c}_2$, the new position $\mathbf{v}'$ of the vertex $\mathbf{v}$ after deformations is determined below.

First, the centre $\mathbf{p}$ of the four vertices $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ is calculated to obtain $\overline{\mathbf{pv}}$. After deformations, the four vertices $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ move to new positions $\mathbf{p}_1'$, $\mathbf{p}_2'$, $\mathbf{p}_3'$ and $\mathbf{p}_4'$. The centre of these four vertices is $\mathbf{p}'$. The new position v′ of the vertex $\mathbf{v}$ after deformations is obtained by superimposing $\overline{\mathbf{pv}}$ to $\mathbf{p}'$. With the above treatment, new shapes of the polygon model after deformations can be obtained.

5 Static Curve-Based Skin Deformations

5.1 Overview

For static deformations, the mathematical model consists of a time-independent vector-valued fourth order ordinary differential equation and boundary constraints. The purpose of introducing a fourth order ordinary differential equation to describe the deformation of curves is that the deformation of a curve is similar to that of beam bending, and it considers the physical and geometrical properties with an externally applied force to change the shape of curves. Therefore, it can produce more realistic deformation of curves than purely geometric shape manipulation. Once the force fields acting on the curves defining a skin surface have obtained, physics-based approaches can be introduced to determine the deformations of these curves.

Ordinary differential equations describe a function or functions of one independent variable and its derivatives. Generally for the numerical solution of Ordinary differential equations, Runge-Kutta formulas and linear multistep formulas (Hairer et al., 2009; Hairer and Wanner, et al., 2010) are used. Ordinary differential equations also requires boundary constraints. Soetaert et al., (2010) implemented three methods to solve boundary value problems.

Finite difference method solves ordinary differential equations with conditions depending on the boundary. The finite difference solution of ordinary differential equations is very efficient, and it has a powerful capacity to tackle the problem of concentrated forces or line distribution forces in a local region, which is very difficult to cope with using analytical solutions. Due to this advantage, many researchers have applied finite difference method in the various field of science such as fluid flow Anderson et al. (1984), Patankhar et al. (1980) and heat transfer problem by Jaluria .et al (1986), Özi¸sik.et al (1994).

5.2 Static Deformation Model

Similar to the equation of beam bending, the static deformations of a curve can be represented by a vector-valued static fourth order ordinary differential equation which has the following form

$$\mathbf{a}_1 \frac{d^4\mathbf{q}(v)}{dv^4} + \mathbf{a}_2 \frac{d^2\mathbf{q}(v)}{dv^2} = \mathbf{F}(v) \tag{5.1}$$

where $\mathbf{a}_1$ and $\mathbf{a}_2$ are vector-valued shape control parameters, $\mathbf{q}(v)$ is a vector-valued deformation function, and $\mathbf{F}(v)$ is a vector-valued externally applied varying force.

The boundary conditions of beam bending consist of the positions and rotations at the two ends of curves. Similarly, the boundary conditions for the curve deformation can be represented by the following Equation (5.2).

$$
\begin{aligned}
&v = 0, \quad \mathbf{q} = \bar{\mathbf{q}}, \quad \frac{d\mathbf{q}}{dv} = 0 \\
&v = 1, \quad \mathbf{q} = \tilde{\mathbf{q}}, \quad \frac{d\mathbf{q}}{dv} = 0
\end{aligned}
\tag{5.2}
$$

where $v = 0$ and $v = 1$ indicate the two ends of a curve.

Equations (5.1) and (5.2) define the mathematical model of deformations of the curves describing a skin surface.

5.3 Solution of Mathematical Model

According to the central finite difference approximation, the finite difference formula (5.3) of the first derivative, (5.4) of the second derivative, and (5.5) of the fourth derivative can be found at a representative node ***i*** as shown in Figure 5.1.

$$
(\frac{d\mathbf{q}}{dv})_i = \frac{1}{2h}[\mathbf{q}_{i+1} - \mathbf{q}_{i-1}] \tag{5.3}
$$

$$
(\frac{d^2\mathbf{q}}{dv^2})_i = \frac{1}{h^2}[\mathbf{q}_{i+1} - 2\mathbf{q}_i + \mathbf{q}_{i-1}] \tag{5.4}
$$

$$
(\frac{d^4\mathbf{q}}{dv^4})_i = \frac{1}{h^4}[6\mathbf{q}_i - 4(\mathbf{q}_{i-1} + \mathbf{q}_{i+1}) + \mathbf{q}_{i-2} + \mathbf{q}_{i+2}] \tag{5.5}
$$

where h is the interval between two adjacent nodes. Substituting equations (5.4) and (5.5) into equation (5.1), the finite difference equation of ODE (5.1) at the representative node *i* is obtained, which is given in equation (5.6).

$$\frac{\mathbf{a}_1}{h^4}[6\mathbf{q}_i - 4(\mathbf{q}_{i-1} + \mathbf{q}_{i+1}) + \mathbf{q}_{i-2} + \mathbf{q}_{i+2}] + \frac{\mathbf{a}_2}{h^2}[\mathbf{q}_{i+1} - 2\mathbf{q}_i + \mathbf{q}_{i-1}] = \mathbf{F}_i \tag{5.6}$$

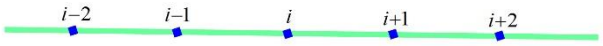


Figure 5.1 : Representative node i

Each curve has been uniformly divided into N equal intervals as indicated in Figure 5.2. In the figure, nodes 0 and N are boundary nodes, -1 and N+1 are two virtual nodes used to indicate the nodes outside the boundary nodes 0 and N, and nodes 1, 2,…, N-1 are inner nodes. Using equation (5.6), the finite difference equations for all the inner nodes can be obtained. Put the finite difference equations for all inner nodes together, N-1 linear algebra equations are obtained, which can be used to determine N-1 unknown constants 1,2,…, N-1.

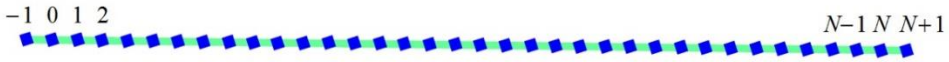


Figure 5.2 : Nodes of a curve

For inner nodes 2 and N-2, their finite difference equations involve boundary nodes 0 and N. For inner nodes 1 and N-1, their finite difference equations involve both boundary nodes 0 and N and virtual nodes -1 and N+1. Therefore, four more equations are required to determine the unknown constants at nodes -1, 0, N and N+1. This can be done by considering boundary conditions

(5.2) and the finite difference formula (5.3) whose finite difference equations are as follows

$$\begin{aligned}
&\mathbf{q}_0 = \bar{\mathbf{q}} \\
&\mathbf{q}_1 - \mathbf{q}_{-1} = 0 \\
&\mathbf{q}_N = \tilde{\mathbf{q}} \\
&\mathbf{q}_{N+1} - \mathbf{q}_{N-1} = 0
\end{aligned} \tag{5.7}$$

Assembling the finite difference equations (5.6) for all inner nodes and equation (5.7) together, and writing them in a matrix form, results in

$$\mathbf{KX} = \mathbf{F} \tag{5.8}$$

The solution of Equation (5.8) is obtained by using the inverse of matrix K to multiply both sides of equation (5.8).

$$\mathbf{X} = \mathbf{K}^{-1}\mathbf{F} \tag{5.9}$$

For the initial pose 0 as shown in Figure 5.3, the force at the pose can be determined, which is

$$\begin{aligned}
&\mathbf{F}_{i,0} = \frac{\mathbf{a}_1}{h^4}[6\mathbf{q}_{i,0} - 4(\mathbf{q}_{i-1,0} + \mathbf{q}_{i+1,0}) + \mathbf{q}_{i-2,0} + \mathbf{q}_{i+2,0}] + \frac{\mathbf{a}_2}{h^2}[\mathbf{q}_{i+1,0} - 2\mathbf{q}_{i,0} + \mathbf{q}_{i-1,0}] \\
&(i = 1, 2, 3\cdots, N-1)
\end{aligned} \tag{5.10}$$

where the second subscript 0 indicates the initial pose, and the first subscript indicates the node index.

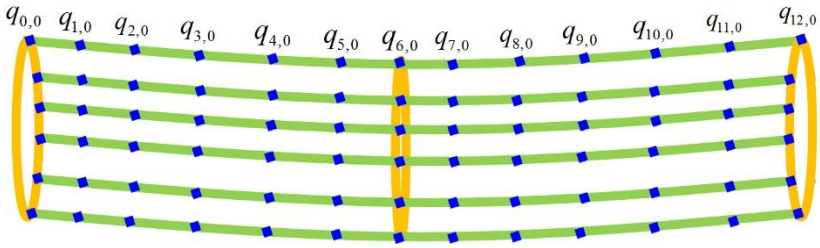


Figure 5.3 : Initial pose 0

For the final pose J as shown in Figure 5.4, the force at the pose can also be determined, which is

$$\mathbf{F}_{i,J} = \frac{\mathbf{a}_1}{h^4}[6\mathbf{q}_{i,J} - 4(\mathbf{q}_{i-1,J} + \mathbf{q}_{i+1,J}) + \mathbf{q}_{i-2,J} + \mathbf{q}_{i+2,J}] + \frac{\mathbf{a}_2}{h^2}[\mathbf{q}_{i+1,J} - 2\mathbf{q}_{i,J} + \mathbf{q}_{i-1,J}]$$
$$(i = 1, 2, 3\cdots, N-1) \tag{5.11}$$

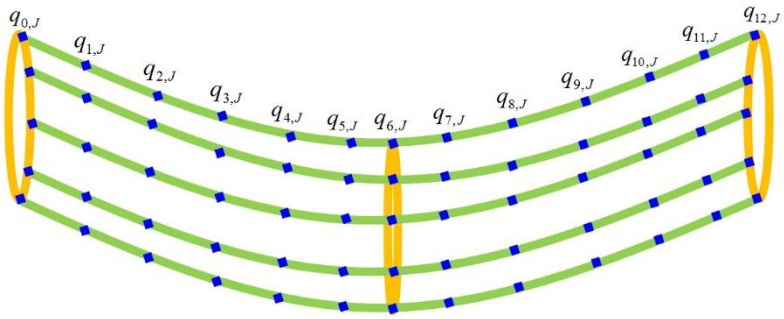


Figure 5.4: Final pose J

Then, the difference of the forces between the initial and final poses can be calculated, using the following equation.

$$\Delta \mathbf{F}_i = \mathbf{F}_{i,J} - \mathbf{F}_{i,0} \quad (i = 1, 2, 3 \cdots, N-1) \tag{5.12}$$

Having known the force difference, the deformation at any pose between the initial and final poses can be calculated. For example, to calculate the deformation at the pose j (0<j<J), the following equation can first be generated

$$\frac{\mathbf{a}_1}{h^4}[6\mathbf{q}_{i,j} - 4(\mathbf{q}_{i-1,j} + \mathbf{q}_{i+1,j}) + \mathbf{q}_{i-2,j} + \mathbf{q}_{i+2,j}] + \frac{\mathbf{a}_2}{h^2}[\mathbf{q}_{i+1,j} - 2\mathbf{q}_{i,j} + \mathbf{q}_{i-1,j}] = \mathbf{F}_{i,j} \quad (i = 1, 2, 3 \cdots, N-1;\ j = 1, 2, \cdots, J-1) \tag{5.13}$$

where

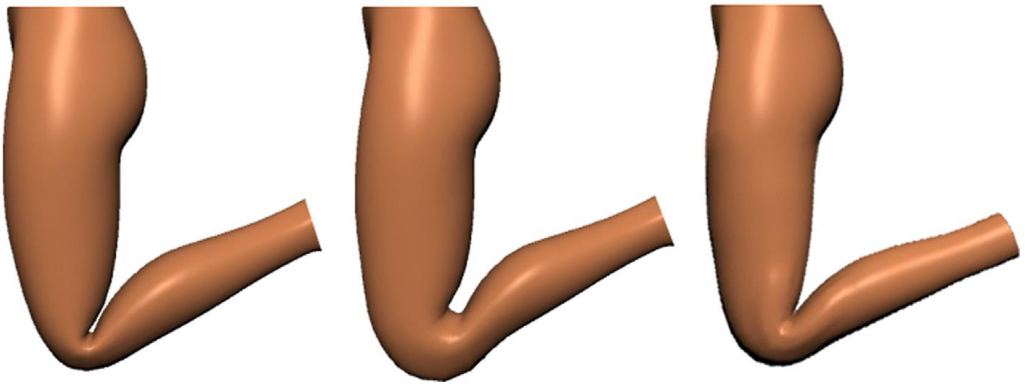
$$\mathbf{F}_{i,j} = \frac{j \times \Delta \mathbf{F}_i}{J} \quad (i = 1, 2, 3 \cdots, N-1;\ j = 1, 2, \cdots, J-1) \tag{5.14}$$

Substituting equation (5.14) into equation (5.13), producing equation (5.8) using equation (5.13) and boundary conditions (5.7), and solving equation (5.8), the curve deformation given by equation (5.9) at the pose j is obtained, which can be used to create the skin deformation at the pose.

5.4 Experimental Comparison

This section describes a comparison of the above approach with two skinning methods used in Autodesk Maya. Figure 5.4(a) and (b) are from the classic linear skinning method and dual quaternion skinning method, respectively, and Figure 5.4(c) is from the proposed approach. Clearly, classic linear skinning method without using the painting tool to apply the weights causes the problem of a collapsing joint, and dual quaternion skinning method solves the problem but still creates less realistic skin deformations at the joint. In contrast, the proposed approach creates more realistic skin deformation of the human leg.

Classic linear skinning method	Dual quaternion skinning method	Proposed method



a b c

Figure 5.5: Comparison between the proposed skin deformation method and other methods (a)This method creates less realistic skin deformations at the joint (b)This method causes the skin surface at the joints to increase (c) Human leg animated by the proposed method

Table 5.1: Computation time of five models

Application Examples	No of Curves	No of Nodes	Time (sec) to get K^{-1}	Time (sec) to get solution for 30 frame	Total Time (sec) for 30 frames
Hand finger	64	22	0.00028	0.00898	0.00926
Human Leg	36	31	0.00066	0.01680	0.01746
Human arm 1	34	27	0.00045	0.01305	0.01350

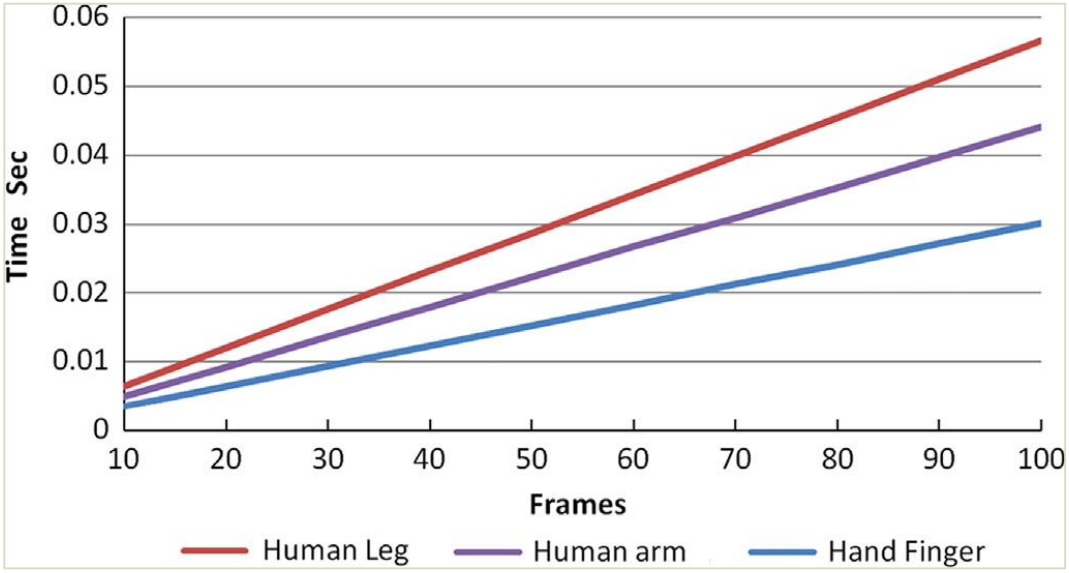


Figure 5.6: Relationship between total time and frames

The timing test demonstrates the increased efficiency of the proposed approach effectively. The test was carried out on a 3GHz Intel Core2 Do PC with 2.98 GB of memory. The Gauss–Jordan elimination was used to find the inverse K^{-1} of the matrix K. For each of the four models shown in Figures (5.7-5.10), different numbers of surface curves and different numbers of vertices were used

to demonstrate the efficiency of the proposed method. The results are given in Table 5.1. In the table, the second column indicates the number of the curves used to define each of the four models. The third column denotes the number of nodes of each curve. The fourth column gives the time taken to find the inverse K^{-1} of the matrix K. The fifth column is the time spent on Equation (5.9) for 30 frames and the sixth column is the total time taken to create 30 frames for each of 4 models. How the total time changes with the increase of the frames for the first three models is plotted in Figure 5.6. It can be seen from the table that when the order of the matrix is small, say for example 27 by 27, the time taken to find the inverse K^{-1} of the matrix K is very small (0.00045s for a matrix of 27 by 27). However, when the order becomes large enough, say for example 103 by 103, the time taken to find the inverse K^{-1} of the matrix K increases to 0.00204s. Figure (5.6) indicates that the total time increases linearly with the increase of the frames for all the models. However, the rate of the increase for different models differs largely. The models which require more time to create one frame increase more quickly than those requiring less time to create one frame.

5.5 Application Examples

This section describes several examples of applications of the proposed approach.

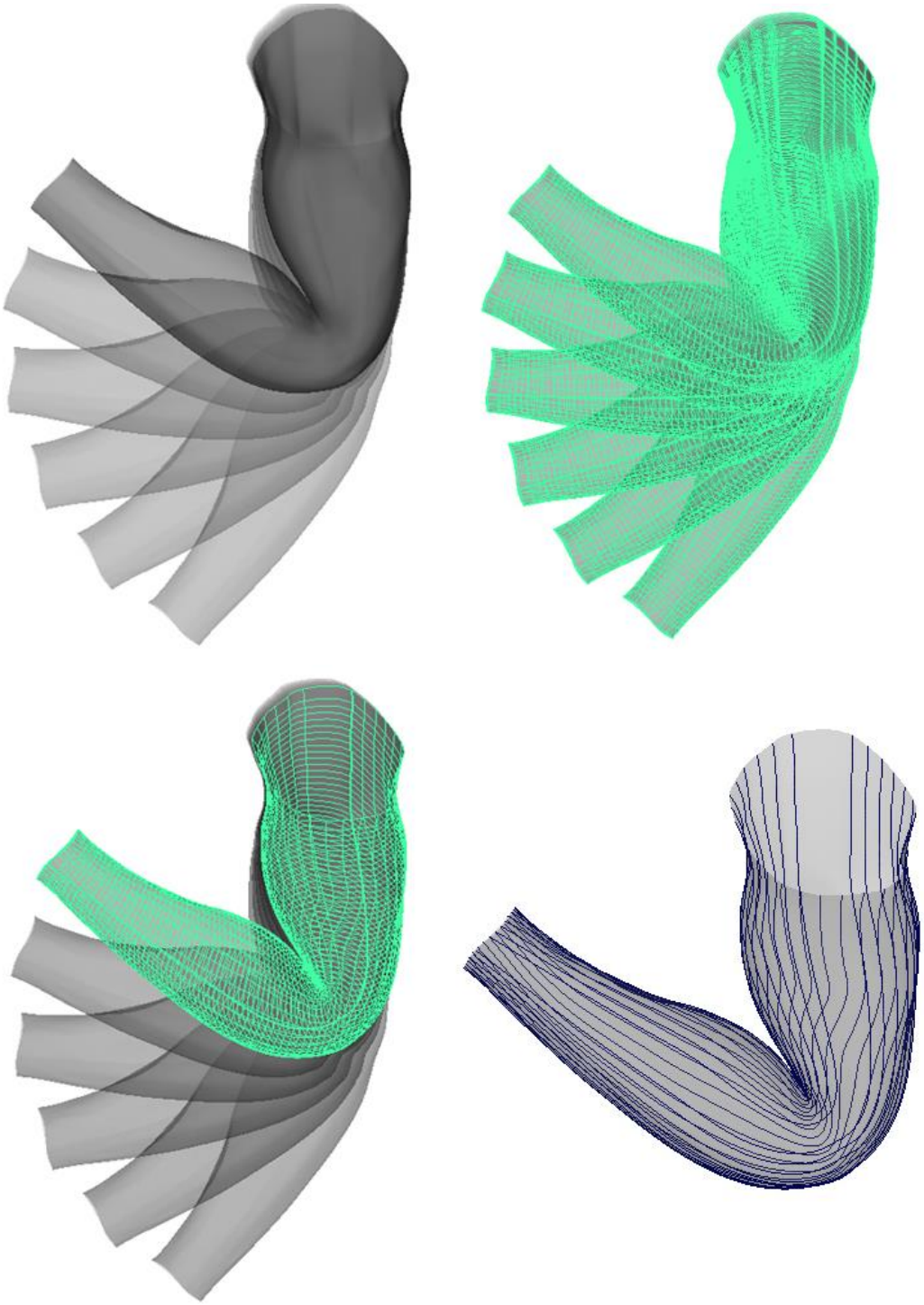
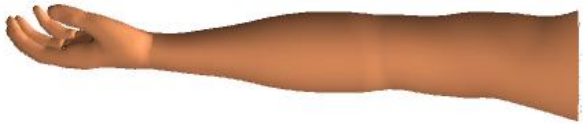


Figure 5.7: Human arm animation

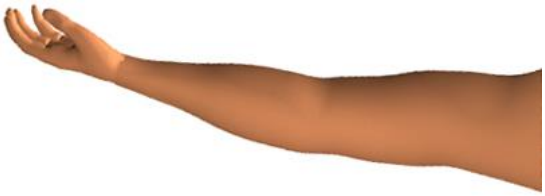
In the first example, the proposed approach has been used to animate two models of a human arm, as shown in Figure 5.7 and Figure 5.8. The arm model used in Figure 5.7 consists of 31 curves and each curve has 103 nodes. The arm model used in Figure 5.8 consists of 34 curves and each curve has 27 nodes. Fig. 5.8 demonstrates how to determine skin deformations. The skin shapes at the initial and final poses are given in Figures. 5.8(a) and 5.8(f), respectively. The skin shapes at the poses between the initial and final poses are depicted in Figures. 5.8(b), 5.8(c), 5.8(d) and 5.8(e). These images indicate that the proposed approach creates plausible skin shapes.

The second example is to animate skin deformations of a hand with the proposed approach. The first and the final images shown in Figure 5.9 are from the initial and final poses, respectively, and those between the first and final poses are from the calculated results. These images also demonstrate the effectiveness of the proposed approach in animating skin deformations.

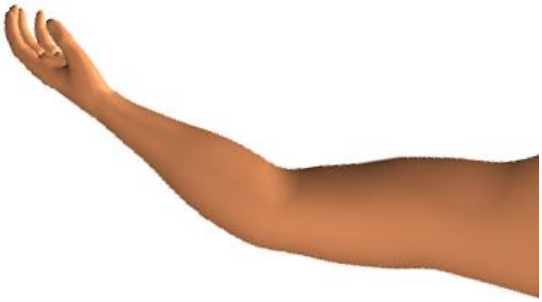
.



(a) Initial pose 0



(b) Pose 1



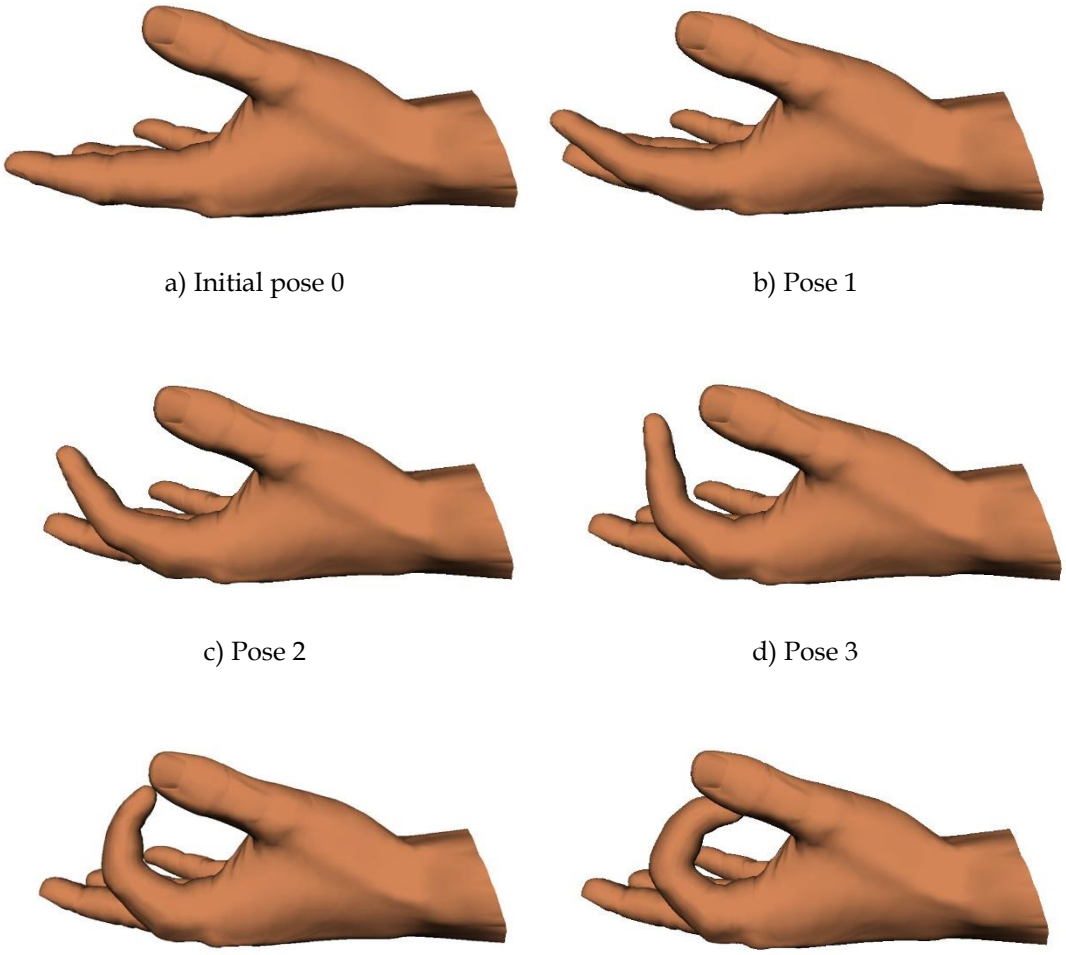
(c) Pose 2

(d) Pose 3

(e) Pose 4

(f) Pose 5

Figure 5.8 : Skin deformation of an arm



a) Initial pose 0 b) Pose 1

c) Pose 2 d) Pose 3

e) Pose 4 f) Final pose J= 5

Figure 5.9 : Skin deformation of a hand

In the third example, a human leg is animated using the proposed approach. The first and final images shown in Figure 5.10 are from the initial and final poses respectively, and those between the first and final poses are from the calculated results. It can be observed that the proposed approach generates a realistic skin deformation of the human leg.

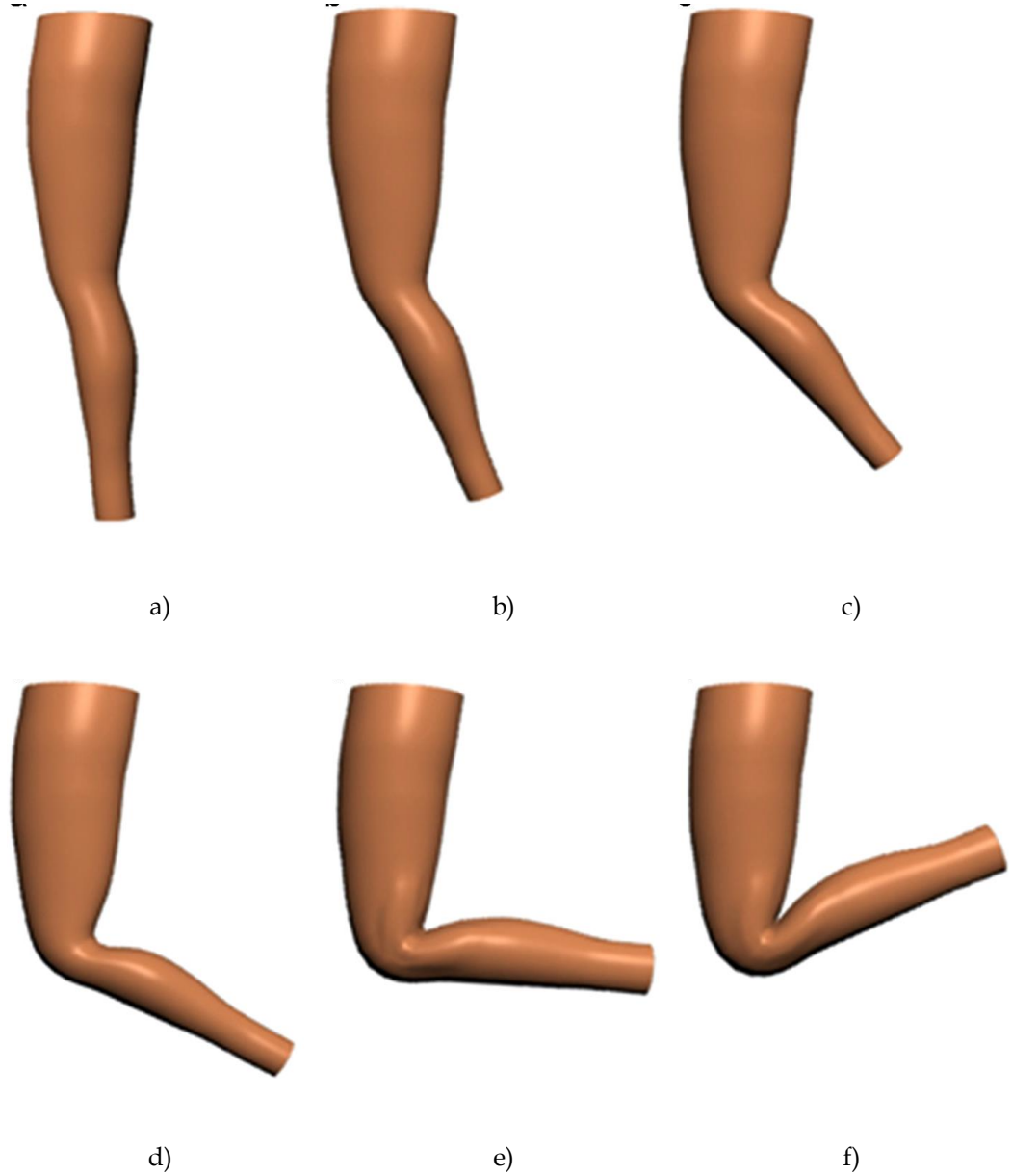


Figure 5.10: Skin deformation of a leg

5.6 Summary

This chapter outlines a new approach to curve based skin deformation which is based on a mathematical model and a fast finite difference solution of static skin deformation. This approach is used to animate the skin deformation of human arms, fingers and legs. A comparison of two important skinning methods and the

proposed one has been made in this chapter. The examples demonstrate that the proposed approach can create realistic skin deformation of character models. The computational efficiency of the proposed approach demonstrates that the method is very efficient and can be used to create skin deformation for real time character animation.

6 Dynamic Curve-Based Skin Deformations

6.1 Overview

For dynamic deformations, dynamic effects are introduced which consider how velocity, acceleration and material properties of a moving skin surface affect its shape. The ordinary differential equation dealing with dynamic deformations is time-dependent and more difficult to solve.

The parametric representation of skin surfaces involves two parametric variables. When physics-based methods are introduced to determine shape changes of skin surfaces, complicated partial differential equations are involved which have to be solved by inefficient numerical methods. In contrast, the parametric representation of curves only involves one parametric variable. When physics-based methods are used to determine shape changes of curves, simple ordinary differential equations are formulated which can be solved efficiently. Due to this advantage, the existing work has developed the finite difference solution of static skin deformations. Since skin deformations are caused by skin surface movements, more realistic skin deformations should consider dynamic effects caused by skin surface movements.

Dynamic skin deformations are determined by dynamic shape changes of the curves defining skin surfaces. Dynamic deformation of curves is similar to dynamic bending of homogeneous elastic beams. Therefore, the dynamic

deformation model of curves is developed. The sculpting forces control deformations of curves.

Then, transforming the dynamic deformation model of curves into the finite difference equations is investigated. These finite difference equations can be used to tackle two different issues described below.

When the sculpting forces are known, they are substituted into the finite difference equations, and the equations are solved to obtain curve deformations which are transferred to skin surfaces to achieve dynamic skin deformations.

When the sculpting forces are unknown, but example skin shapes at two different poses are known, an iterative method is proposed below to determine the unknown sculpting forces from the finite difference equations. The obtained sculpting forces are introduced back into the finite difference equations, and the equations are solved to obtain dynamic shape changes of curves which are used to determine dynamic skin deformations.

6.2 Dynamic Deformation Model

The dynamic bending equation of homogeneous elastic beams can be written as (Coşkun et al. 2003)

$$EI\frac{\partial^4 w}{\partial x^4}+\rho A\frac{\partial^2 w}{\partial t^2}=F(x,t) \tag{6.1}$$

where E is Young's modulus, I is the second moment, ρ is the mass density, A is the cross-section area of the beam, $F(x)$ is a lateral load acting on the beam,

w is the lateral deformation, x is a position variable in the beam direction, and t is the time variable.

The above equation has already been widely applied in engineering to achieve accurate predictions of dynamic beam bending. Curve deformations following the same physics should give a more realistic result.

For the parametric representation of a curve, there are three position components x, y and z. For dynamic deformations, each of them is the function of the parametric variable v and the time variable t. Accordingly, the function w in the above equation should be replaced by the three position components x, y and z, respectively, and the variable x in Equation (6.1) should be replaced by the parametric variable v. Using the symbol $\mathbf{q}$ to represent a vector-valued position function which has three position components x, y and z ($\mathbf{q} = \begin{bmatrix} x & y & z \end{bmatrix}^T$), and a_1 and a_2 to represent EI and ρA, Equation (6.1) is changed into the following differential equation

$$a_1 \frac{\partial^4 \mathbf{q}(v,t)}{\partial v^4} + a_2 \frac{\partial^2 \mathbf{q}(v,t)}{\partial t^2} = \mathbf{f}(v,t) \tag{6.2}$$

Since a_1 and a_2 have a big impact on shape changes of deformed curves, they are called shape control parameters. The lateral load $\mathbf{f}(v,t) = \begin{bmatrix} f_x(v,t) & f_y(v,t) & f_z(v,t) \end{bmatrix}^T$ is called a vector-valued sculpting force.

Equation (6.2) governs the physics-based deformations of a curve. It does not include any rigid-body transformations. These rigid-body transformations must be removed first before Equation (6.2) is applied.

In order to ensure two connected curves always maintain position and tangential continuities during the deformations described by Equation (6.2), boundary conditions must be applied when solving Equation (6.2). These boundary conditions are determined below.

Position continuity requires that two connected curves have the same deformation, and tangential continuity requires them to have the same deformation rate at their joint position. For a curve, its joint positions are its two end points which are at $v=0$ and $v=1$, respectively. According to Equation (6.2), the deformation is $\mathbf{q}(v,t)$ and the deformation rate in the length direction of a curve is $\partial\mathbf{q}(v,t)/\partial v$. If the deformation is $\mathbf{d}_0$ at $v=0$ and $\mathbf{d}_1$ at $v=1$, and the deformation rate is $\mathbf{s}_0$ at $v=0$ and $\mathbf{s}_1$ at $v=1$, at the final pose $t=1$ of a movement, the boundary conditions can be written as

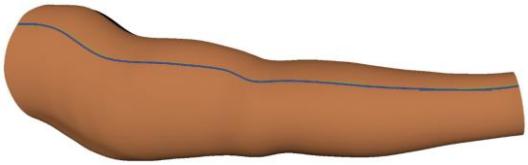
$$\begin{aligned} t=1 \quad & \mathbf{q}(0,1)=\mathbf{d}_0 \qquad \left.\frac{\partial\mathbf{q}(v,1)}{\partial v}\right|_{v=0}=\mathbf{s}_0 \\ & \mathbf{q}(1,1)=\mathbf{d}_1 \qquad \left.\frac{\partial\mathbf{q}(v,1)}{\partial v}\right|_{v=1}=\mathbf{s}_1 \end{aligned} \tag{6.3}$$

The deformations $\mathbf{d}_0$ and $\mathbf{d}_1$ and the deformation rates $\mathbf{s}_0$ and $\mathbf{s}_1$ are determined below.

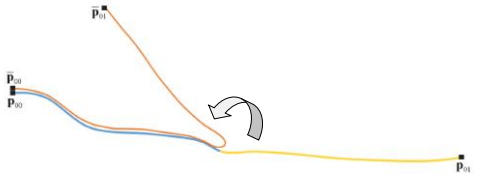
When a 3D skin surface is deformed into a new one, it usually involves rigid-body transformations such as translations and rotations. The rigid-body transformations can be removed by rotating the curve at the initial pose to the final pose as shown in Figure 6.1(a),(b) and calculating the deformation differences between the skin shape at the final pose and the curve at the initial

pose after the rotation as shown in Figure 6.1(e). This can be easily achieved by applying the operations of translation and rotation transformations. The deformation differences obtained are used to determine the sculpting forces. By introducing sculpting forces into the analytical solution obtained, the deformations of the representative curves at any pose are determined.

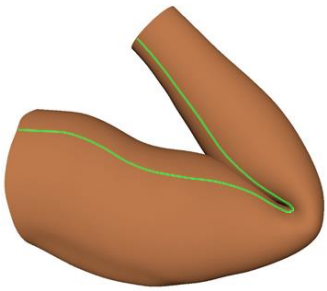
Assuming the position value of a curve at the initial pose is $\mathbf{p}_{00}$ at $v=0$ and $\mathbf{p}_{01}$ at $v=1$ as shown in Figure 6.1(b) the half section of curve highlighted in yellow is rotated with the linear rotation. The position value of the curve at the final pose is $\mathbf{p}_{10}$ at $v=0$ and $\mathbf{p}_{11}$ at $v=1$ as shown in Figure 6.1(d), the deformation after excluding rigid-body transformations is $\mathbf{d}_0=\mathbf{p}_{10}-\bar{\mathbf{p}}_{00}$ at $v=0$ and $\mathbf{d}_1=\mathbf{p}_{11}-\bar{\mathbf{p}}_{01}$ at $v=1$ as shown in Figure 6.1(e), where $\bar{\mathbf{p}}_{00}$ and $\bar{\mathbf{p}}_{01}$ are respectively the position value of $\mathbf{p}_{00}$ and $\mathbf{p}_{01}$ after rigid-body transformations. With the same method, the deformation $\bar{\mathbf{d}}_0$ of the node can be found next to the node at $v=0$ which is $\bar{\mathbf{d}}_0=q_{10}-\bar{q}_{00}$ and the deformation $\bar{\mathbf{d}}_1$ of the node next to the node at $v=1$ is $\bar{\mathbf{d}}_1=q_{11}-\bar{q}_{01}$. The deformation rates of the curve at $v=0$ and $v=1$ are determined by $\mathbf{s}_0=\left(\bar{\mathbf{d}}_0-\mathbf{d}_0\right)/h_0$ and $\mathbf{s}_1=\left(\mathbf{d}_1-\bar{\mathbf{d}}_1\right)/h_1$, respectively, where h_0 is the length between two adjacent nodes at $v=0$ and h_1 is the length between two adjacent nodes at $v=1$.



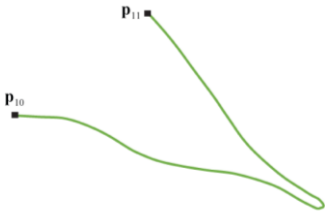
a) Curve at the initial pose



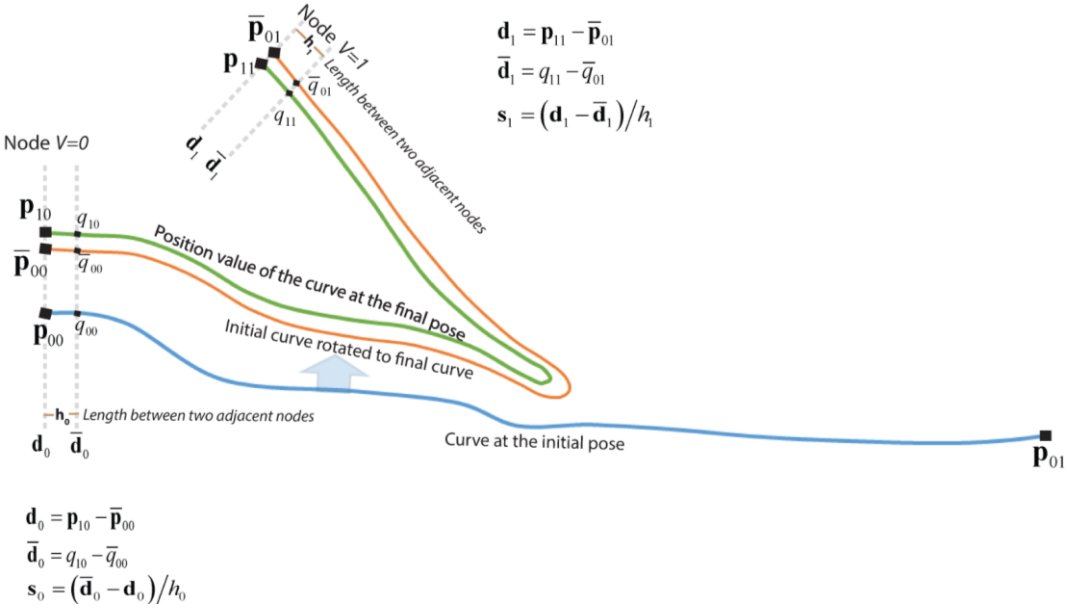
b) Curve at Initial pose rotated to the final pose



c) Skin shape at the final pose and curve at final pose



d) Curve at final pose



e) Deformation differences between the skin shape at the final pose and the curve at the initial pose after the rotation

Figure 6.1: Deformation of curve excluding rigid-body transformation

At the initial pose $t=0$, the deformations and deformation rates are zero, and the following initial conditions can be reached

$$t=0 \qquad \mathbf{q}(v,0)=0 \qquad \left.\frac{\partial \mathbf{q}(v,t)}{\partial t}\right|_{t=0}=0 \tag{6.4}$$

Equation (6.4) means that the deformations and deformation rates at the two ends of a curve in the length direction at the initial pose are zero, i.e.,

$$\begin{aligned} &v=0 \qquad \mathbf{q}(0,0)=0 \qquad \left.\frac{\partial \mathbf{q}(v,0)}{\partial v}\right|_{v=0}=0 \\ &v=1 \qquad \mathbf{q}(1,0)=0 \qquad \left.\frac{\partial \mathbf{q}(v,0)}{\partial v}\right|_{v=1}=0 \end{aligned} \tag{6.5}$$

Using a linear relationship to describe how boundary conditions change against time, the following boundary conditions at any time instant *t* are reached

$$\begin{aligned} &v=0 \qquad \mathbf{q}(0,t)=t\mathbf{d}_0 \qquad \left.\frac{\partial \mathbf{q}(v,t)}{\partial v}\right|_{v=0}=t\mathbf{s}_o \\ &v=1 \qquad \mathbf{q}(1,t)=t\mathbf{d}_1 \qquad \left.\frac{\partial \mathbf{q}(v,t)}{\partial v}\right|_{v=1}=t\mathbf{s}_1 \end{aligned} \tag{6.6}$$

After the above treatment, determination of dynamic curve deformation is transformed into a solution involving the dynamic mathematical model of curves consisting of Equation (6.2) and boundary conditions (6.6).

6.3 Finite Difference Equations

As demonstrated in Table 6.1 of Section 6.6, the implicit finite difference solution of the dynamic deformation model is very efficient.

In order to develop such a finite difference solution, the above dynamic deformation model into finite difference equations is transformed as introduced below.

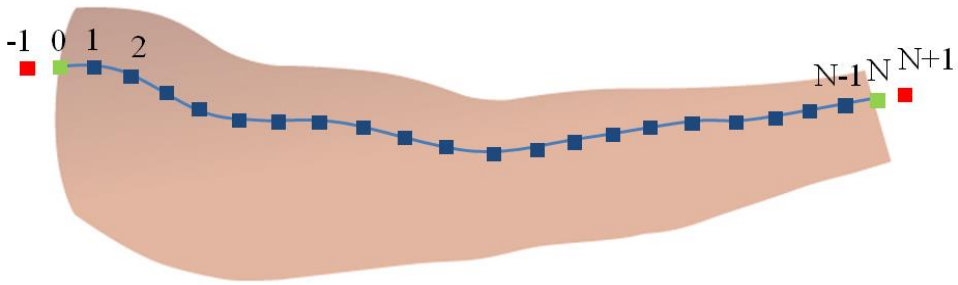


Figure 6.2: Nodes on a curve

As shown in Figure 6.2, the domain $0 \le v \le 1$ is divided into N equal intervals. The parametric value between any two adjacent nodes is $\Delta v = 1/N$, and the parametric value of the n^{th} node is $v_n = n/N$ (n= 0, 1, 2,…, N). Similarly, the time range $0 \le t \le 1$ into M is divided into equal time intervals. The time increment is $\Delta t = 1/M$, and the time variable takes the value of $t_m = m/M$ (m= 0, 1, 2,…, M) at the m^{th} time step. The derivatives of $\mathbf{q}(v,t)$ with respect to the parametric variable v and time variable t at the $(m+1)^{th}$ time step and the n^{th} node can be written as

$$
\begin{aligned}
&\left[\frac{\partial \mathbf{q}(v,t)}{\partial t}\right]_n^{m+1} = \frac{\mathbf{q}_n^{m+1} - \mathbf{q}_n^{m-1}}{2\Delta t} \\
&\left[\frac{\partial \mathbf{q}^2(v,t)}{\partial t^2}\right]_n^{m+1} = \frac{\mathbf{q}_n^{m+1} - 2\mathbf{q}_n^{m} + \mathbf{q}_n^{m-1}}{\Delta t^2} \\
&\left[\frac{\partial \mathbf{q}(v,t)}{\partial v}\right]_n^{m+1} = \frac{\mathbf{q}_{n+1}^{m+1} - \mathbf{q}_{n-1}^{m+1}}{2\Delta v} \\
&\left[\frac{\partial^4 \mathbf{q}(v,t)}{\partial v^4}\right]_n^{m+1} = \frac{1}{\Delta v^4}\left[6\mathbf{q}_n^{m+1} - 4\mathbf{q}_{n-1}^{m+1} + \mathbf{q}_{n+1}^{m+1}) + \mathbf{q}_{n-2}^{m+1} + \mathbf{q}_{n+2}^{m+1}\right]
\end{aligned}
\tag{6.7}
$$

Substituting $\Delta t = 1/M$ and $\Delta v = 1/N$ into the above equations, the following equations are obtained

$$
\begin{aligned}
&\left[\frac{\partial \mathbf{q}(v,t)}{\partial t}\right]_n^{m+1} = \frac{M}{2}\left[\mathbf{q}_n^{m+1} - \mathbf{q}_n^{m-1}\right] \\
&\left[\frac{\partial \mathbf{q}^2(v,t)}{\partial t^2}\right]_n^{m+1} = M^2(\mathbf{q}_n^{m+1} - 2\mathbf{q}_n^{m} + \mathbf{q}_n^{m-1}) \\
&\left[\frac{\partial \mathbf{q}(v,t)}{\partial v}\right]_n^{m+1} = \frac{N}{2}(\mathbf{q}_{n+1}^{m+1} - \mathbf{q}_{n-1}^{m+1}) \\
&\left[\frac{\partial^4 \mathbf{q}(v,t)}{\partial v^4}\right]_n^{m+1} = N^4\left[6\mathbf{q}_n^{m+1} - 4\mathbf{q}_{n-1}^{m+1} + \mathbf{q}_{n+1}^{m+1}) + \mathbf{q}_{n-2}^{m+1} + \mathbf{q}_{n+2}^{m+1}\right]
\end{aligned}
\tag{6.8}
$$

Substituting Equation (6.8) into Equation (6.2), the differential equation (6.2) becomes

$$
\begin{aligned}
&a_1 N^4\left[6\mathbf{q}_n^{m+1} - 4(\mathbf{q}_{n-1}^{m+1} + \mathbf{q}_{n+1}^{m+1}) + \mathbf{q}_{n-2}^{m+1} + \mathbf{q}_{n+2}^{m+1}\right] \\
&+a_2 M^2(\mathbf{q}_n^{m+1} - 2\mathbf{q}_n^{m} + \mathbf{q}_n^{m-1}) = \mathbf{f}(v_n, t_{m+1}) \\
&(m = 0,1,2,\cdots,M-1; n = 1,2,\cdots,N-1)
\end{aligned}
\tag{6.9}
$$

If $\mathbf{q}_n^{m-1}$ and $\mathbf{q}_n^m$ are known, the total unknown constants $\mathbf{q}_n^{m+1}$ in Equation (6.9) are $N+3$ since the green boundary nodes $n=0$ and $n=N$ and the red virtual nodes $n=-1$ and $n=N+1$ will be involved when formulating the finite difference equation at the blue internal nodes $n=1$ and $n=N-1$ (Figure 6.1). Since Equation (6.9) only includes $N-1$ linear algebra equations, four additional linear equations are required to determine these $N+3$ unknown constants. These four additional equations can be obtained by considering boundary conditions (6.6).

Substituting Equation (6.8) into (6.6), the following boundary conditions are obtained

$$\begin{aligned} \mathbf{q}_0^{m+1} &= \frac{m+1}{M}\mathbf{d}_0 \qquad \frac{N}{2}(\mathbf{q}_1^{m+1}-\mathbf{q}_{-1}^{m+1}) = \frac{m+1}{M}\mathbf{s}_0 \\ \mathbf{q}_N^{m+1} &= \frac{m+1}{M}\mathbf{d}_1 \qquad \frac{N}{2}(\mathbf{q}_{N+1}^{m+1}-\mathbf{q}_{N-1}^{m+1}) = \frac{m+1}{M}\mathbf{s}_1 \end{aligned} \tag{6.10}$$

Introducing Equation (6.10) into (6.9), and denoting $\mathbf{f}_n^m = \mathbf{f}(n/N, m/M)$, the unknown constants in Equation (6.9) are reduced to $N-1$ and determined by $N-1$ equations below

$$(6a_1N^4 + a_2M^2)\mathbf{q}_n^{m+1} + a_1N^4\left[-4(\mathbf{q}_{n-1}^{m+1} + \mathbf{q}_{n+1}^{m+1}) + \mathbf{q}_{n-2}^{m+1} + \mathbf{q}_{n+2}^{m+1}\right] + a_2M^2(-2\mathbf{q}_n^m + \mathbf{q}_n^{m-1}) = \mathbf{f}_n^{m+1}$$
$$(m = 0,1,2,\cdots,M-1; 2 < n < N-2)$$

$$(6a_1N^4 + a_2M^2)\mathbf{q}_1^{m+1} + a_1N^4\left[-4(\frac{m+1}{M}\mathbf{d}_0 + \mathbf{q}_2^{m+1}) + \mathbf{q}_1^{m+1} - \frac{2(m+1)}{MN}\mathbf{s}_0 + \mathbf{q}_3^{m+1}\right] + a_2M^2(-2\mathbf{q}_1^m + \mathbf{q}_1^{m-1}) = \mathbf{f}_1^{m+1}$$
$$(m = 0,1,2,\cdots,M-1; n = 1)$$

$$(6a_1N^4 + a_2M^2)\mathbf{q}_2^{m+1} + a_1N^4\left[-4(\mathbf{q}_1^{m+1} + \mathbf{q}_3^{m+1}) + \frac{m+1}{M}\mathbf{d}_0 + \mathbf{q}_4^{m+1}\right] + a_2M^2(-2\mathbf{q}_2^m + \mathbf{q}_2^{m-1}) = \mathbf{f}_2^{m+1}$$
$$(m = 0,1,2,\cdots,M-1; n = 2)$$

$$(6a_1N^4 + a_2M^2)\mathbf{q}_{N-2}^{m+1} + a_1N^4\left[-4(\mathbf{q}_{N-3}^{m+1} + \mathbf{q}_{N-1}^{m+1}) + \frac{m+1}{M}\mathbf{d}_1 + \mathbf{q}_{N-4}^{m+1}\right] + a_2M^2(-2\mathbf{q}_{N-2}^m + \mathbf{q}_{N-2}^{m-1}) = \mathbf{f}_{N-2}^{m+1}$$
$$(m = 0,1,2,\cdots,M-1; n = N-2)$$

$$(6a_1N^4 + a_2M^2)\mathbf{q}_{N-1}^{m+1} + a_1N^4\left[-4(\frac{m+1}{M}\mathbf{d}_1 + \mathbf{q}_{N-2}^{m+1}) + \mathbf{q}_{N-3}^{m+1} - \frac{2(m+1)}{MN}\mathbf{s}_1 + \mathbf{q}_{N-1}^{m+1}\right] + a_2M^2(-2\mathbf{q}_{N-1}^m + \mathbf{q}_{N-1}^{m-1}) = \mathbf{f}_{N-1}^{m+1}$$
$$(m = 0,1,2,\cdots,M-1; n = N-1) \tag{6.11}$$

Writing the above finite difference equations into a matrix form, the following equation is obtained.

$$\left[\mathbf{K}_{nm}\right]\left\{\mathbf{Q}_{nm}\right\} = \left\{\mathbf{F}_{nm}\right\} \tag{6.12}$$

where the column vector $\left\{\mathbf{Q}_{nm}\right\}$ consists of the unknown constants $\mathbf{q}_n^{m+1}$, the column vector $\left\{\mathbf{F}_{nm}\right\}$ is from the contributions of the sculpting forces $\mathbf{f}_n^{m+1}$ and

the previous skin deformations $\mathbf{q}_n^{m-1}$ and $\mathbf{q}_n^m$, and the square matrix $[\mathbf{K}_{nm}]$ consists of the coefficients of the unknown constants $\mathbf{q}_n^{m+1}$ in equation (6.11).

Since the iterative calculations start from $t=0$, $m=0$ and $\mathbf{q}_n^{m-1}$ and $\mathbf{q}_n^m$ become $\mathbf{q}_n^{-1}$ and $\mathbf{q}_n^0$. They are determined through Equation (6.12) obtained by substituting Equation (6.8) into Equation (6.4)

$$\mathbf{q}_n^0 = 0 \qquad \mathbf{q}_n^{-1} = \mathbf{q}_n^1 \qquad (n = 0,1,2,\cdots,N) \tag{6.13}$$

Considering Equation (6.12), Equation (6.11) becomes

$$(6a_1N^4 + 2a_2M^2)\mathbf{q}_n^1 + a_1N^4\left[-4(\mathbf{q}_{n-1}^1 + \mathbf{q}_{n+1}^1) + \mathbf{q}_{n-2}^1 + \mathbf{q}_{n+2}^1\right] = \mathbf{f}_n^1$$
$$(m=0; 2 < n < N-2)$$

$$(6a_1N^4 + 2a_2M^2)\mathbf{q}_1^1 + a_1N^4\left[-4(\frac{1}{M}\mathbf{d}_0 + \mathbf{q}_2^1) + \mathbf{q}_1^1 - \frac{2}{MN}\mathbf{s}_0 + \mathbf{q}_3^1\right] = \mathbf{f}_1^1$$
$$(m=0; n=1)$$

$$(6a_1N^4 + 2a_2M^2)\mathbf{q}_2^1 + a_1N^4\left[-4(\mathbf{q}_1^1 + \mathbf{q}_3^1) + \frac{1}{M}\mathbf{d}_0 + \mathbf{q}_4^1\right] = \mathbf{f}_2^1$$
$$(m=0; n=2)$$

$$(6a_1N^4 + 2a_2M^2)\mathbf{q}_{N-2}^1 + a_1N^4\left[-4(\mathbf{q}_{N-3}^1 + \mathbf{q}_{N-1}^1) + \frac{1}{M}\mathbf{d}_1 + \mathbf{q}_{N-4}^1\right] = \mathbf{f}_{N-2}^1$$
$$(m=0; n=N-2)$$

$$(6a_1N^4 + 2a_2M^2)\mathbf{q}_{N-1}^1 + a_1N^4\left[-4(\frac{1}{M}\mathbf{d}_1 + \mathbf{q}_{N-2}^1) + \mathbf{q}_{N-3}^1 + \frac{2}{MN}\mathbf{s}_1 + \right. \tag{6.14}$$
$$\left.\mathbf{q}_{N-1}^1\right] = \mathbf{f}_{N-1}^1$$
$$(m=0; n=N-1)$$

Similarly, the above finite difference equations can be written as a matrix equation below

$$\left[\mathbf{K}_{n0}\right]\left\{\mathbf{Q}_{n0}\right\}=\left\{\mathbf{F}_{n0}\right\} \tag{6.15}$$

where the column vector $\left\{\mathbf{Q}_{n0}\right\}$ consists of the unknown constants $\mathbf{q}_n^1$, the column vector $\left\{\mathbf{F}_{n0}\right\}$ is from the contributions of the sculpting forces $\mathbf{f}_n^1$, $\mathbf{d}_0$, $\mathbf{d}_1$, $\mathbf{s}_0$ and $\mathbf{s}_1$, and the square matrix $\left[\mathbf{K}_{n0}\right]$ consists of the coefficients of the unknown constants $\mathbf{q}_n^1$ in Equation (6.14).

The finite difference Equation (6.15) are a special form of the finite difference Equation (6.12) at $t=0$, i. e., m=0.

6.4 Solution of Finite Difference Equations

Depending on whether the sculpting forces are known or not, the finite difference Equations (6.12) and (6.15) can be solved with two different algorithms below.

When the sculpting forces $\mathbf{f}_n^m=\mathbf{f}(n/N,m/M)$ $(m=1,2,\cdots,M-1;$ $n=1,2,3,\cdots,N-1)$ are known, they can be substituted into Equation (6.8) and the equations solved to obtain all the unknown constants $\mathbf{q}_n^1$ $(n=1,2,3,\cdots,N-1$).

Next, m is set to 1 in Equation (6.12). Since $\mathbf{q}_n^{m-1}=\mathbf{q}_n^0$ and $\mathbf{q}_n^m=\mathbf{q}_n^1$ are known, Equation (6.12) can be used to determine all the unknown constants $\mathbf{q}_n^2$ (

$n=1,2,3,\cdots,N-1$). Repeating the process, all the unknown constants $\mathbf{q}_n^m$ ($m=0,1,2,\cdots M$; $n=1,2,3,\cdots,N-1$) are determined to generate the deformed skin shapes at these poses.

When the sculpting forces ($m=1,2,\cdots,M-1$; $n=1,2,3,\cdots,N-1$) are unknown, but the skin shape $\mathbf{q}_n^M=\tilde{\mathbf{q}}_n$ at the final pose is known, the following iterative method can be used to determine the unknown sculpting forces and deformed skin shapes from two example skin shapes which are at the initial and final poses, respectively.

First, $M=1$ in Equation (6.15) is set. Since $\mathbf{q}_n^1=\mathbf{q}_n^M$ ($n=1,2,3,\cdots,N-1$), Equation (6.15) can be used to determine $\mathbf{f}_n$ ($n=1,2,3,\cdots,N-1$). After $\mathbf{f}_n$ have been obtained, the sculpting forces at the time t_m are obtained by

$$\mathbf{f}_n^m=\frac{m}{M}\mathbf{f}_n \quad (m=1,2,\cdots,M-1;n=1,2,\cdots,N-1) \tag{6.16}$$

Substituting Equation (6.16) into (6.12), it is used to obtain all the unknown constants $\mathbf{q}_n^m$ ($m=1,2,\cdots,M-1$; $n=1,2,3,\cdots,N-1$).

If the obtained q_n^M is different from the known skin deformations $\tilde{q}_n$ at the final pose, the difference between $\tilde{q}_n$ and q_n^M, is calculated, i.e.,

$$\Delta q_n=\tilde{q}_n-q_n^M \tag{6.17}$$

Setting $M=1$ and $q_n^1=\Delta q_n$ in Equation (6.15), ΔF_n can be calculated from Equation (6.15), and the new sculpting forces $\tilde{F}_{qn}$ obtained with the following equation

$$\tilde{F}_n = F_{qn} + \Delta F_n \tag{6.18}$$

using the new sculpting forces $\tilde{F}_{qn}$ to replace F_{qn} in Equation (6.16), and introducing the updated F_{qn}^m into Equation (6.12) to determine the new skin deformation q_n^m $(m=1,2,\cdots,M-1;\quad n=1,2,3,\cdots,N-1)$. The iteration is carried out until the required accuracy is reached.

The deformed skin shapes at different poses were obtained by superimposing the calculated skin deformations to the skin shapes at the initial pose after rigid-body transformations. In the following two sections, a comparison will be made between various skin deformation techniques and two examples given to demonstrate the applications of the proposed technique.

6.5 Experimental Comparisons

In this section, three experimental comparisons have been made: 1). between real models and those of the alternate approaches, 2). between the dynamic finite difference solution and the static finite difference solution, and 3). between the proposed dynamic deformation model and SSDR.

6.5.1 Comparison between real models and those of alternate approaches

The human male arm models shown in Figure 6.3 give the real deformed skin shapes of human male arm movement. In order to facilitate the comparison between various skin deformation techniques, they are given in first row of Figure 6.3.

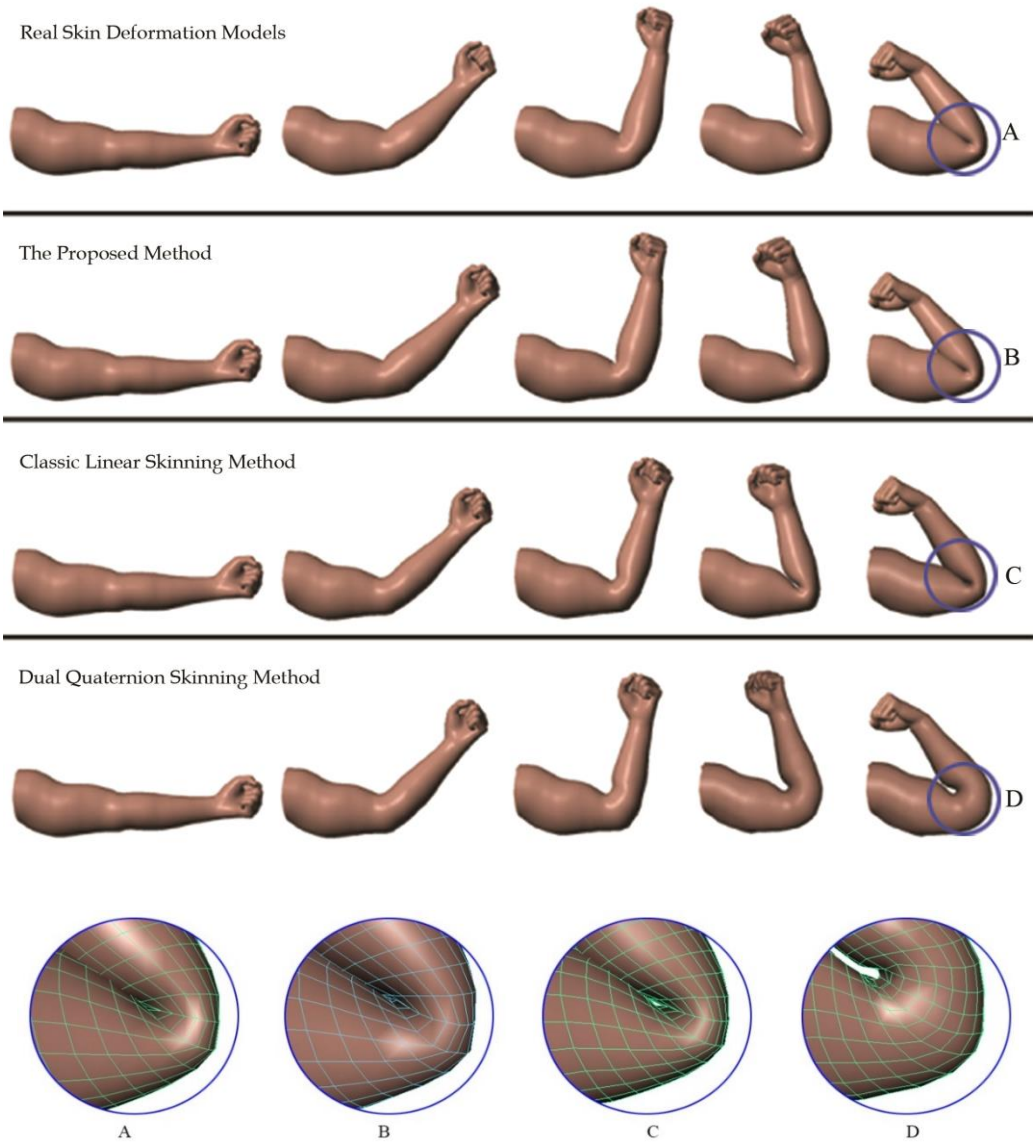


Figure 6.3 : Comparison of deformed skin shapes obtained with different techniques

Taking the real arm model shown in the first row and first column of the figure as the skin shape at the initial pose, two different techniques available in Autodesk Maya were applied to create the skin shapes at the other poses. These two techniques are: the Classic Linear Skinning Method (Thalmann et al. 1988; Lander et al. 1998, 1999) and the Dual Quaternion Skinning Method (Kavan and ˇZ´ara et al. 2013). The resulting skin shapes are depicted in the third and fourth rows, where the third row is from the Classic Linear Skinning Method, and the fourth row is from the Dual Quaternion Skinning Method. Comparing these skin deformation models with the real ones and the calculated ones, it is clear that the proposed skin deformation technique generates more realistic skin deformation shapes than the two skin deformation techniques available in Autodesk Maya.

Although the deformed skin shape at the final pose created by the Dual Quaternion Skinning Method is less realistic compared to the real one, this method avoids the collapsing joint artefact, as indicated by the deformed skin shapes at the third and fourth poses of the fourth row. In contrast, the Classic Linear Skinning Method suffers from the collapsing joint artefact, as indicated by the deformed skin shape at the fourth pose of the third row.

In addition to its modelling capacity in creating realistic skin deformations, the technique proposed in this chapter is also very efficient as demonstrated below. For $m=0$, the square matrix $\left[\mathbf{K}_{n0}\right]$ is determined by Equation (6.15) due to the contributions of the previous skin deformations $\mathbf{q}_n^{-1}=\mathbf{q}_n^1$. For all other values of m, the previous skin deformations $\mathbf{q}_n^{m-1}$ and $\mathbf{q}_n^m$ have no contributions to the square matrix $\left[\mathbf{K}_{nm}\right]$. If a_1, a_2, M and N for all the

curves and iterations are the same, the square matrix $[\mathbf{K}_{nm}]$ for all the values of $m > 0$ is the same. That is to say, it is only necessary to compute the inverse matrix $[\mathbf{K}_{nm}]^{-1}$ once. In total, only two inverse matrices are calculated: one for $m = 0$ and the other for all the values of $m > 0$. Then, all the skin deformations can be quickly obtained by carrying out the following calculations

$$\{\mathbf{Q}_{nm}\} = [\mathbf{K}_{nm}]^{-1}\{\mathbf{F}_{nm}\}$$
$$(m = 0,1,2,\cdots,M-1) \tag{6.19}$$

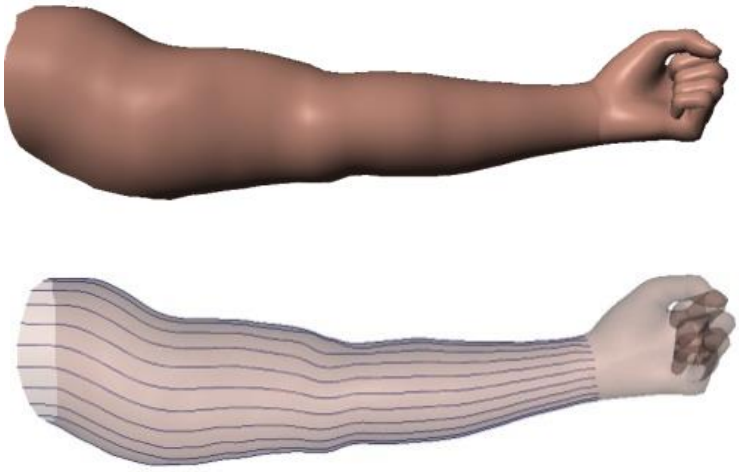


Figure 6.4 : Comparison between the real model and that of the curve-based representation

For the skin deformations of the human male arm movement, 22 curves are used to describe the arm model. On each curve, 34 points are uniformly collocated. In total, the human male arm model is described by 748 points. In contrast, the original polygon model has 1390 vertices, which is far more than the

points of the curve-based representation. Both the real arm model at the initial pose and the corresponding model described by the curve-based representation are shown in Figure 6.4. Comparing both models, there are no visible differences, indicating that the curve-based representation can describe a model of the same details with a much smaller data size than the polygon-based representation.

In Table 6.1, basic data are given using the proposed technique to determine skin deformations and the time taken to obtain the inverse matrix and determine skin deformations. The Gauss-Jordan elimination is used to find the inverse matrix $[\mathbf{K}_{nm}]^{-1}$. All the calculations were carried out on a Hewlett-Packard HP Z420 Workstation with 3.2GHz Intel(R) CPU E5-1650 32 GB of memory. As shown in the table, the total time taken to obtain deformed skin shapes of 30 frames is 0.01692 seconds. It indicates the proposed technique is efficient enough to generate real time skin deformations for character animation.

6.5.2 Comparison between dynamic and static finite difference solutions

The proposed dynamic deformation model has a greater coverage than the static deformation model and creates more realistic skin deformations.

When setting $a_2 = 0$ in Equation (6.2), the dynamic deformation model becomes a static one, indicating the former covers the latter. Using the finite difference method to solve the static deformation model, deformed skin shapes were obtained at the second, third and fourth poses and are depicted in the third row of Figure 6.5. The first row is from real skin deformations, and the second row is from the dynamic deformation model. Comparing the real skin

deformation models with those of the dynamic and static deformation models, it can be concluded that the dynamic deformation model gives more realistic skin deformations than the static deformation model.

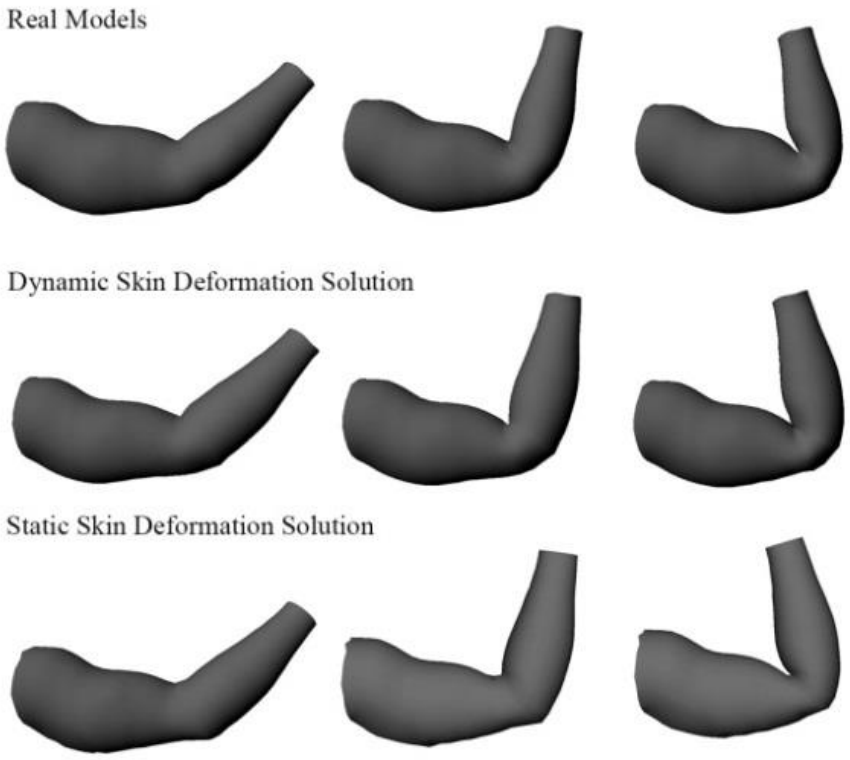


Figure 6.5 : Comparison between real deformed shapes and those of the dynamic and static deformation models

The efficiency comparison between the dynamic finite difference solution and the static finite difference solution is given in Table 6.1. It can be seen that the total time for 30 frames required by the former is very close to the latter, although it is a little larger.

6.5.3 Comparison between the Dynamic Deformation Model and SSDR Model

Smooth skinning decomposition with rigid bones as put forward by Le and Deng et al. (2012) is a state-of-art approach. Here the proposed approach is compared with SSDR in terms of both error metric and computational efficiency.

First, a description of how the proposed approach can be modified to determine dynamic skin deformations with J+1 (J>1) example skin shapes is given.

If J+1 example skin shapes at the poses j=0, 1, 2,…, J-1, J, i.e., at the time $t_j = 1/J \quad (j = 0,1,2,3\cdots, J-1, J)$ are known, the deformations and deformation rates at the poses j=1, 2,…, J-1, J can be determined with the following algorithm.

Assuming the position value of a curve at the pose j is $\mathbf{p}_0^j$ at $v=0$ and $\mathbf{p}_1^j$ at $v=1$, and the position value of the curve at the pose j+1 is $\mathbf{p}_0^{j+1}$ at $v=0$ and $\mathbf{p}_1^{j+1}$ at $v=1$, the deformation at the pose j+1 after excluding rigid-body transformations is $\mathbf{d}_0^{j+1} = \bar{\mathbf{p}}_0^{j+1} - \bar{\mathbf{p}}_0^j$ at $v=0$ and $\mathbf{d}_1^{j+1} = \bar{\mathbf{p}}_1^{j+1} - \bar{\mathbf{p}}_1^j$ at $v=1$ where $\bar{\mathbf{p}}_0^j$, $\bar{\mathbf{p}}_0^{j+1}$, $\bar{\mathbf{p}}_1^j$ and $\bar{\mathbf{p}}_1^{j+1}$ are respectively the position value of $\mathbf{p}_0^j$, $\mathbf{p}_0^{j+1}$, $\mathbf{p}_1^j$ and $\mathbf{p}_1^{j+1}$ after rigid-body transformations. With the same method, the deformation $\bar{\mathbf{d}}_0^{j+1}$ of the node can be found next to the node at $v=0$ and the deformation $\bar{\mathbf{d}}_1^{j+1}$ of the node next to the node at $v=1$. The deformation rates of the curve at $v=0$ and $v=1$ are determined by $\mathbf{s}_0^{j+1} = \left(\bar{\mathbf{d}}_0^{j+1} - \mathbf{d}_0^{j+1}\right)/h_0^{j+1}$ and $\mathbf{s}_1^{j+1} = \left(\bar{\mathbf{d}}_1^{j+1} - \mathbf{d}_1^{j+1}\right)/h_1^{j+1}$, respectively, where h_0^{j+1} is the length between two adjacent nodes at $v=0$ and

h_1^{j+1} is the length between two adjacent nodes at $v=1$ at the pose j+1. The deformation of curve excluding rigid-body transformation has been explained above in Figure 6.1 (page: 94).

Using a linear relationship to describe how boundary conditions change against the time, the following boundary conditions are reached at any instant $t_j \le t \le t_{j+1}$

$$
\begin{aligned}
&v=0 \quad \mathbf{q}(0,t) = \frac{t-t_j}{t_{j+1}-t_j}\mathbf{d}_0^{j+1} \\
&\qquad \frac{\partial \mathbf{q}(0,t)}{\partial t} = \frac{t-t_j}{t_{j+1}-t_j}\mathbf{s}_0^{j+1} \\
&v=1 \quad \mathbf{q}(1,t) = \frac{t-t_j}{t_{j+1}-t_j}\mathbf{d}_1^{j+1} \\
&\qquad \frac{\partial \mathbf{q}(1,t)}{\partial t} = \frac{t-t_j}{t_{j+1}-t_j}\mathbf{s}_1^{j+1} \\
&(t_j \le t \le t_{j+1})
\end{aligned}
\tag{6.20}
$$

Since the skin shape $\mathbf{q}_n^{j+1}$ at pose j (j=-1, 0,1,2,…, J-1; n=0,1,2,…,N) is known, $\mathbf{q}_n^{j+1}$ is substituted into (6.9) and the following equation obtained:

$$
\begin{aligned}
&a_1 N^4 \left[6\mathbf{q}_n^{j+1} - 4(\mathbf{q}_{n-1}^{j+1} + \mathbf{q}_{n+1}^{j+1}) \right] + \mathbf{q}_{n-2}^{j+1} + \mathbf{q}_{n+2}^{j+1} \\
&+ a_2 M^2 (\mathbf{q}_n^{j+1} - 2\mathbf{q}_n^{j} + \mathbf{q}_n^{j-1}) = \mathbf{f}(v_n, t_{j+1}) \\
&(j = 0,1,2,\cdots,J-1; n = 1,2,\cdots,N-1)
\end{aligned}
\tag{6.21}
$$

Considering Equation (6.13), Equation (6.21) can be solved subject to boundary conditions (6.20) to obtain the sculpting forces at all poses $\mathbf{f}(v_n, t_{j+1})$. Then the sculpting forces, Equation (6.21) and (6.20), can be used to calculate skin deformations between all two adjacent ones of the example poses.

6.5.3.1 Quantitative Error Analysis

The error metric $E_{RMS} = 1000\sqrt{E/(3.|V|.|t|)}$ has been used in Le and Deng et al. (2012) to quantitatively evaluate the SSDR model where E is the sum of the squared errors between the example skin shapes and those determined by the SSDR model at the example skin poses. For the horse-gallop with 49 example skin shapes (Sumner et al. 2004) used in Le and Deng et al. (2012), the approximate error generated by the SSDR model with 10 bones ($|B| = 10$) is $E_{RMS} = 4.6$. The error between desired and actual output has been calculated, the proposed approach gives the same skin deformations as the example skin shapes at the example skin poses, leading to zero error metric $E_{RMS} = 0$.

For the same horse-gallop, the SSDR model takes 2.1 minutes on a computer with a 2GHz single core CPU to obtain the weights and bone transformations used to calculate new skin shapes. On a computer with a 3.2GHz single core CPU, the proposed approach takes 0.045 seconds to determine all the inverse matrices $[\mathbf{K}_{nm}]^{-1}$ of all the curves defining the horse model, which are also used to calculate new skin shapes. The time using all obtained inverse matrices $[\mathbf{K}_{nm}]^{-1}$ to achieve 30 frames is 0.2568 seconds. Putting the two numbers together, the total time to obtain 30 frames is 0.3018 seconds.

6.6 Application Examples

In this section, two examples are given which demonstrate how to use the proposed technique to create skin deformations.

The first example is to create twisting deformations with the proposed technique, and examine whether the candy-wrapper anomaly arises. An original arm model at the time t=0 (pose 0) is shown in the top row of Figure 6.6, and one adjacent twisted shape at the time t=0.125 (pose 1) which is twisted a little downwards, depicted in the middle row of Figure 6.6. Solving Equation (6.14) with M=8, the twisting forces $\mathbf{f}_n^1$ at the pose 1 are obtained. Substituting the twisting forces $\mathbf{f}_n^j = j\mathbf{f}_n^1$ $(j = 2,3,\cdots,M)$ into Equation (6.12) with M=8 and m=1,2,…,M-1, the twisting deformations at the pose 2 to pose 8 are obtained, and depicted in pose 8 Figure 6.6. The proposed approach causes no candy-wrapper collapse problem as shown in Figure 6.6.

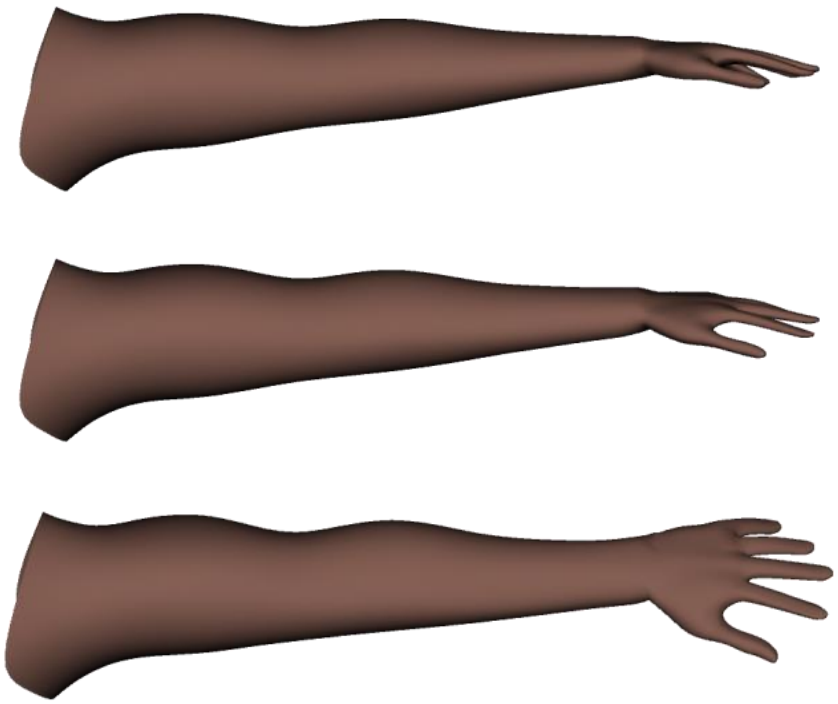


Figure 6.6: Twisting deformations of a human arm

The second example is to create skin deformations of horse running. As shown in Figure 4.4 of Chapter 4, all curves of a horse model were extracted. Information about the curves and the points on the curves is given in Table 6.1. With the proposed curve-based representation, the whole horse model is defined by 11,641 points in total. In contrast, the original polygon horse model has 30,134 vertices. Once again, the curve-based representation uses far fewer data to describe the same horse model.

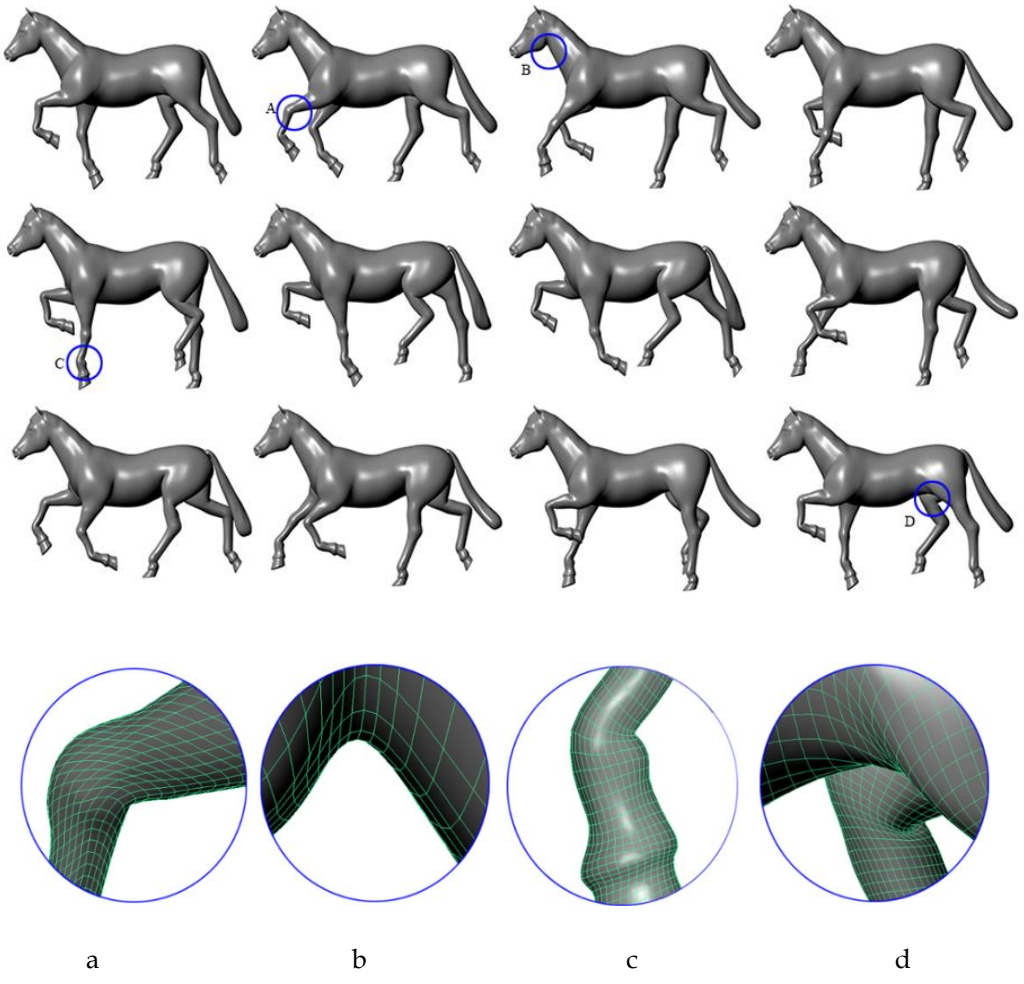


Figure 6.7 : Skin deformations of a horse running

With the proposed technique, skin deformations of a running horse were calculated and some of the obtained deformed models depicted in Figure 6.7.

These images and the animation also demonstrate that the proposed technique produces realistic skin deformations of a horse running. The time taken to obtain deformed skin shapes of 30 frames for horse legs, neck, torso, and tail is also given in Table 6.1. Once again, the time data demonstrate the efficiency of the proposed technique.

Table 6.1: Time used to determine skin deformations

Application Examples	No of Curves	No of Nodes	Dynamic solution			Static solution
			Time (sec) to get $[\mathbf{K}_{nm}]^{-1}$	Time (sec) to get solution for 30 frames	Total Time (sec) for 30 frames	Total Time (sec) for 30 frames
Hunan arm	22	34	0.00072	0.01548	0.01692	0.01546
Horse Leg1	32	68	0.00316	0.08854	0.09486	0.08638
Horse Leg2	32	68	0.00316	0.08854	0.09486	0.08638
Horse Leg3	32	66	0.00292	0.08361	0.08945	0.08460
Horse Leg4	32	66	0.00292	0.08361	0.08945	0.08460
Horse Neck	32	21	0.00046	0.00860	0.00952	0.00813
Horse Torso	32	17	0.00042	0.00556	0.00640	0.00590
Horse Tail	32	28	0.00050	0.01520	0.01620	0.01477

6.7 Summary

This chapter presents a new dynamic skin deformation technique. This new technique is based on a new model of dynamic deformations and an efficient finite difference solution of the model. Combining it with data-driven methods and curve-based representation of 3D models brings a number of advantages,

including realistic skin deformation, high computing efficiency, and fewer data are required.

7 Conclusions: Summary and Future Work

7.1 Conclusions

Skin deformation is one of the most challenging topics in computer animation. Various skin deformation techniques have discussed in this thesis (Chapter 2). Exisiting skin deformation techniques rely on the perception and skills of animators. As a result, these techniques become tedious, labour-intensive and costly. Data-driven methods do not consider the physics of skin deformations; they require sufficient skin shapes to achieve realism. Physics-based deformations are computationally intensive and are not popular in commercial animation. Partial differential equations (PDE) techniques have a great advantage for obtaining smooth surfaces, but require an efficient numerical method to solve the complicated partial differential equations. Curve-based methods are more efficient than physics-based methods and achieve realistic skin deformations. Compared to the existing skin deformation techniques, the curve-based dynamic method developed in this thesis makes the following contributions.

Existing character modelling and animation techniques, modeling and animation are dealt with separately. Even if a character model at the rest position has been well built, a lot of time and effort is required to animate the model by changing it into different shapes. This is the most important factor influencing the efficiency of character modelling and animation. This thesis has developed a new approach

for curve-based representation which simplifies the description of 3D skin surfaces and determination of skin deformations. This method can save time in animating geometric models by integrating modelling and animation into the same framework. The method can extract the representative curves from character models at their initial and final poses. Then, it determines the deformation differences of the curves between the initial and final poses. These deformation differences are introduced into a vector-valued dynamic differential equation to determine total sculpting forces, which drive a character model at its initial pose to deform to its final pose. From the total sculpting forces, the sculpting forces at different poses are obtained and introduced into the same dynamic differential equation to generate the deformation of the curves at these poses. Finally, the deformations of the curves are transferred to the polygon model to create different skin deformations at different poses.

- This thesis has developed a new method which can accurately determine the joint position and apply rigid-body transformations to exclude shape and position changes caused by translations and joint rotations. It also relates sculpting forces and skin deformations to joint rotations, combines the physics-based method with data-driven approaches to achieve realistic skin deformations with only two skin shape examples, and uses dynamic curve deformations to achieve high efficiency in dynamic skin deformations.

- This thesis has combined data-driven methods with physics of curve deformations to achieve good realism and high efficiency of skin deformations. The curve-based dynamic method developed in this thesis is particularly suitable for applications in computer graphics because most models in computer graphics

are expressed as surfaces. Efficiency and accuracy can be controlled by adjusting the sampling pose. The mathematical model treats the curve deformation of skin surfaces as beam bending, and describes the deformation with a vector-valued fourth order ordinary differential equation involving a varying force field. An efficient finite difference method is presented to solve the mathematical model and animate skin deformation. The finite difference solutions of static and dynamic skin deformations are based on time-dependent fourth-order ordinary differential equations. This new technique can save time and effort in animating virtual characters, produce character models and character animation in shorter timescales with great realism.

- The task of creating convincing skin deformation with a small data set is one of the biggest challenges in computer animation. Most character models are created using polygon-based modeling. The data size of polygon models is usually large. In order to improve realism, various physics-based modeling techniques have been developed. Unfortunately, these techniques rely on computationally expensive numerical methods. How to increase the efficiency of physics-based modeling techniques is an interesting and important topic. This thesis addressed this challenge by combining data-driven methods and curve-based representation of 3D models. With the proposed curve-based representation discussed in (Chapter 6), the whole horse model is defined by 11,641 points in total. In contrast, the original polygon horse model has 30,134 vertices. Evidence of this assertion is the observed curve-based representation, which uses far fewer data to describe the same horse model. The work described in thesis will be of great benefit to industrial users and researchers involved in

the modeling and animation of virtual characters. Since Autodesk Maya is a popular commercial software package, the Maya plug-in software can be directly employed to model and animate physics-based virtual characters efficiently. Due to its small data size, the proposed technique is also suitable for online applications.

Dynamic ordinary differential equations based skin deformation methods describe in this thesis consist of using curves to define geometric models, applying the rigging forces caused by joint rotations on curves, applying a mathematical model and fast resolution for dynamic deformations of curves, and animating the skin shape changes by those of the curves. In Chapter 6 (page 103) three detail experimental comparisons were made: 1) between real models and those of the alternate approaches, 2) between the dynamic finite difference solution and the static finite difference solution, and 3) between the proposed dynamic deformation model and SSDR. These experimental comparisons demonstrate that the curve-based dynamic method developed in this thesis creates a convincing skin deformation, however the current research has some limitations which can be investigated in future work.

7.2 Future Work

Following are some limitations which can be the focus of future work:

- The accurate joint determination technique presented in this thesis is based on a statistical analysis of the existing X-ray images available from the internet and hospitals. In future, this technique can be extended to create a large dataset of X-ray images of various joints of the human body.

This dataset could be very useful for animators, who could choose from the data or the position between maximum and minimum positions to determine accurate joint positions for their 3D models. Also, this dataset could be easily integrated into various 3D animation tools. However, the biggest challenge is that X-ray images for various parts of a human body are not always available, and X-rays are harmful to human health. Alternatively, we could create a dataset of human skeleton images.

- The work described in this research simplifies mathematical treatment of complex skin deformations. However, this research does not tackle the collision problem. Although some research (Volino et al. (1997), Cordier et al. (2000) and Maule et al. (2010)), has attempted to solve the collision problem. However, it is still an open problem. In future, this issue can be addressed by considering the collision of the surfaces defined by the curves defining the surfaces. The curve collision can be transformed into a surface collision, and then whether the two surfaces collide with each other can be investigated. One of the biggest challenges here is to develop a method which solves the collision problem in real-time, particularly in computer animation applications.

- The limitation of the method requiring two example skin shapes can be overcome by determining the force field, using the information from the joint rotation. Developing such a technique could help animators to produce animation quickly for games and movies. In future, further

investigation is required to study the relationship between force and joint moment; this could aid in understanding how force changes with different joint rotation and with different models.

- The realism of skin surfaces can be further enhanced by taking into account skin wrinkles. How muscle and other tissues affect skin deformations has not been investigated in this research. One way to alleviate this problem is to develop a curve-based dynamic wrinkle model and to consider the effects of other tissues, or to consider a sweep based surface. At the moment, it is clear that mathematical treatment of complex skin deformations using a curve-based dynamic method can produce realistic and convincing results.

- In this thesis, various polygonal models such as complete human models and a running horse have been investigated, which demonstrates that curve-based dynamic method developed in this thesis can be extended to work on other articulated character models. In future, this method can be generalised, which could be used in real-time character animation.

- In future, there we be a great opportunity to implement the existing curve-based dynamic method on a GPU. The computational time calculated in this thesis for generating new skin shapes for a horse-gallop model (Chapter: 6, page: 114) is based on a single-core CPU with the process done offline. The GPU-based algorithm can compute a curve-based dynamic method run-time, which could save animator time in

achieving realistic skin deformations. This could also help animators to control animation by adjusting the sampling pose in one framework.

References

Allen, B., Curless, B., Popovi´c, Z., 2002. Articulated body deformation from range scan data. In: SIGGRAPH Computer Graphics, volume 21, p. 612-619.

Aubel, A., Thalmann, D., 2001. Interactive modelling of the human musculature. In: proceedings of the 14th Conference on Computer Animation, p. 167-173.

Anderson, D. A. , Tannehill, J. C., Pletcher., R. H., 1984. Computational Fluid Mechanics and Heat Transfer, York: Hemisphere.

Beeler, T., Hahn. F., Bradley, D., Bickel. B., Beardsley, P., Gotsman, C., Sumner, R. W., Gross, M., 2011. High-quality passive facial performance capture using anchor frames. In: SIGGRAPH Computer Graphics, volume 30(4), p. 75:1–75:10.

Benjamin, C. L., Garman, G. R., Funston, J. H., 1997. Human Biology, Mac Graw-Hill, Newyork.

Bloor, M. I., Wilson, M. J., 1990, Using Partial differential equations to generate free-form surfaces, Computer-Aided Design, volume 22(4), p. 202-212.

Cao, C., Hou, Q., Zhou, K., 2014. Displaced dynamic expression regression for real-time facial tracking and animation. ACM Trans Graph, volume 33(4), p. 43:1–43:10.

Capell, S., Green, S., Curless, B., Duchamp, T., Popovi´c, Z., 2003. Interactive skeleton-driven dynamic deformations. In : Proceedings of Conference on Computer Graphics and Interactive Techniques, p. 586–93.

Capell, S., Burkhart, M., Curless, B., Duchamp, T., Popovic, Z., 2005. Physically based rigging for deformable characters. In : Proceedings of ACM SIGGRAPH/Eurographics symposium on Computer animation. ACM Press, p. 301-310.

Chai, J. X., Xiao, J., Hodgins, J., 2003. Vision-based control of 3D facial animation. Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation; 193–206.

Chadwick, J. E., Haumann, D. R., Parent, R. E., 1989. Layered construction for deformable animated characters, In SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques (New York, NY, USA,), ACM Press, p. 243-252.

Cosķun, S. B., Atay, M. T., O¨ztu¨rk, B., 2011. Advances in Vibration Analysis Research. InTech.

Cordier, F., Thalmann, N., M, 2000. Integrated system for skin deformation. In Computer Animation. Proceedings (pp. 2-8). IEEE.

Debunne, G., Desbrun, M., Cani, M. P., Barr, A. H., 2001. Dynamic real time deformations using space & time adaptive sampling. In : Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM Press, p. 31-36.

Debunne, G., Desbrun, M., Barr, A., and Cani, M. P., 1999. Interactive multiresolution animation of deformable models. In: Proceedings of the Euro graphics Workshop on Computer. Animation and Simulation, p. 133–144.

Debunne, G., Desbrun, M., Cani, M. P., Barr, A., 2000. Adaptive simulation of soft bodies in real-time. In : Computer. Animation., Philadelphia, USA, p. 15–20.

Dekanski, C. W., Bloor, M. I. G., Wilson, M. J., 1995, The generation of propeller blade geometries using the PDE method, Journal of Ship Research 39(2), 108-116.

Decarlo, D., Metaxas, D., 2000. Optical flow constraints on deformable models with applications to face tracking. Int. Journal of Computer Vision; 38(2):99–127.

Du, H., Qin, H., 2005. Dynamic PDE-based Surface Design Using Geometric and Physical Constraints. Graphical Models 67, p. 43–71.

Du, H., Qin, H., 2000. Direct manipulation and interactive sculpting of PDE surfaces, Computer Graphics Forum 19(3), p. 261-270.

Feng, W. W., Kim, B. U., Yu, Y., 2008. Real-time data driven deformation using kernel canonical correlation analysis. In: ACM Transactions on Graphics, 27, p. 91:1–91:9.

Floater, M. S., 2003. Mean value coordinates. In: Computer Aided Geometric Design 20(1), p. 19–27.

Fung, Y., C., 1981. Biomechanics: Mechanical Properties of Living Tissues. In: Springer Verla, p. 371-379.

Funck, V. F., Theisel, H., Seidel, H. P., 2006. Vector field based shape deformations. In: ACM Transactions on Graphics (SIGGRAPH 06), 25(3), p. 1118-1125.

Galoppo, N., Otaduy, M. A., Tekin, S., Gross, M., Lin, M. C., 2007. Soft articulated characters with fast contact handling. Computer Graphics Forum, 26(3), p. 243–53.

Georgii, J., Echtler, F., Westermann, R., 2005. Interactive simulation of deformable bodies on GPUs, In Simulation and Visualisation, p. 247-258.

Gourret, J. P., Thalmann, N. M., Thalmann, D., 1989. Simulation of Object and Human Skin Deformations in a Grasping Task, MIRALab, University of Montreal, University of Geneva, Swiss Federal Institute of Technology. Computer Graphics, vol. 23, p. 21-30.

Grinspun, E., Krysl, P., Schröder, P., 2002. Charms: a simple framework for adaptive simulation. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, volume 21, no 3, p. 281-290.

Guo, Z., Wong, K. C., 2005. Skinning with deformable chunks. Computer Graphics Forum (EUROGRAPHICS), volume 24, no 3, p. 373-381.

Hairer, E., Norsett, S. P. ,Wanner., G. . Solving Ordinary Differential Equations I: Nonstiff Problems. Second Revised Edition. Springer-Verlag, Heidelberg, 2009.

Hairer, E., Wanner., G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Second Revised Edition. Springer-Verlag, Heidelberg, 2010.

Huang, H., Chai, J., Tong, X., Wu, H. T., 2011. Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition. ACM Transactions on Graphics, 30(4), p. 74:1–74:10.

Hyun, D. E., Yoon, S. H., Chang, J. W., Kim, M. S., Jüttler, B., 2005. Sweep-based human de-formation. The Visual Computer, 21, p. 542–550.

Jane, W., Allen, V.G., 1997. Anatomically based modelling, In Proceedings of the 1997 Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 97), ACM Press, p. 173-180.

Jaluria, Y., Torrance, K. E., 1986. Computational Heat Transfer, New York: Hemisphere.

Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T., 2007. Harmonic coordinates for character articulation. ACM Transactions on Graphics 26.

Ju, T., Schaefer, S., Warren, J., 2005. Mean value coordinates for closed triangular meshes. In: ACM SIGGRAPH Papers, SIGGRAPH '05, p. 561–566. ACM, New York, NY, USA (2005).

Kavan, L., ˇZ ´ara, J., 2005. Spherical blend skinning: a real-time deformation of articulated models. Proceedings of Symposium on Interactive 3D Graphics and Games ACM, p. 9–16.

Kavan, L., Collins, S., ˇZ´ara, J., O'Sullivan, C., 2013. Geometric skinning with approximate dual quaternion blending. ACM Transactions on Graphics, 27(4), p. 105:1–105:23.

Kavan, L., Sorkine, O., 2012. Elasticity-inspired deformers for character articulation. ACM Transactions on Graphics, 31(6), p196:1–196:8.

Kim, J., Pollard, N. S., 2011. Fast simulation of skeleton driven deformable body characters. ACM Transactions on Graphics, 30(5), p121:1–121:19.

Kurihara, T., Miyata, N., 2004. Modeling deformable human hands from medical images. In: Proceedings of ACM SIGGRAPH/Eurographics symposium on Computer Animation, p. 355–363.

Lander, J., 1998. Skin them bones: Game programming for the web generation. Game Developer Magazine, 5(1), p.10-18.

Lander, J., 1999. Over my dead, polygonal body. Game Developer Magazine, 17(1), p.1–4.

Lee, S. H., Sifakis, E., Terzopoulos, D., 2009. Comprehensive biomechanical modeling and simulation of the upper body. ACM Transactions on Graphics, 28(4), p. 99:1–99:17.

Le, B. H., Deng, Z., 2012. Smooth skinning decomposition with rigid bones. ACM Transactions on Graphics, 31(6), p.199:1–199:10.

Le, B. H., Deng, Z., 2014, Robust and accurate skeletal rigging from mesh sequences. ACM Transactions on Graphics, 33(4), p. 84:1–84:10.

Li, D., Sueda, S., Neog, D. R., Pai, D. K., 2013. Thin skin elastodynamics. ACM Transactions on Graphics, 32(4), p. 49:1–49:10.

Li, S., Huang, J., de Goes, F., Jin, X., Bao, H., Desbrun, M., 2014. Space-time editing of elastic motion through material optimization and reduction. ACM Transactions on Graphics, 33(4), p. 108:1–108:10.

Lipman, Y., Kopf, J., Cohen-Or, D., Levin, D., 2007. Gpu-assisted positive mean value coordinates for mesh deformations. In: Proceedings of the fifth Eurographics symposium on Geometry processing, p. 117–123.

Lipman, Y., Levin, D., Cohen-Or, D., 2008. Green coordinates. ACM Transactions on Graphics, volume 27, no 3, p. 78:1–78:10.

Lowe, T. W., Bloor, M. I. G.,Wilson, M. J.,1990. Functionality in blend design, In: Computer Aided Design, 22(10), p. 655-665.

Lewis, J. P., Cordner M, Fong N., 2000. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In: Proceedings of SIG-GRAPH, p. 165–72.

Lewis, J. P., Anjyo, K., 2007. A Region-of-Influence Measure for Automatic Skinning. In: Proceedings of Image and Vision Computing New Zealand, p. 187–191.

Maule, M., Maciel, A., Nedel, L. 2010. Efficient Collision Detection and Physics-based Deformation for Haptic Simulation with Local Spherical Hash. In Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on (pp. 9-16). IEEE.

McAdams, A., Zhu, Y., Selle, A., Empey, M., Tamstorf, R., Teran, J., 2011. Efficient elasticity for character skinning with contact and collisions. ACM Transactions on Graphics, volume 30, no 4, p. 37.

Müller, M., Dorsey, J., Mcmillan, L., Jagnow, R., Cutler, B., 2002. Stable real-time deformations. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, p. 49-54.

Müller, M., Gross, M., 2004. Interactive virtual materials. In: Proceedings of Graphics Interface GI '04, p. 239-246.

Mimis, A. P., Bloor, M. I. G. and Wilson, M. J., 2001. Shape parameterization and optimization of a two-stroke engine. In: Journal of Propulsion and Power 17(3), p. 492-498.

Mohr, A., Gleicher, M., 2003. Building efficient, accurate character skins from examples. In: ACM Transactions on Graphics, volume 22, no 3, p. 562-568.

Monterde, J., Ugail, H., 2006. A general 4th-order PDE method to generate Bézier surfaces from the boundary. In: Computer Aided Geometric Design 23(2), p. 208-225

Nedel, L. P., Thalmann, D., 1998. Real time muscle deformations using mass spring systems, In: Proceedings of the Computer Graphics International, IEEE Computer Society, p. 156-165.

Nedel, L., Thalmann, D., 1998. Modeling and deformation of the human body using an anatomically-based approach. In: Proceedings of Computer Animation; p. 34-40.

Nedel, L., Thalmann, D., 2000. Anatomic modelling of deformable human bodies. In: The Visual Computer, 16(6), p. 306-321.

Neumann, T., Varanasi, K., Hasler, N., Wacker, M., Magnor, M., Theobalt, C., 2013. Capture and statistical modeling of arm-muscle deformations. In: Computer Graphics Forum, volume 32, no. 2pt3, p. 285-294.

Özi¸sik, M. N., 1994. Finite Difference Methods in Heat Transfer. Boca Raton, FL: CRC Press.

Park, S. I., Hodgins, J. K., 2008. Data-driven modeling of skin and muscle deformation. ACM Transactions on Graphics, volume 27, no 3, p. 96:1–96:6.

Pan, J. J., Yang, X., Xie, X., Willis, P., Zhang, J. J., 2009. Automatic rigging for animation characters with 3D silhouette. Computer Animation and Virtual Worlds, 20, p. 121–131.

Patankhar, S. V., Numerical Heat Transfer and Fluid Flow. New York: Hemisphere, 1980.

Rhee, T., Lewis, J. P., Neumann, U., 2006. Real-time weighted pose space deformation on the GPU. In: Computer Graphics Forum, volume 25, no 3, p. 439-448.

Richer, P., Artistic Anatomy, Watson-Gutpill Publications, 1981, Translated by Robert Beverly Hale.

Pighin, F., Salesin, D. H., Szeliski R., 1999. Resynthesizing facial animation through 3d model-based tracking. In: Computer Vision, Proceedings of Seventh International Conference on Computer Vision, volume 1, p. 143-150.

Platt, S. M., Badler, N. I., 1981. Animating facial expressions. ACM SIGGRAPH computer graphics, volume 15, no 3, p. 245-252.

Saragih, J., Lucey, S., Cohn, J., 2011. Deformable model fitting by regularized landmark mean-shift. International Journal of Computer Vision, 91(2), p. 200-215.

Scheepers, F., Parent, R. E., Carlson, W.E., May, S.F., 1997. Anatomy-based modeling of the human musculature. In: Proceedings of conference on Computer graphics and interactive techniques, p. 163–72.

Sederberg, T. W., Parry, S. R., 1986. Free-form deformation of solid geometric models. In: ACM SIGGRAPH computer graphics, volume 20, no 4, p. 151-160.

Soetaert, K., Cash, J. R., Mazzia, F., bvpSolve: Solvers for boundary value problems of ordinary differential equations. R package version 1 (2010).

Shi, X., Zhou, K., Tong, Y., Desbrun, M., Bao, H., Guo, B., 2008. Example Based dynamic skinning in real time, Zhejiang University, Microsoft Research Asia, Michigan University & Caltec, ACM Transactions on graphic, volume 27, No 3, Article 29.

Shen, J., Thalmann, N. M., Thalmann, D., 1994. Human skin deformation from cross sections. In Proc. Computer Graphics International, volume 94.

Singh, K., Fiume, E., 1998. Wires: a geometric deformation technique. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, p. 405-414.

Sumner, R. W., Popovi´c, J., 2004. Deformation transfer for triangle meshes. In: ACM Transactions on Graphics, 23(3), p. 399-405.

Terzopoulos, D., Waters, K., 1990. Physically-based facial modelling, analysis and animation. In: Journal of Visualization and Computer Animation, 1(2), p. 73-80.

Terzopoulos, D., Waters, K., 1991. Modelling animated faces using scanned data. In: Journal of Visualization and Computer Animation, 2(4), p. 123-128.

Terzopoulos, D., Platt, J., Barr, A., Fleischer, K., 1987. Elastically deformable models. ACM Siggraph Computer Graphics, volume 21, no 4, p. 205-214.

Thalmann, N. M., Laperri´ere, R., Thalmann, D., 1988. Joint-dependent local deformations for hand animation and object grasping. In: Proceedings on Graphics interface'88, p. 26–33.

Ugail, H., Bloor, M. I., Wilson, M. J., 1999. Techniques for interactive design using the PDE method. In: ACM Transactions on Graphics, 18.2, p. 195-212.

Ugail, H. and Wilson, M. J., 2003. Efficient shape parametrisation for automatic design optimisation using a partial differential equation formulation. In: Computers and Structures, 81(28-29), p. 2601-2609.

Vaillant, R., Barthe, L., Guennebaud, G., Cani, M. P., Rohmer, D., Wyvill, B., 2013. Implicit skinning: Real-time skin deformation with contact modeling. ACM Transactions on Graphics, 32(4), p. 125:1–125:12.

Venkataraman, K., Lodha, S., Raghavan, R., 2005. A kinematic variational model for animating skin with wrinkles. Computers and Graphics 29, p. 756–70.

Volino, P., Thalmann, N. M. ,1997. Developing simulation techniques for an interactive clothing system. In Virtual Systems and MultiMedia. VSMM'97. Proceedings., International Conference on (pp. 109-118). IEEE.

Wang, X. C., Phillips, C., 2002. Multi-weight enveloping: least-squares approximation techniques for skin animation. In: Proceedings of ACM SIGGRAPH/Eurographics symposium on Computer animation, p. 129–38.

Weber, O., Sorkine, O., Lipman, Y., Gotsman, C., 2007. Context aware skeletal shape de-formation. In: Computer Graphics Forum, 26(3), p. 265–74.

Weber, J., 2000. Run-time skin deformation. In: Proceedings of Game Developers Conference.

Wilhelms, J., Gelder, Van Gelder, A., 1997. Anatomically based modeling. In: Proceedings of Conference on Computer Graphics and Interactive Techniques, p. 173–80.

Wojtan, C., Turk, G. 2008. Fast viscoelastic behavior with thin features. In: Proceedings of ACM Transactions on Graphics, 27 (3), 47:1–47:8.

Xiong, X., De la Torre, F., 2013. Supervised descent method and its applications to face alignment. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 532–539.

Xiao, X., Lewis, J. P., Soon, S., Fong, N., Feng, T., 2006. A Powell optimization approach for example-based skinning in a production environment. In: Computer Animation and Social Agents, p. 141–150.

Yang, X. S., Somasekharan, A., Zhang, J.J., 2006. Curve skeleton skinning for human and creature characters. In: Computer Animation and Virtual Worlds, 17(3-4), p. 281–92.

Yoon, S. H., Lim, C. G., Kim, M. S., 2008. Sweep-Based Plausible Elastic Deformations. ETRI Journal, 30(1), p. 152-154.

You, L. H., Yang, X. S., Zhang, J. J., 2008. Dynamic skin deformation with characteristic curves. In: Computer Animation and Virtual Worlds, 19, p. 433–44.

You, L. H., Yang, X. S., Pachulski, M., Zhang, J. J., 2007. Boundary Constrained Swept Surfaces for Modelling and Animation, EUROGRAPHICS 2007 and Computer Graphics Forum 26(3), p. 313-322.

You, L. H., Chaudhry, E., Zhang, J. J., 2009. Physics and example based skin deformations for character animation. In: Computer-Aided Design and Computer Graphics, CAD/Graphics' 09. 11th IEEE International Conference, p. 62-67.

You, L. H., Zhang J. J., 2003. Fast Generation of 3D Deformable Moving Surfaces, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 33(4), p. 616-625.

Zhang, L., Snavely, N., Curless, B., Seitz, M. S., 2004. Space time faces: High resolution capture for modeling and animation. In: ACM SIGGRAPH Proceedings, volume 23, p. 548–58.

Zhang, J. J., You, L. H., 2002. PDE based surface representation-Vase design, Computers & Graphics, 26(1), p. 89-98.

Zhang, J. J., You, L. H., 2004. Surface blending using a power series solution to fourth order partial differential equations. In : Proceedings of International Journal of Shape Modeling, 10(2), p. 155-185.