Efficient Sketch-Based 3D Character Modelling

ISMAIL KHALID KAZMI

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

June, 2016

Copyright

Abstract

Sketch-based modelling (SBM) has undergone substantial research over the past two decades. In the early days, researchers aimed at developing techniques useful for modelling of architectural and mechanical models through sketching. With the advancement of technology used in designing visual effects for film, TV and games, the demand for highly realistic 3D character models has skyrocketed. To allow artists to create 3D character models quickly, researchers have proposed several techniques for efficient character modelling from sketched feature curves. Moreover several research groups have developed 3D shape databases to retrieve 3D models from sketched inputs.

Unfortunately, the current state of the art in sketch-based organic modelling (3D character modelling) contains a lot of gaps and limitations. To bridge the gaps and improve the current sketch-based modelling techniques, this research aims to develop an approach allowing direct and interactive modelling of 3D characters from sketched feature curves, and also make use of 3D shape databases to guide the artist to create his / her desired models. The research involved finding a fusion between 3D shape retrieval, shape manipulation, and shape reconstruction / generation techniques backed by an extensive literature review, experimentation and results. The outcome of this research involved devising a novel and improved technique for sketch-based modelling, the creation of a software interface that allows the artist to quickly and easily create realistic 3D character models with comparatively less effort and learning. The proposed research work provides the tools to draw 3D shape primitives and manipulate them using simple gestures which leads to a better modelling experience than the existing state of the art SBM systems.

Table of Contents

List of Figures

List of Tables

Acknowledgement

First of all, I am thankful to Almighty God for giving me wisdom and guidance throughout my life and bestowing me with the will to finish my PhD.

My sincere appreciation and thanks goes to my supervisors Dr. Lihua You and Professor Jian Jun Zhang for giving me their invaluable advices and support throughout the duration 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 would also like to thank Bournemouth University and the National Centre for Computer Animation (NCCA) for providing me with a great learning environment, and Ms. Jan Lewis for all her support with non-academic matters. I would also like to thank Dr Xiaosong Yang at the NCCA for his excellent feedback and support in improving the quality of my published papers.

This research is supported by grants of the 2013 International Exchanges Scheme (Grant no. IE131367), the Royal Society, United Kingdom, and the Sino-UK Higher Education Research Partnership for Ph.D Studies Project. In this research, I have also gained knowledge and developed research during a three-month research project at Zhejiang University, China. This research project was sponsored by Sino-UK Higher Education Research Partnership. The project evaluated the importance of efficient and modern techniques in sketch-based motion retreival, which were essential for my literature review of 2D and 3D shape descriptors. This research project helped improve my understanding of advanced problem solving skills such as evaluation and implementation of Computer Vision algorithms. My survey on 2D and 3D shape descriptors talks heavily about some important algorithms I encountered during this research project.

I would also like to thank my loving parents, my sisters, and my research colleagues for their support, which made it possible for me to undertake this research.

I would like to express my special gratitude to my wife whose patience and moral support encouraged me to complete my PhD research, and also for giving birth to my beautiful daughter in the third year of my PhD, which came as an immense joy and blessing for me and my wife.

This thesis is dedicated to my parents who taught me how to read and write, and to the loving memory of my uncle (1930 - 2013) who encouraged me to undertake my PhD research.

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.

Thesis Outline

This thesis consists of six chapters.

- **Chapter 1 - Introduction:** This chapter provides a brief introduction to the importance of sketch-based modelling, followed by conventional approaches for 3D character modelling and finally the aims and objectives 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:** In this chapter, a thorough literature review of the 2D and 3D shape descriptors is provided followed by a survey on Sketch-based modelling systems. Both of these surveys review the strengths and weaknesses of state of the art methods, identifies some research challenges, and presents suggestions for future work.

- **Chapter 3 - Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Deformations:** In this chapter, a novel sketch-based modelling pipeline is presented which deforms a template model using a semi automatic method of computing accurate occluding contours and a database of 3D template models.

- **Chapter 4 - A Hybrid Approach for Character Modelling Using Geometric Primitives and a Shape-from-Shading Algorithm:** This chapter presents a hybrid sketch-based modelling approach which combines base mesh modelling using geometric primitives, with surface detail modelling using a shape-from-shading algorithm.

- **Chapter 5 - Sketch-to-Box: A New Character Modelling Technique:** This chapter outlines a new approach to sketch-based character modelling, which uses boxes/cubes as the basic building blocks of the entire character creation. Starting from creating a pose of the character made of simple boxes, this new approach attempts to add details to the boxes using sketching gestures in order to sculpt them to resemble different body muscles.

- **Chapter 6 - Conclusion and Future Work:** This chapter discusses the conclusion of work presented in this thesis with its impact. It also discussed the limitations of the contributions presented and the possible future directions that can take this research further.

- **References**

1 Introduction

1.1 Overview

3D characters are vital assets that are widely used in many areas in the creative industry especially in the visual effects and games development industries. The rapid advancement of technology has made the creative industry one of the fastest growing industries with a huge market. The current global computer animation and video games industry has reached a total sale of $222.8 billion. As the gaming platforms and special effects in movies and TV are becoming more and more sophisticated, the demand for highly realistic 3D characters is also rising. However the process of generating 3D characters is not straightforward. It usually takes several days for artists to model, sculpt and polish a 3D character model. Therefore, a more efficient modelling technique for realistic 3D characters will create a huge impact on this fast growing and huge market.

Within the games and simulation industry, characters with a low polygon count are the preferred choice as these are performance efficient when rendered in real-time. On the other hand, within the animation and VFX (Visual Effects) industry, high polygon characters are needed to be rendered on the big screen. However, low polygon characters are also increasing in popularity in the animation industry, especially during the design and prototyping phase as these assets provide a visual cue to the artists about how to pose the characters to complete the final scenes.

Currently in the context of 3D character modelling, box modelling using polygons, NURBS modelling, and organic modelling are popular approaches. These are discussed in Section 1.2. Unfortunately without considerable devotion of time and effort, these techniques are very hard to master and without considerable experience result in undesired 3D character models.

Sketch-base modelling (SBM), which has been under research over the past two decades, is a technique that allows artists to generate 3D characters by transforming the features present in a 2D sketch into a 3D model, either directly or via data-driven support. It is considered to be a more artist friendly and natural approach of modelling 3D characters. Unfortunately, since SBM is still in its infancy with lack of professional tools, it can result in rough and low polygon models with little or no details. Moreover, without careful modelling the topology of 3D models resulting from SBM approaches is inaccurate. Consequently, adding details to such models using sculpting tools like Pixologic ZBrush (ZBrush) or Autodesk Mudbox (Mudbox), is a time consuming process.

1.2 Conventional approaches for character modelling

Traditionally, 3D character modelling is achieved using several well-known approaches: 1) Box modelling, 2) Organic sculpting, 3) NURBS/Spline modelling, 4) Photogrammetry and 5) 3D laser scanning.

1.2.1 Box modelling

This approach requires the artist to use a single cube as a starting object and then progressively modify the cube by adding more details in order to depict muscles, bones, and other body parts such as arms, legs and torso (Daniele, 2012). During the modelling process, the artist needs to adjust the topology of the mesh in order to adhere to the anatomical flow of the skin, muscles and bones. The artist can also use as an aid, a set of character model sheets for the input sketch showing the character from two or more orthographic views. There are several advantages to this approach. Firstly, it is suited for modelling low polygon characters for games, where there is usually a requirement to create a substantial character asset base. Secondly, characters modelled using this technique are easy to animate and pose. Unfortunately, with this approach it is very hard to create highly detailed characters and thus not suitable for photo-realistic rendered images. Another drawback is that the topology resulting from this approach is often inconsistent and hard to fix.

1.2.2 Organic sculpting

Organic sculpting deals with sculpting a very high polygon base mesh from the beginning using a suitable package like Pixologic ZBrush (ZBrush) or Autodesk Mudbox (Mudbox) (Spencer, 2011). The artist usually start out with a very high polygon base mesh and then molds it to depict different muscles of the human body such as torso muscles, arm muscles and leg muscles. In this approach, the artist need not model the character in a T-pose or A-pose; instead he has the freedom to sculpt out additional details in an articulate pose. Thus this approach is favored by artists creating high resolution assets for movies, games, and 3D printing.

The above two approaches are hard to learn and master, and require a lot of manual operations to create and manipulate character models. Over the past two decades, sketch-based modelling (SBM) has been an active topic in the computer graphics research community to overcome the drawbacks of traditional character modelling approaches (Olsen et al., 2009). Since SBM techniques mimics traditional arts tools (pencil and paper), these approaches provide free hand and natural tools to model characters that carry out geometric modelling efficiently. Unfortunately, sketch-based modelling does not produce accurate and detailed characters that can be used in movies or games, and thus is mostly suited for casual or experimental 3D modelling.

1.2.3 NURBS/Spline Modelling

NURBS (Non Uniform Rational Basis Spline) modelling is a modelling technique that uses 3D spline curves to create very smooth models (Rogers, 2000). Unlike polygons, NURBS are resolution independent, they are controlled by fewer points called control points. Control points are the basic building blocks of a 3D spline curve. One of the main advantages to using NURBS is that the modelled character can achieve a very high level of smoothness. The splines (for drawing curves) within NURBS modelling can typically be controlled easily and manipulated, and their structures are influenced by control points, drawing the curves in particular directions.

NURBS based models can only be created with a grid of NURBS curves, like moulding a wire mesh. This rigid topology surface is hard to extend, but one can add extra geometry, blend between two surfaces, adding a new surface between with both ends matching perfectly with the originals. Another disadvantage of using NURBS is that artist cannot cut holes in them because they are rigid, unless the NURBS patch can be converted into a polygon patch.

1.2.4 Photogrammetry

Photogrammetry is actually image based modelling, which is a process by which 3D objects are algorithmically derived from a set of static two-dimensional images (Mikhail et al, 2001). Image based modelling is often used in situations where time or budgetary restrictions do not allow for a fully realized 3D asset to be created manually. An example of a model achieved with photogrammetry is shown in the Figure 1 below.

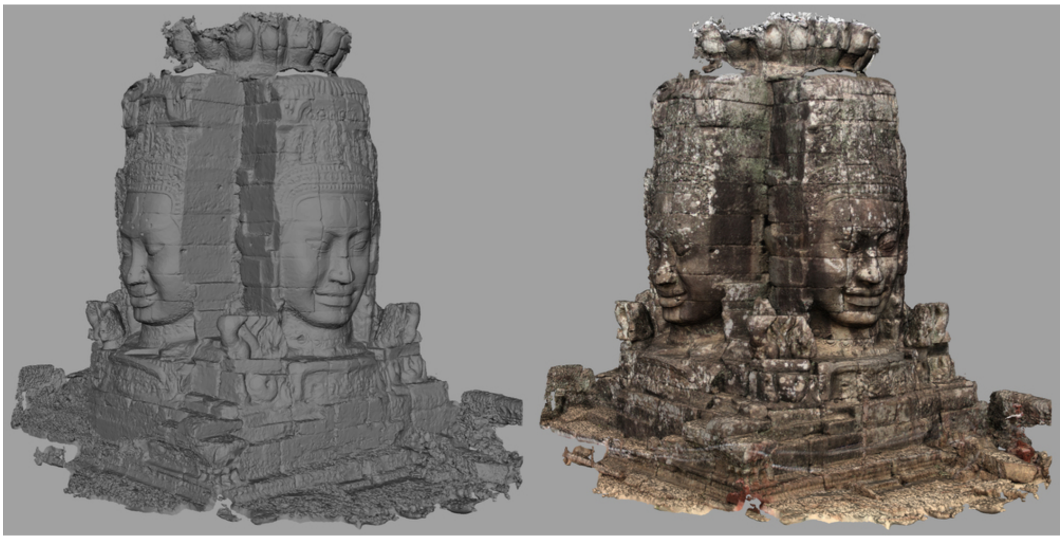


Figure 1: Photogrammetry example. *Image source: http://www.joshelmore.com/photogrammetry.*

1.2.5 3D laser scanning

3D Scanning is a method of digitizing real world objects when an incredibly high level of photo-realism is required (Bernardini and Rushmeier, 2002). It can be considered as a more accurate approach to photogrammetry. A real world object (or even actor) is scanned, analyzed, and the raw data (typically an x,y,z point cloud) is used to generate an accurate polygonal or NURBS mesh. 3D scanning produces the most accurate and detailed 3D models, however, it requires the artist to fix the topology of the mesh heavily due to the presence of a high amount of polygon soups in the raw mesh, which is a very challenging task. A notable technique for acquiring geometry using 3D laser scanning is Microscribe (http://www.3d-microscribe.com/). In this technique the user only needs to trace over the contours of a physical object and build complex 3D models in a matter of minutes. It was designed to capture the physical properties of actual objects and transform them accurately into complete 3D models. One major advantage of this technique is that it works with physical objects of any shape, size and material.

1.3 Aims and objectives of this research

From the above discussion it appears that there clearly exists a gap in popular modelling approaches and sketch-based modelling methods. This gap is due to the shortage of novel and practical modelling approaches and user interface which is easy to learn and use, and able to create fine 3D models quickly and readily through fewer manual operations. In order to bridge this gap, the overall aim of this research is to develop a new, simple and efficient modelling approach and software environment for easy and quick creation of virtual 3D characters.

The main question for this PhD project can be posed as follows: Is there any system versatile enough to allow the user to create a character as realistic as possible using sketch-based input for creating and handling geometry through mouse/pen strokes, and in addition to that uses a database-driven approach to achieve the desired realism in 3D character? Another question posed in the project is: What kind of character models will the system support for modelling? A character can be an extra-terrestrial creature, underwater creature, four-legged animal, or simply biped character models like human beings? For the latter question, the primary focus is on human like characters.

The research questions posed above served as important rationale behind this PhD thesis. To address these questions, the following are the proposed aims and objectives of this research.

1. To survey and understand feature curves extraction techniques or shape descriptors to automatically extract relevant features from 3D character models. Feature curves are defined as those curves which are most suitable for describing the shape of character models. They include contour curves in two orthotropic views of character models and those of their parts which have not been described by the contour curves of the character models, together with some other representative characteristic curves not present in the two orthotropic views.

2. To survey and understand sketch-based modelling (SBM) techniques. These include multi-view sketch-based modelling systems, and systems that reconstruct 3D surfaces based on input strokes. The former type of SBM systems require two or more views from the users and in return generate 3D models based on the views. The latter type of SBM systems require free hand strokes from the artists to draw on a canvas and based on some mathematical equations, infer the strokes and generate 3D surfaces corresponding to the input strokes. A detailed survey of these two types of SBM systems is in Chapter 2.
3. To design and develop a novel approach to sketch-based modelling based on computer vision techniques to retrieve a template 3D model from a database and then deforming of the model using deformation algorithms to resemble the input sketch.
4. To develop a sketch-based modelling technique for direct creation of 3D models utilizing 3D geometric primitives and enhancing the plausibility by adding details to the surface using shape-from-shading techniques.
5. To develop a new sketch-based modelling technique for learners or novice artists, which can aid them in teaching 'surface anatomy' of a human character. This new approach should be designed to solving the dilemma arising due to the steep learning curve required for human character modelling.
6. To implement a software environment in the form of a C++ application. It should provide an easy to learn user interface for the proposed feature curve-based modelling approach and quick creation of the 3D character models. This software environment will be developed and used to justify the theoretical development.

Researchers have for decades tried to solve the problem of efficiently and effectively describing models using its geometrical properties such as curves, polygons, depth buffer images and voxels etc. The upshot of this persistence is the immense applications in designing modern 3D modelling tools that provide a feeling of natural sketching. In a survey from (Yubin et al., 2007), the authors have identified that the realistic character modelling process can be greatly improved by re-using existing characters by retrieving them from a database using 3D shape retrieval algorithms. In this way existing characters can be fully or partially utilized to produce the desired realistic characters by making some minor and simple tweaks to their feature curves. Thus the aim of this research is to develop user friendly tools which allow artists to use conventional sketching techniques to generate 3D character models bypassing the tedious process of hand crafting a 3D model from scratch.

1.4 Contributions

To fulfill the aims and objectives of the research, this thesis have made three novel contributions, which are as follows.

1. **Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Deformations:** In this proposed approach, a novel sketch-based modelling pipeline is presented which deforms a template model using a semi automatic method of computing accurate occluding contours and a database of 3D template models. This approach attempts to overcome the problems of lengthy computations involved with automatic deformation of the template model inherent in previously proposed methods, and makes it efficient by introducing semi-automatic, thus descreasing the overall deformation duration. Another contribution of this approach is that it provides a user interface to draw rough occluding contours easily and perform skeleton deformation.

2. **A Hybrid Approach for Character Modelling Using Geometric Primitives and a Shape-From-Shading Algorithm:** This chapter presents a hybrid sketch-based modelling approach which combines base mesh modelling using geometric

primitives, with surface detail modelling using a shape-from-shading algorithm. This approach attempts to overcome the limitations present in some earlier sketch-based modelling systems such as the production of very basic and less detailed character models from sketches. Thus the proposed appraoch improves over the previous systems by utilizing a robust algorithm to quickly transfer surface detail from a source mesh to a target mesh.

3. **Sketch-to-Box: A New Character Modelling Technique:** This chapter outlines a new approach to sketch-based character modelling, which uses boxes/cubes as basic building blocks for the entire character creation. Starting from creating a pose of the character made of simple boxes, this new approach attempts to add details to the boxes using sketching gestures in order to sculpt them to resemble different body muscles. The approach has been proposed as a teaching tool for novice artists and aims to train them to learn the skills of human character creation following reasonably well surface anatomy. Through this technique the aim was to solve the problems present in current 3D modelling packages which offer steep learning curves. Moreover since the current character modelling approaches are considered to be less suitable for teaching how to model artistic / surface anatomy, the proposed approach aims to bridge this gap by providing a sketch-based modelling interface and toolset.

Apart from the above contributions, this thesis also provides a detailed review of 2D and 3D shape descriptor techniques as well as sketch-based modelling systems.

1.5 Publications

The research presented in this thesis has led to the following publications in peer reviewed journals and conferences:

1. **Kazmi, I.K.**, You, L. and Zhang, J.J., 2016. A hybrid approach for character modelling using geometric primitives and shape-from-shading algorithm. *Journal of Computational Design and Engineering*. *3*(2), pp.121-131.

2. **Kazmi, I.K.**, You, L., Yang, X., Jin, X. and Zhang, J.J., 2015. Efficient sketch-based creation of detailed character models through data-driven mesh deformations. *Computer Animation and Virtual Worlds*, *26*(3-4), pp.469-481.

3. **Kazmi, I.K.**, You, L. and Zhang, J.J., 2014, August. A Survey of Sketch-based Modelling Systems. In *Computer Graphics, Imaging and Visualization (CGIV), 2014 11th International Conference on* (pp. 27-36). IEEE.

4. **Kazmi, I.K.**, You, L. and Zhang, J.J., 2013, August. A survey of 2D and 3D shape descriptors. In *Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference* (pp. 1-10). IEEE.

2 Related Work

The aim of this research is to propose and develop efficient sketch-based modelling systems for creating character models. This required me to carry out a thorough literature review on recent advancements in 2D/3D shape retrieval and sketch-based modelling systems. Initially, a thorough review of the state of the art in 2D and 3D shape retrieval algorithms was carried out. As described in the following section, these algorithms are based on feature vectors called shape descriptors. Following this review, another thorough review was done on state of the art sketch-based modelling systems.

2.1 A Survey of 2D and 3D Shape Descriptors

Three dimensional objects possess superior importance in the fields of gaming, movies, TV, engineering design, medicine, and military research. The demand has been growing to efficiently transmit these over the Internet with the rise of cloud computing technologies. Currently the common way to search for images and 3D models over the internet is via text queries and online catalogues. Retrieval of images using another image as a query is still not sophisticated enough to provide us with the desired results. The same is true for online 3D model retrieval. Moreover in the entertainment, games and special effects industries, there is a huge demand for modelling highly realistic characters quickly and easily. Unfortunately, modelling realistic 3D characters is still a time-consuming and laborious task, which requires highly skilled and experienced modellers. Sketch-based modelling techniques exist, however, these techniques are prone to roughness, inaccuracy, and difficulty. Thus there exists a gap between traditional modelling techniques and sketch-based modelling techniques. This requires powerful content-based retrieval techniques and sketch-based modelling interfaces to be implemented.

Content-based information retrieval (CBIR) has been studied by researchers for several decades. CBIR is the process of using shape descriptors to extract important visual features from the boundary and the interior of a shape from shape databases. Some important features include contours curves, shape signature, shape histogram, shape context, and spectral features (Zhang and Lu, 2004). Several notable databases have been developed in the past including MPEG-7 shape database (Bober, 2001), Princeton Shape Benchmark (Shilane et al., 2004), Konstanze University (CCCC) 3D benchmark, McGill Shape Benchmark (MSB) (Zhang et al., 2005), and National Taiwan University database (NTU) (Chen et al., 2003). Several excellent surveys on 2D and 3D shape descriptors (SDs) exist including but are not limited to (Tangelder and Veltkamp, 2008), (Zhang and Lu, 2004), (Zhang et al., 2007), and (Yang et al., 2007). This chapter discusses and summarizes several important 2D and 3D shape descriptors which have undergone extensive research in CBIR, and discuss their advantages and disadvantages. Taxonomies of the 2D and 3D shape descriptors are also presented here. Moreover a few hybrid techniques have been presented, which have gained importance in the recent years because of their promising effectiveness and performance.

2.1.1 Definition and characteristics of shape descriptors

Generally speaking, a shape descriptor is a simplified representation of a 2D or 3D shape in the form of a vector containing a set of numerical values or a graph-like structure used to describe the shape geometrically or topologically (Yang et al., 2008), (Akgül, 2007). Shape descriptors are evaluated on the basis of several characteristics which define the overall quality and effectiveness of a shape descriptor. The following common characteristics of an effective shape descriptor have been proposed in (Yang et al., 2008), (Tangelder and Veltkamp, 2008), (Zhang and Lu, 2004), and (Iyer et al., 2005).

a) Discriminative accuracy: To accurately distinguish one shape from another based on subtle differences
b) Transformation (translation, scaling, and rotation) invariance: Also known as pose normalization
c) Robustness against model degeneracies / roughness

Table 1: Classification of 2D shape descriptors.

Contour Based Descriptors	Fourier Descriptors (FD)
	Wavelet Descriptors (WD)
	Curvature Scale Space (CSS)
	Shape Context (SCD)
Region Based Descriptors	Zernike Moment Descriptors (ZMD)
	Scale Invariant Feature Transform (SIFT)
	Angular Radial Transform (ART)
Hybrid Descriptors	FD + ART
	FD + ZMD

d) Uniqueness: Each shape descriptor must be uniquely coupled with a unique shape
e) Performance and memory efficient
f) Partial matching: robust against incomplete shapes
g) Insensitive to noise: Small changes in the shape to lead to small changes in the shape descriptor

2.1.2 2D Shape Descriptors

Over the past two decades, 2D SDs have been actively utilized in 3D search engines and sketch-based modelling techniques. Some of the most popular 2D SDs are Curvature Scale Space (CSS), Fourier Descriptor, SIFT, ART, ZMD, and Shape Contexts. In existing literature, 2D SDs are classified into two broad categories: contour based and region based. Contour based SDs extract shape features from the contour of a shape only. In contrast, region based SDs obtain shape features from the whole region of a shape. In addition, hybrid techniques have also been proposed in the past, which are a combination of contour and region based techniques (Pooja, 2012). Therefore this chapter classifies 2D SDs into three categories where the third category in the taxonomy is called Hybrid 2D Shape Descriptors. Table 1 shows the classification of the 2D SDs.

2.1.2.1 ***Contour Based Descriptors:*** Contour based descriptors (CBD) only consider the boundary of the shape and neglect the information contained in the shape interior (Singh, 2011). These descriptors are very efficient at filtering out the results based on the boundary points because of their low computation complexity. However, they are not good at handling image noise and thus are not accurate.

1) Fourier Descriptor (FD): A Fourier descriptor (FD) is a representation of the shape obtained after the application of the Fourier transform on the coefficients of the shape signature of the shape. It can be obtained through extracting boundary coordinates, computing shape signatures, and generating Fourier descriptors. Here, a shape signature means any 1-D function used to represent 2-D shapes or boundaries.

Boundary coordinates are extracted from the planar closed curves such as sketches or silhouette images. One of the four frequently used shape signature functions: centroid distance, complex coordinates, curvature function, and cumulative angular function, can be used to compute shape signatures. Among them, the Fourier descriptor method using centroid distance performs the best (Zhang et al., 2012).

When using centroid distance, the centroid of N boundary points is first found. Then, a shape signature $s(t)$ $(n = 0,1, \ldots, N-1)$ is obtained to represent the distances of the N boundary points to their centroid. Finally, the discrete Fourier transform of $s(t)$

$$u_n = \frac{1}{N}\sum_{t=0}^{N-1} s(t)\exp\left(\frac{-j2\pi nt}{N}\right), n = 0,1, \ldots, N-1 \qquad (2.1)$$

is used to determine Fourier coefficients u_n, which are usually called Fourier descriptors (FDs) of the shape, denoted as FD_n $(n = 0,1, \ldots, N-1)$ (Zhang and Lu, 2001).

Shape signature computation is a preliminary step of FD in which a 1-D function is obtained from the boundary coordinates of the shape.

According to Zhang and Lu (2004) the advantages of FDs are: 1) simple to compute, 2) simple to normalize for simplifying the shape matching process, 3) able to capture both global and local features, 4) insensitive to noise.

Zhang et al. (2012) proposed a modified FD (MFD). Their technique rests on the idea that geometric information along with non-geometric information in the contour of an object is important for accurate discrimination which according to the authors is ignored in the previous works.

2) Wavelet Descriptor (WD): Wavelet descriptor (WD) was initially proposed by Chuang and Kuo (1996) in which the authors employed wavelet transforms to describe the shape of planar closed curves. It is a multi-resolution approach which decomposes the shape into several components in multiple scales. The components in the higher resolutions contain the global information and the components in the finer resolutions contain the more detailed local information. Wavelet descriptors have several useful advantages over other descriptors, including insensitive to noise, invariance, uniqueness, and stability against boundary variations. Chuang and Kuo (1996) concluded that as compared to FDs, WDs are more efficient in representing and detecting local features of a curve.

Khoje and Bodhe (2012) used a combination of FD and WD to evaluate the irregularity of the mango fruit shape, and named the combination Wavelet Fourier Descriptor (WFD). WFD outperforms region based and contour based FD with classification efficiency of 89.83% making it a promising 2D descriptor to be used in view based 3D object retrieval. However WFD has not been tested for shape recognition where the shape resembles a human character. To evaluate this technique in human character shape retrieval and 3D search engines could be a possible future research direction.

3) ***Curvature Scale Space (CSS):*** CSS is one of the most widely used algorithms in content-based image retrieval (CBIR) and has undergone a substantial amount of refinement since it was introduced by Asada and Brady (1986). The authors in (Berrada et al., 2011) and (Bober, 2001) have clearly described the CSS algorithm. CSS divides the shape into convex and concave segments by identifying a set of inflection points – points where the curvature of the shape is zero. The CSS algorithm involves calculating the curvature of the contour while progressively smoothing down the curve and then finally generating the CSS image. Smoothing is done till the point when there are no zero curvature points left and the whole contour becomes convex. The CSS image is a curve on a 2 dimensional plane which plots all the zero crossing points. The horizontal axis contains the position of the points on the contour while the vertical axis contains the level of iteration / smoothing performed on the contour. A two dimensional vector is associated with each inflection (zero crossing) point expressed as (s,l) where s corresponds to the amount of smoothing applied while l is the position of the point on the contour curve. To compute l, the distance from an arbitrary chosen point to the inflection point on the contour is taken, while moving clockwise on the contour. The positions of the inflection points mapped onto the CSS image are the indices of these positions, which are peak values. Mokhtarian et al. (1996) were the earliest to improve the CSS algorithm. They proposed a new way for extracting the feature vector by taking into account the maxima of the zero crossing points on the contour. They proposed a fast indexing method using the aspect ratio of the CSS image along with eccentricity and circularity of the contour.

4) Shape Context Descriptor (SCD): Shape context was used as a novel shape descriptor for measuring shape similarity by Belongie et al. (2002). The main idea of shape context is to find the correspondence between the two shapes and find out the dissimilarity measure between the two shapes. To find the correspondence between the two shapes, N points are sampled from the contour of the shape and a reference point is fixed. The points are sampled using an edge detector algorithm. Then a set of vectors are computed originating from the reference point to all the other sampled points. The shape context for each point P_i is defined as a histogram $h_i(k)$ of the relative polar coordinates of the remaining sampled points.

$$h_i(k) = \#\{Q \neq P_i : (Q - P_i) \in bin(k)\} \tag{2.2}$$

Relative to the reference point the distribution of the remaining points is computed. To find the correspondence between the two shapes the method finds for each sample point on one shape a sample point on the other shape for which the displacement between the histogram values is the least. After the correspondence of points has been calculated, the shapes are mapped onto each other using alignment transformation estimations. Belongie et al. (2002) have used the regularized Thin Plate Spline (TPS) for representing flexible coordinate transformations. This transformation idea was derived from the classical work by Thompson (1942) in which he explained that the similarity of two similar biological forms can be measured by simple mathematical transformation between corresponding features. The dissimilarity between the two shapes is found by summing up the matching errors between corresponding points. After the dissimilarity measurements, object recognition is performed using nearest neighbour techniques. Shape context is a very robust shape descriptor which is highly discriminative. Besides being highly discriminative, shape context is transformation invariant and robust against shape variations, and has few outliers (Yang et al., 2008).

2.1.2.2 ***Region Based Descriptors:*** Region based descriptors (RBD) take into account the boundary as well as the internal information of the image and are more robust against noise and other shape variations. Most popular RBDs are Zernike moments descriptor (ZMD), Angular Radial Transform (ART), and Scale Invariance Feature Transform (SIFT) descriptor.

1) Zernike Moments Descriptor (ZMD): The applications of Zernike moments as a 2D shape descriptor in image analysis began with the pioneering work by Teague (1980). The Zernike Moments Descriptor is one of the most commonly used region based descriptors and has undergone constant improvement since it was first introduced. Zernike moments are continuous orthogonal moments derived from Zernike polynomials, a ground-breaking work by the Nobel Laureate Fritz Zernike (Zernike, 1934). Liao and Pawlak (1998) provided a very clear definition and explanation of Zernike moments. In order to understand Zernike moments, it is important to understand the concept behind Zernike functions. Liao and Pawlak (1998) defined Zernike functions as: "A set of complex orthogonal functions with a simple rotational property which forms a complete orthogonal basis over the class of square integral functions defined over the unit disk."

Celebi and Aslandogan (2005) pointed out the following advantages and disadvantages of Zernike moments. The advantages are:

a) Rotation Invariance

b) Robustness against small changes in shape

c) Insensitive to noise

d) Expressiveness: There is minimum information redundancy as the basis are orthogonal

The disadvantages are:

a) Coordinate space normalization: The image coordinate space must be transformed to the domain where the orthogonal polynomial is defined (unit circle for the Zernike polynomial).

b) Discrete approximation of continuous integrals: The continuous form of the Zernike moments must be approximated to discrete form. This approximation leads to errors in the computations as investigated by Liao and Pawlak (1998).

As the order of the Zernike moments increases, the computational complexity increases accordingly.

2) Scale Invariant Feature Transform (SIFT) Descriptor: The research into the SIFT algorithm began with the pioneer work by Lowe (1999). Lowe observed that image recognition should involve extracting local features which should be invariant to transformation, occlusion, illumination and affine transformations and highly distinctive in nature. Lowe based his approach on the observations and contribution from Schmid and Mohr (1997) that efficient object recognition can be achieved by using local image descriptors that can be sampled at a large number of repeatable locations. The SIFT algorithm converts the image into a huge collection of location feature vectors that are scale, rotation and translation invariant. The first step of the algorithm is to extract the scale invariant features in the image using the staged feature approach. Lowe used a scale space analysis approach and Gaussian kernels because it allows high efficiency and rotation invariance. The feature vectors extracted from the image are called SIFT keys. SIFT keys are then used for indexing for identifying candidate object models by using a nearest neighbour algorithm. The advantages of the SIFT algorithm are:

a) Invariant to scale, rotation and translation.
b) Partially invariant to illumination changes.
c) Robust against occluded objects.
d) Robust against object degeneracies.
e) Insensitive to noise.

3) Angular Radial Transform (ART) Descriptor: Angular Radial Transform (ART) is a popular region based descriptor which is used in the MPEG-7 standard (Bober, 2001). It was concluded by Core Experiments that among the descriptors, the overall performance of ART descriptor is the best for region-based similarity (Bober, 2001). ART is defined by Bober et al. as “the orthogonal unitary transform defined on a unit disk that consists of the complete orthogonal sinusoidal basis functions in polar coordinates” (Bober, 2001). Mathematically the ART coefficients are determined by:

$$F_{nm} = \int_0^{2\pi} \int_0^1 V_{nm}(\rho, \theta), f(\rho, \theta)\rho d\rho d\theta \qquad (2.3)$$

where F_{nm} is an ART coefficient of order n and m, $f(\rho, \theta)$ is an image function in polar coordinates, and $V_{nm}(\rho, \theta)$ is an ART basis function that are separable along the angular and radial directions, i.e,

$$V_{nm}(\rho, \theta) = A_m(\rho) R_n(\theta) \tag{2.4}$$

In order to achieve rotation invariance, an exponential function is used for the angular basis function,

$$A_m(\theta) = \frac{1}{2\pi} \exp(jm\theta) \tag{2.5}$$

where j refers to the jth projection of the basis function on the image.

2.1.2.3 ***Hybrid 2D Shape Descriptors:*** Wei et al. (2009) proposed the two-component solution (TCS), in which centroid distances, contour curvature and Zernike moments were selected as the shape features, while a two-component strategy was applied in feature matching. They used their approach in trademark image retrieval (TIR). Pooja (2012) proposed a hybrid approach which combines Fourier descriptor with ART. Fourier descriptor is used to extract local features while ART is used to extract global features. The authors performed experiments with FD + ART and FD + ZMD combinations. They showed that their techniques perform better than the TCS technique.

2.1.3 3D Shape Descriptors

3D shape descriptors have been extensively used by researchers since the 1990's in 3D search engines and sketch-based modelling systems. Tangelder and Veltkamp (2008) and Zhang et al. (2007) have published comprehensive surveys of 3D SDs. Table 2 shows a classification of 3D SDs.

Table 2: Classification of 3D shape descriptors.

View Based	Adaptive Views Clustering
	Compact Multi-View Descriptor
	LightField Descriptor (LFD)
Histogram Based	Shape Spectrum
	Generalized shape distributions
	Bag-of-Features (BoF)
Transform Based	Spherical Harmonics Descriptor
	PCA Spherical Harmonics Trns.
	Spherical Trace Transform
Graph Based	Skeletal Graph Based
	Reeb Graph Based
Hybrid 3D Descriptors	CMVD + STT
	Depth-Buffer + Silhouette + REXT
	SIFT + Bag-of-Features
	Depth-Buffer + Spherical Harmonics

2.1.3.1 ***View Based Descriptors (VBD):*** View based descriptors (VBD) use silhouette, greyscale or depth-buffer images extracted from multiple views of 3D objects. According to Liu (2012), the advantages of a VBD are twofold: 1) It does not require the explicit virtual model information, which makes the method robust to real practical applications. 2) The view-based 3D model analysis methods can benefit from existing image processing technologies which has been studied for several decades. These views are then used in the second step to retrieve 3D models from the database which best match the photos and is referred to as the clustering problem by K-means. The smaller set is found using Bayesian Information Criteria (BIC) which gives scores to the sets and the one with the highest score is the most optimal characteristic views set. The third step employs Bayesian Probability distributions for indexing and retrieving 3D models based on the input photos. The authors found that the Light Field Descriptor (LFD) algorithm is second best in terms of retrieval accuracy and performance.

1) Compact Multi-View Descriptor (CMVD): Daras and Axenopoulos (2009, 2010) have presented a new method for 3D shape retrieval called Compact Multi-View Descriptor (CMVD). Their method accepts multi-modal queries (2D images, sketches, and 3D models). The first step of the method is pose estimation in which Principal Component Analysis (PCA) and Visual Contact Area (VCA) methods are used to estimate the pose and apply rotation invariance to the 3D object. The next step is to generate 24 sets of 2D image views of the 3D object from 18 different viewpoints of a 32-hedron surrounding the 3D object, by rotating the object 24 times in 90 degrees along the three principal axes. Two types of views are extracted, binary views (silhouette views) and depth views. After extraction of the 2D views, the method generates 2D shape descriptors of the extracted views by applying three rotation invariant 2D functionals namely 2D Polar Fourier Transforms, 2D Zernike Moments, and Krawtchouk Moments (Yap et al., 2003). The final step, 3D to 3D matching is achieved by computing the total dissimilarity of the extracted 2D views of the two 3D objects. 2D to 3D matching is achieved by computing the dissimilarity of the 2D image query and the extracted 2D view of the 3D object that is the most similar to the 2D query image. According to the authors, the CMVD method when combined with the Spherical Trace Transform (STT) algorithm outperforms other well-known algorithms.

2) Light-Field Descriptor (LFD): The Light-field Descriptor (LFD) was introduced by Chen et al. (2003), which is based on the idea that two 3D objects are similar if they look similar from all viewing angles. In their approach, ten silhouette images are taken from 10 viewing angles distributed evenly on a dodecahedron. To extract the features of the silhouette images, they used Zernike moments and Fourier descriptors. To calculate the dissimilarity, they found the minimal dissimilarity obtained from rotating the viewing spheres of one lightfield descriptor relative to the other lightfield descriptor. Experiments indicate that LFD performs better than Spherical Harmonics Descriptors (Funkhouser et al., 2003). Advantages of LFD include invariant to translation, scale and rotation, and robust against noise and degeneracies.

2.1.3.2 ***Histogram Based Descriptors (HBD):*** Histogram based descriptors collect the features of a 3D shape in numerical values in bins defined over the feature domain.

1) 3D Shape Spectrum Descriptor (3D SSD): A 3D shape spectrum descriptor (3D SSD) is a shape descriptor which contains a shape index distributed over the entire mesh (Zaharia and Preteux, 2001). A shape index is defined as a local geometric feature of the shape expressed as the angular coordinate of the polar representation of the principal curvature vector. Regarding the original feature the shape index is invariant to scale and Euclidean transform, and it represents by salient elementary shapes (convex, concave, rut, ridge, saddle, and so on). 3D SSD locally characterizes free form discrete polygon 3D meshes. 3D SSD possesses the characteristics of: (1) generality, since 3D meshes may include open surfaces that do not have an associated volume; (2) invariance to scale and Euclidean transforms; (3) robustness that different triangulations of the same object are permitted and it successfully retrieves articulated objects with different postures. Since this descriptor is a simple local feature representation, it should be combined with some global representation schemes.

2) Generalized Shape Distributions (GSD): Liu et al. (2006) proposed a novel technique for 3D shape retrieval called the Generalized Shape Distributions (GSD), which is a 3D histogram. GSD is based on local and global shape signatures / descriptors of a 3D model. Before generating a GSD histogram, there are some preliminary steps, which involves the generation of a dictionary of local shape descriptors / signatures using the Spin Images approach. 50,000 points are sampled on the surface of the 3D shape and these sampled points are then accumulated to create spin images. These spin images are then clustered into 1500 clusters using the k-means algorithm. Then each spin image is assigned an index based on the index of its nearest cluster. After these steps, GSD representation is generated. The GSD histogram has three dimensions. The first dimension stores the Euclidean distance of the 2 point pairs, while the other two dimensions store the index value of the two points. This technique has proved to be more accurate and efficient than its predecessor technique 'Shape distributions' and Bag-of-Features technique (Osada et al., 2002).

3) Bag-of-Features Histogram: The Bag-of-Features (BoF) technique is a method of accumulating the visual features of a 3D model in a histogram where thousands of visual features are extracted from range images thus improving retrieval efficiency. Moreover Bag-of-Features has proved to be robust against articulated or non-rigid 3D models.

2.1.3.3 ***Transform Based Descriptors (TBD):*** The Princeton Group and Konstanz Group have undertaken considerable research on transform based descriptors (TBD). The theoretical foundations of TBD are in classical processing such as spherical harmonics, and Fourier transform. Usually, for pose normalization Principal Component Analysis (PCA) is applied as a preliminary step in TBDs (Vranić et al., 2001).

1) Spherical Harmonics Descriptor (SHD): Funkhouser et al. (2003) were the first researchers to use spherical harmonics to describe a 3D shape, and considerable work has been done in this domain following that. With this descriptor, the 3D model is first voxelized with each voxel having either a 1 or 0 value. The binary voxel grid of the 3D model is then placed under concentric spheres and decomposed into spherical functions. Next, a set of harmonic functions are computed from each concentric sphere which are rotation invariant. Each harmonic function is represented as a histogram called the spherical signature. The spherical signatures are then combined to generate a rotation invariant 3D shape descriptor. To compare two spherical harmonics descriptors the authors used Euclidean distance. The disadvantage of SHD is that a 3D model cannot be reconstructed from the feature vector. In (Funkhouser et al., 2003), Funkhouser et al. avoided the PCA step because they believe that PCA is an unstable approach for pose normalization.

2) PCA Spherical Harmonics Transform: The independent work on spherical harmonic based descriptors carried out by the Konstanz University group is in parallel with the Princeton University group. There has been a debate around whether to use PCA for pose normalization or not. Vranić et al. (2001) and Vranic (2003) proposed another spherical based shape descriptor that uses PCA as its pose estimation step and claimed that this descriptor has outperformed the spherical harmonic descriptor proposed by the Princeton Group. Vranic's spherical harmonics descriptor (Vranić et al., 2001) is different from that of Princeton Group in a way that this descriptor involves a generalized PCA step for pose estimation not only considering the vectors and coordinate axes, but also all the points on the mesh with equal weights. For feature extraction, Fourier transform is applied after the spherical harmonics functions are computed which finally generates a feature vector. According to Vranić et al. (2001), the PCA approach is slightly more expensive but more accurate than the original approach.

3) Spherical Trace Transform Descriptor (STTD): The STTD proposed by Daras et al. (2004) is an extension of the 'Trace Transform' with is actually a generalization of 'Radon Transform'. STTD does not employ PCA as its preliminary step but it uses rotation invariant spherical functions to produce a completely rotation invariant shape descriptor. The first step which is the pre-processing stage is to achieve translation and scaling normalization. In this stage the 3D model is placed inside a bounding cube and the cube is partitioned in equal cube shaped voxels. Voxelization allows us to achieve translation and scale normalization. Then, a set of initial 2D functions are applied to the model which creates a set of concentric spheres. After applying initial functions a set of spherical functions are applied to produce the final descriptor vector. For similarity matching, weights are assigned to each descriptor to achieve effectiveness in shape retrieval.

2.1.3.4 ***Graph Based Descriptors (GBD):*** GBDs tend to represent the topology of a 3D shape in the form of a graph or a tree structure. These descriptors are easy to compute. However they are not computationally efficient. According to Iyer et al. (2005), an advantage of GBDs is that they allow representation at multiple levels of detail and facilitate matching of local geometry.

1) Skeletal Graph Descriptors: Skeletal graph-based techniques utilize a skeletal graph of a shape as its shape descriptor by computing the 'skeleton' for a model. Blum proposed the concept of a skeleton in (Blum, 1967). A skeleton in 2D is the medial axis, while in 3D it is the medial surface. Several methods have been proposed to perform Skeletonization such as distance transform (Borgefors, 1984), thinning (Lam et al., 1992), or Voronoi-based methods (Ogniewicz and Kübler, 1995). Additionally, curve skeletonization methods have been proposed to convert a 3D model into a medial axis type representation (Sanniti et al., 2002). The skeletal graph stores various entities obtained after skeletonization in a graph data structure. An advantage of skeletal graph-based methods is that they are topology preserving. Hence, they can be used for subgraph isomorphism at a very low computational cost. Additionally, local part attributes can be stored for a more accurate comparison.

2) Reeb Graph Descriptors: Reeb (1946) defined a skeleton structure, called the Reeb graph, which is determined using a continuous scalar function on an object. Three types of scalar functions have been used namely height function, curvature function, and geodesic distance. Geodesic distance has been used in many applications because it provides invariance against rotation and robustness against noise and small perturbations. The function is integrated over the whole body to make it invariant to the starting point and is also normalized to achieve scale invariance.

2.1.3.5 ***Hybrid 3D Shape Descriptors:*** In recent years, researchers in the field of 3D object retrieval have combined several algorithms to produce hybrid algorithms in order to improve the quality of 3D shape retrieval and analysis. Vranić (2004) proposed one of the earliest hybrid algorithms, and indicated that the cross breeding of the depth-buffer, silhouette, and REXT descriptor (Depth-Buffer + Silhouette + REXT) has superior performance over all other state-of-the-art descriptors at the time.

1) CMVD + STT: Daras and Axenopoulos (2009) demonstrated that Compact Multi-View Descriptor (CMVD) when combined with Spherical Trace Transform (STT) performs better than all the algorithms proposed in the past. They tested their hybrid algorithm on the three databases i.e. Princeton Shape Benchmark (PSB) (Shilane et al., 2004), ITI database used in Victory 3D Search Engine (Daras et al., 2008), and the Engineering Shape Benchmark (ESB) (Jayanti et al., 2006). The authors compared their hybrid algorithm (CMVD + STT) with three other successful algorithms 1) The LightField Descriptor, 2) SIFT + Bag-of-Features (discussed next), and 3) Depth-Buffer + Silhouette + REXT (DSR). The precision-recall results have shown that the proposed hybrid algorithm performs better than all other algorithms.

2) SIFT + Bag-of-Features (BF-SIFT): Ohbuchi et al. (2008) proposed a hybrid descriptor based on extracting local visual features of a 3D model using the SIFT algorithm and efficiently integrating them in a histogram using the Bag-of-Features approach. In their algorithm, several 2D range images are obtained from the 3D model. Then, the SIFT algorithm is used to extract local features. Each feature is a vector quantized using a visual codebook. K-means learning is used to cluster the local features into a bag of visual words. Then a histogram is generated using the frequencies of visual words, which acts as the feature vector for the 3D model. Some advantages of BF-SIFT are 1) suitable for articulated models, 2) high discriminative power, 3) suitable for 2D image and sketch-based queries, and 4) effective for partial matching.

3) ***Depth-Buffer + Spherical Harmonics:*** Papadakis et al. (2008) proposed a hybrid descriptor which is composed of the depth buffer algorithm for extracting 2D features and spherical harmonics for encoding 3D features. For pose normalization two alignment methods namely CPCA and NPCA are used while compactness of the feature vector is supported via scalar feature quantization to a set of values that is further compressed using Huffman coding. The authors have demonstrated superior performance of the proposed retrieval technique through an extensive comparison against state-of-the-art methods such as LFD and DSR on standard datasets including Princeton Shape Dataset, National Taiwan University Benchmark, CCCC, MPEG-7, and Engineering Shape Benchmark.

2.1.4 SHREC - 3D Shape Retrieval Contest

SHREC is an international shape retrieval contest started in 2006, which aims to compare and evaluate 3D shape retrieval algorithms using a benchmark. In 2012 the performance of five hybrid algorithms were compared on a generic shape benchmark (SHREC'12) devised from other well-known benchmarks including PSB, NTU, and CCCC (Godil et al., 2012). These algorithms include Local Shape Distribution descriptor, ZFDR (hybrid descriptor based on Zernike, Fourier, depth-buffer, and ray based descriptors) (Li and Johan, 2011), 3D Spatial Pyramid descriptor, Dense Voxel Spectrum descriptor (DVD) (Tatsuma and Aono, 2009), and Dense Grid SIFT (DGSIFT) descriptor (Ohbuchi and Furuya, 2010). The precision-recall results showed that DGSIFT performs the best amongst others followed by DVD and ZFDR respectively.

2.1.5 Comparison & Discussion

Yang et al. (2007) and Yang et al. (2008) have compared the computational complexity of all the major categories of 3D shape descriptors in order to assess their performances. According to their findings, transform based methods have computational complexity as $O(b N^3)$ where b is the number of spherical functions used, and N is the number of voxels in each axis. As compared to these, histogram based methods are faster with $O(t \text{ NlogN})$ complexity, where *t* is the number of histogram bins and N is the number of mesh vertices in each axis. However, transform based methods have higher discriminative power than histogram based methods. Graph based methods capture shape features very well however they are computationally expensive with O(NlogN) complexity. Histogram based descriptors are compact, fast, very robust, and are therefore suitable for hybrid descriptors. Such an example is the Bag-of-Features (BoF) approach.

2.1.6 Conclusion & Future Research Directions

Over the past few years, the research in content-based 3D model retrieval has moved towards inventing and utilizing hybrid descriptors, and dealing with objects that are non-rigid or articulated. 3D model retrieval is still in its infancy and much work needs to be done to improve the accuracy of the techniques. Moreover, semantic based retrieval has also become a hot topic amongst researchers in recent times. Hybrid techniques have proved to be more successful than the original individual techniques.

Researchers have found that region-based 2D shape descriptors are great tools for local feature matching in 3D shape retrieval. Amongst the 3D SDs, graph based algorithms are the least preferred algorithms because of their performance inefficiency. View-based and histogram-based algorithms are the most favourite amongst the researchers due to their decency in accuracy and performance.

Much debate has happened between global feature-based matching and local feature-based matching. Global feature-based methods take the whole geometry into consideration and produce a feature vector describing the whole shape. Thus they are not suitable for local or partial matching. However these are easy and faster to compute. On the other hand, techniques like SIFT have paved a way for robust matching of local features.

Bustos et al. (2007) identified that there is currently a lack of databases dedicated to domain specific models such as models of human characters of several levels of detail. This could be a very important future direction for benchmark databases. Moreover, techniques to retrieve articulated human character models (models in different poses) also need to be further investigated.

In surveys by Yang et al. (2008) and Yang et al. (2004, 2007), the authors have identified the following areas that deserve future investigation.

a) 3D models these days are available in different formats however there is a lack of algorithms that extracts features directly the compressed format of the 3D models. Partial matching of 3D objects that utilize the local feature of a model at different resolutions requires different kinds of feature vectors.

b) Graph based techniques need to be improved in terms of computation complexity and efficiency. Graph based techniques can describe an object very accurately on several levels of detail. However current approaches take topology of the mesh into account which is computationally impractical. Moreover graph comparison is proportional to graph-size.

c) Non-shape features of a model such as colour, texture, and material should be investigated further.

d) The animation industry needs artist friendly and intuitive tools for creating 3D models and currently the interfaces are not very artist friendly.

e) Very few techniques exist that retrieve 3D objects from scenes containing multiple 3D objects. This requires a novel hierarchical object structure to be investigated to recognize 3D objects from cluttered scenes.

2.2 A Survey of Sketch-based Modelling Systems

3D modelling is an intricate and laborious process which requires considerable investment of time to master. Most skilled 3D artists generally possess years of modelling experience and thorough understanding of drawing skills (human anatomy, still life, environments, machinery, vehicles etc.) which allow them to create 3D models quickly and accurately. Researchers have long pondered on creating systems that would automate the steps followed by skilled 3D artists to generate detailed and realistic 3D models. Professional modelling packages such as Autodesk Maya (Maya), Pixologic ZBrush (Zbrush) and Autodesk Mudbox (Mudbox) are loaded with tools that make modelling easy, fun and swift for artists of intermediate to expert levels. However these packages have steep learning curves for beginners, purely work on 3D input data and do not support the interpretation of sketched lines and curves into 3D geometry thus limiting the scope of adding details to 3D models.

Sketch-based Modelling (SBM) is a very broad area within the field of Computer Science spanning several sub-fields such as Computational Geometry, 3D Shape

Retrieval, Computer Vision, Computer Graphics. Object Oriented Programming has considerable importance in SBM system. This is because an increasing number of companies are developing and adopting reusable software systems. With the increasing demand of reusable off-the-shelf SBM components, some companies are likely to develop large SBM systems made out of several reusable SBM components. Moreover there is also a demand of open-source SBM components, which can be beneficial for the computer graphics research community.

The interpretation of sketches to assist artists to create 3D models serves as the main idea behind efficient SBM systems. The research in SBM can be traced back to as early as the mid 1980's. Among the early works, John Canny published his seminal work in edge detection famously known as the Canny Edge Detection algorithm (Canny, 1986). This algorithm is used to extract feature curves from 2D images including silhouette and local feature curves. Researchers have explored many avenues in computational geometry, and computer vision to create powerful tools for 3D artists. Some of these notable algorithms are discussed in this survey.

Sketch-based Modelling has undergone a considerable amount of research over the past two decades with researchers investigating novel and innovative techniques at a rapid pace. So far researchers in the field of SBM have aimed their research on organic and inorganic modelling (architectural and mechanical) as stated by Rivers et al. (2010). Therefore, the research in SBM has been broadly divided into Organic Sketch-based Modelling and Inorganic Sketch-based Modelling. Inorganic SBM involves CAD/CAM systems, architectural / building modelling systems and computer aided automobile design system. Notable examples of inorganic SBM systems are Trimble SketchUp (Sketchup), (Schmidt et al., 2009), and more recently Rivers et al. (2010). On the other hand organic sketch-based modelling systems provide tools to create character models mainly using smooth feature curves, and suggestive contours (DeCarlo et al., 2003) etc. Organic modelling systems also provide tools for creating 3D models from simple primitives such as ellipsoids and using inflation techniques to inflate a closed 2D region / sketch (e.g. a circle, oval etc) such as Teddy (Igarashi et al., 1999), and FiberMesh (Nealen et al., 2007).

In order to effectively investigate and experiment with new techniques in SBM, it is very important to possess a basic understanding of how traditional artists draw sketches of human characters, and how 3D artists model human like characters. Studying the process of sketching and drawing by artists can be found in a study by Cole et al. (2008). Several excellent surveys are available on sketch-based modelling techniques including (Olsen et al., 2009), and (Cook and Agah, 2009).

Sketch-based modelling techniques are usually categorized into construction based and recognition based as indicated by Olsen et al (Olsen et al. 2009). Construction based techniques tend to generate a 3D model directly from 2D sketched feature curves while recognition based techniques uses 3D search engines to retrieve 3D models from the database which are similar to the 2D input sketches by applying shape descriptors to the input or matching feature curves to their counterparts in the 3D model database. A major problem faced when designing sketch-based modelling techniques is the reconstruction of 3D objects directly from very limited information. Olsen et al. (2009) have also concluded in their survey that a hybrid system that contains a substantial shape memory, robust creation rules, and perhaps even a capacity to learn new shapes, hold the most potential for approaching human-like sketch understanding.

Most artists require tools that could provide them the feeling of sketching on a canvas in a free hand manner. To address this fact, many SBM systems have been published and patented which give the users an experience of free hand sketching. In ILoveSketch (Bae et al., 2008), the authors have designed a curve sketching system that allows artists to draw strokes on a 3D canvas, and automatically approximate the strokes using NURBS. Several mouse inputs are mapped to navigation of the 3D canvas. Several gestures are provided for aligning the curves with other curves, scaling, rotation, and erasing curves.

2.2.1 SBM Pipeline

Generally speaking, SBM systems follow a systematic pipeline (Olsen et al., 2009), with each step employing several unique algorithms. The main steps in the pipeline are: 1) Sketch Acquisition, 2) Sketch Filtering, and 3) Sketch Interpretation.

2.2.1.1 Sketch acquisition: Sketch acquisition involves making choices about the medium and input devices for acquiring and drawing a sketch. Recent techniques have supported a pen as an input medium and drawing tablets as a drawing canvas (Bae et al., 2008). More haptic ways of 3D sketching have also emerged such as SPACESKETCH (Nam and Chai, 2012). SPACESKETCH is a unique application that performs shape modelling by using 2 space wands in a 3D environment. By using a 3DTVdisplay shown in Figure 2, sketching an object in a stereoscopic space is made possible.

2.2.1.2 Sketch filtering: Before a given sketch can undergo processing to be promoted to a 3D model, most state-of-the-art techniques perform pre-processing on the input curves or strokes. This step is necessary to refine the input curves to be ready for processing. Denoising is an important class of algorithms for sketch refinement being employed heavily in SBM systems.

2.2.1.3 Sketch Interpretation: Sketch interpretation involves the transformation of 2D sketched contours into a 3D surface mesh. This is the main component of an SBM system which includes several algorithms for the generation of 3D geometry from input strokes.

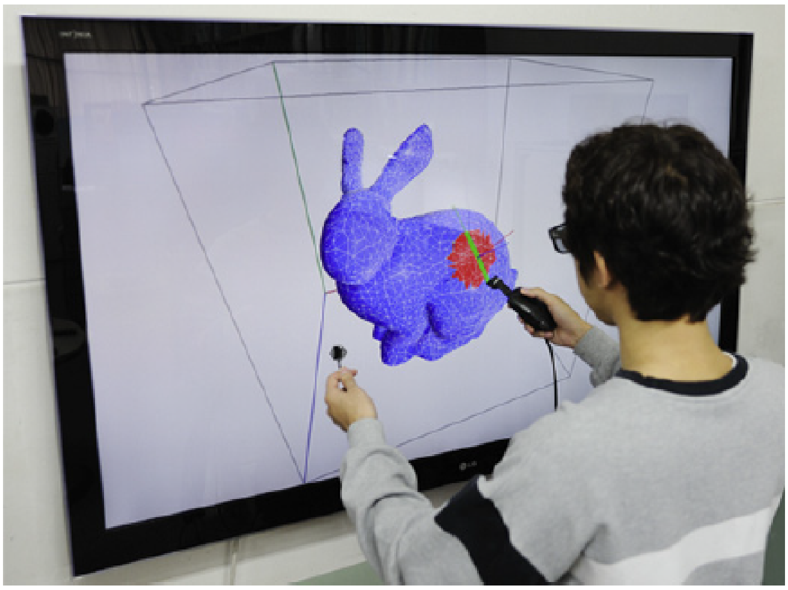


Figure 2: 3D Sketching with SPACESKETCH. *Reproduced from (Nam and Chai, 2012).*

In this survey, the techniques are categorized into the following major classes. 1) Single View systems, 2) Multi-View Systems, 3) Curve Networks Based Systems, 4) Data Driven Systems

2.2.2 Single View Systems

Single view SBM systems allow the artists to create models using just one sketch. These systems make use of algorithms to create surfaces which give an appearance of different parts of the human body, such as an ellipsoid which is an inflated silhouette of a 2D ellipse.

2.2.2.1 Teddy: a sketching interface for 3D freeform design: Igarashi et al. (1999), pioneered the research in inflated geometrical surfaces to create simple organic toy like character models. Their interface is designed to quickly and easily model freeform surfaces for creating stuffed animals or other organic models. The system includes tools to add 3D geometry to the model when the user sketches the outline. The algorithm first finds the spine of the silhouette by using the chordal axes introduced in (Prasad, 1997).

The system then wraps the spine with the polygonal mesh and then uses a constrained Delaunay triangulation of the polygon. Pruning of insignificant branches is performed using (Prasad, 1997). The algorithm progresses through several refinement steps to obtain a smooth and symmetric 3D surface. Their system also provides tools for extruding and cutting of the mesh. Extrusion on the 3D surface is implemented using a sweeping algorithm. Some disadvantages of the Teddy system are: 1) it does not accept complex and unexpected strokes such as T-junction strokes and cusps. 2) The features provided are limited since it was not suited to modelling complex and production ready models. In Figure 3, note that where the neck contour passes behind the chin, a T shape can be seen in the projected contour (called a T-junction), and the chin contour ends abruptly (called a cusp). T-junctions and cusps indicate the presence of a hidden contour; Williams and Hanson (1994) has proposed a method for using these to infer hidden contour lines in an image.

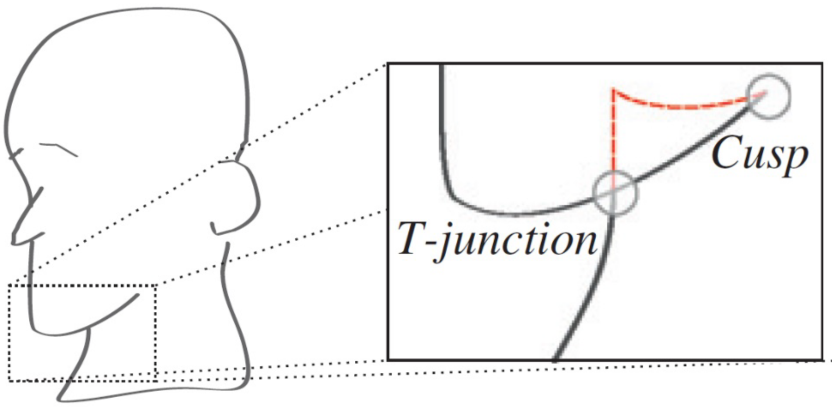


Figure 3: T-junction and cusp. *Reproduced from (Olsen et al., 2009).*

2.2.2.2 SmoothSketch: 3D free-form shapes from complex sketches: Several researchers have proposed improvements to the Teddy system. Karpenko & Hughes (2006) have addressed the problem of T-junctions and cusps by introducing an efficient algorithm extended from the work by Williams and Hanson (1994) for inferring shapes with hidden contours, on figural completion of hidden contours containing T-junctions to contours containing cusps as well. It should be noted that SmoothSketch does not offer a complete SBM system; instead this work can be used as a component to be embedded in SBM systems. Williams' work (Williams, 1994) included three steps to find the 3D surface to fit the given contours. These were 1) Drawing completion by inferring hidden contours, 2) Map the contours to the abstract topological surface and map the surface to $\mathbf{R}^2$, and 3) Lift this mapping to a smooth surface in $\mathbf{R}^3$, whose projection is the mapping to $\mathbf{R}^2$.

2.2.2.3 FiberMesh: designing freeform surfaces with 3D curves: Nealen et al. (2007) have introduced a system called FiberMesh, which is based on Teddy, but overcomes some of its limitations. Unlike Teddy, the user-drawn strokes stay on the model surface and serve as handles for controlling the geometry. The user can add, remove, and deform these control curves easily, as if working with a 2D line drawing. The curves can have arbitrary topology; they need not be connected to each other. For a given set of curves, the system automatically constructs a smooth surface by applying functional optimization. The system also provides tools for model deformation such as adding ridges and creases to the 3D surface, and also cutting, extruding, pulling and tunnelling the geometry of the model using simple mouse movements. Figure 4 shows all the operations that FiberMesh offers to be performed on a mesh.

FiberMesh is a research bred SBM system that can be utilized for modelling organic models. An advantage of its interface is that the user does not need to worry about the topology of the curves. Traditional methods require the user to cover the entire surface with triangle or quad regions. In FiberMesh curves need not be connected to other curves and much fewer curves can represent simple geometry. It is also important that, instead of providing individual points as an interface, the interface treats curves as continuous entities. Its main disadvantage is that it is not suitable for creating models that are ready for animation production, also known as 'production ready 3D models'. These production ready 3D models can instead be achieved with professional modelling packages such Autodesk Maya (Maya) or Pixologic ZBrush (ZBrush).

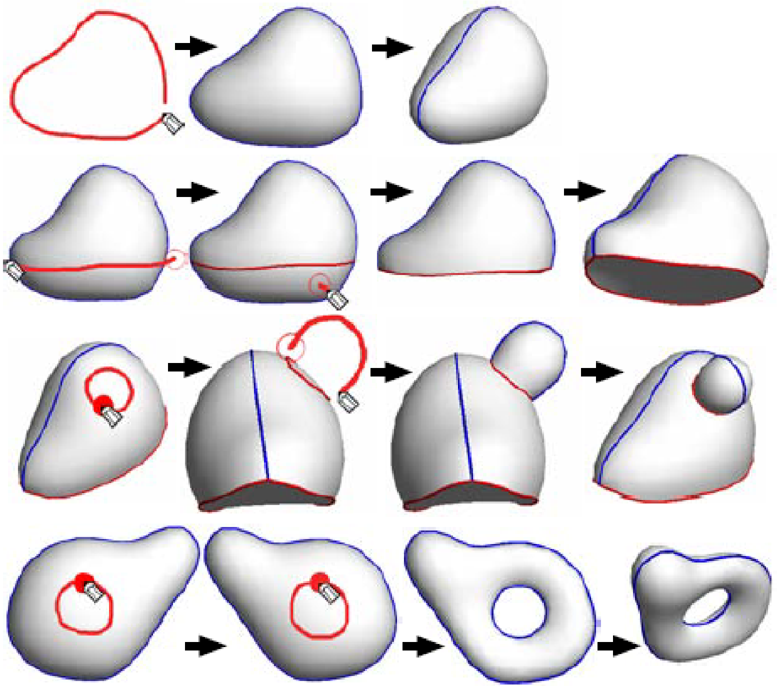


Figure 4: Sketching operations in FiberMesh (from top to bottom): creation, cut, extrusion and tunnel. *Reproduced from (Nealen at al., 2007).*

2.2.2.4 Structured annotations for 2D-to-3D modelling: Gingold et al. (2009) have used generalized cylinder and ellipsoids for modelling of organic models directly from a single view model. The authors have used geometric primitives: Generalized Cylinder, and Ellipsoid to quickly model a simple low-polygon model. Moreover the authors have provided annotations to make it easier for the authors to create plausible character with easy to use tools. The annotations—same-lengths and angles, alignment, mirror symmetry, and connection curves—allow the user to communicate higher level semantic information; through them their system builds a consistent model even in cases where the original image is inconsistent.. Figure 5 shows an overview of this approach. One disadvantage of this system is that it provides a limited set of tools to the artist for modelling and only provide two primitives (cylinder and ellipsoid). Most artists make use of more primitives to model a human character. One important primitive used by artists is the box primitive as demonstrated in a tutorial video by the leading comic artist Stan Lee (webfox100, 2009). In this tutorial, he tells us how easily one can decompose a human body into simple primitive geometric forms.

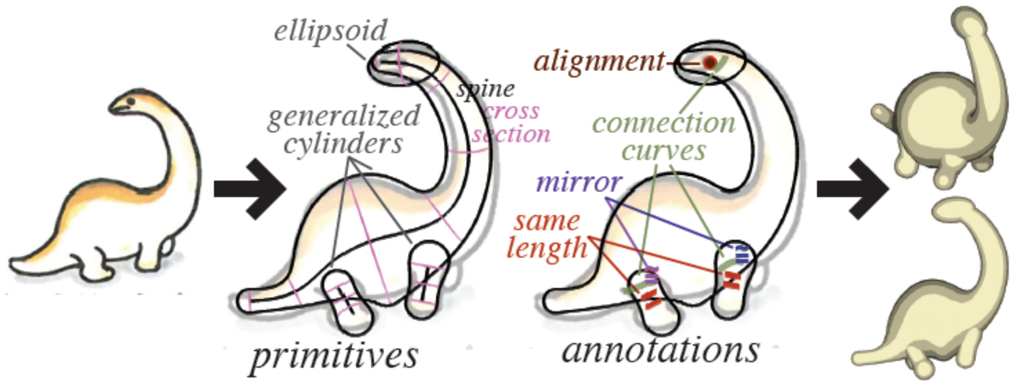


Figure 5: Modelling process of (Gingold et al., 2009). The user places primitives and annotations on an image resulting in a 3D model. *Reproduced from (Gingold et al., 2009).*

2.2.2.5 Automatic single-view character model reconstruction: Buchanan et al (2013) have demonstrated a system which generates low-polygon models directly from input character sketches without any interactivity by the artist. Their algorithm first computes a skeleton from the image and then draws cross section curves along the skeleton using a novel proposed heuristic. The overall benefit of the technique is to create a complete low-polygon base mesh model with the single click of a button, however since it does not provide enough interactivity to the user; this tool is not suitable for modelling high-resolution character models with well-defined feature curves. Their approach also demonstrates a shell-based meshing algorithm that allows for the ability to change the cross-sectional profile of the model based upon the type of character and even the type of limb in the model. Additionally, it contains a low complexity automatic skeletonization algorithm for raster images, with an optional user-controlled complexity parameter. The algorithm converts 2D outlines to 3D meshes using a heuristic that balances the skeletal relevancy of the mesh against reduced visual artefacts, using line style metrics to influence the 3D style of the generated mesh. By extracting a skeleton structure, approximating the 3D orientation and analysing line curvature properties, appropriate centre-points can be found around which to create cross-sectional slices and build the final triangle mesh. This technique can be useful for rapidly generating sub parts of the human character, such as limbs and torso as these parts are not very complicated anatomically. Figure 6 shows an overview of the process in (Buchanan et al., 2013).

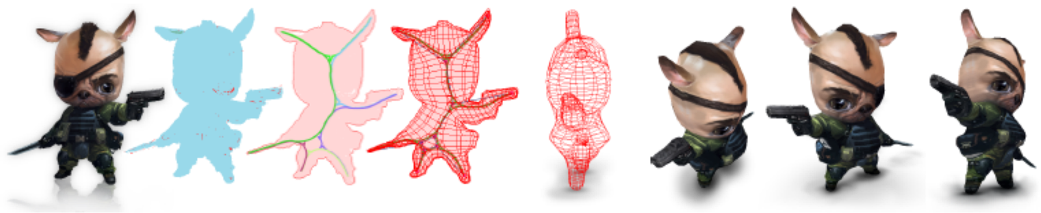


Figure 6: Overview of process in (Buchanan et al., 2013). *Reproduced from (Buchanan et al., 2013).*

2.2.2.6 Geosemantic Snapping for Sketch-Based Modelling: Shtof et al. (2013) have introduced a snapping technique which automatically reshapes and snaps simple 3D geometric primitives to 2D sketch primitives, then improves the model globally by inferring geosemantic constraints that link the different parts. The authors have used non-linear non-convex optimization techniques (Augmented Lagrangian Method) to accomplish the task. The system requires the user to manually place the 3D primitives to their appropriate places as humans are better at this task than computers. The computer performs the tedious and precise alignments and snapping in real-time. The automatic identification of geosemantic relationships between the primitives such as co-planarity and continuity is an attractive feature of this system. The snapping behaviour of this system can be utilized to model several complex muscles of the human character which otherwise can be a tedious task. Figure 7 demonstrates an overview of (Shtof et al., 2013).

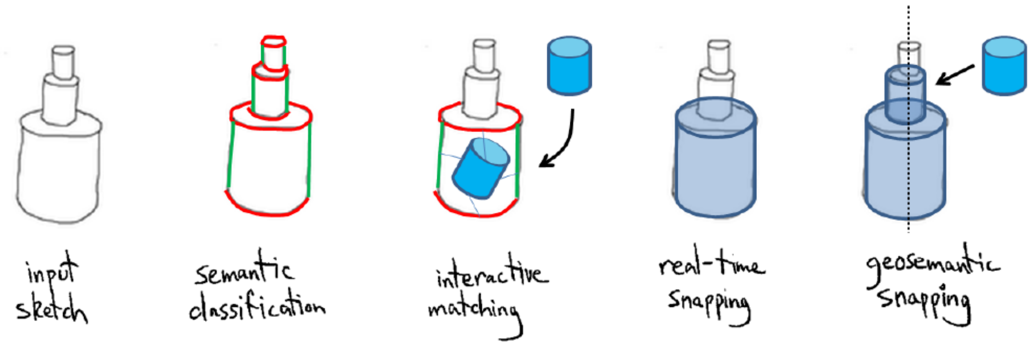


Figure 7: Overview of the process in (Shtof et al., 2013). *Reproduced from (Shtof et al., 2013).*

2.2.2.7 Single-view sketch-based modelling: Andre and Saito (2011) have proposed a single view SBM system which is also based on the sweeping algorithm (generalized cylinders) but requires the user to draw two outlines (cross section of the shape and a closed curve perpendicular to the cross section) and an optional silhouette. In return the system sweeps the cross section along the silhouette outline curve. The attractive feature of this system is that the user draws the shape in the desired pose and uses a single view. This system is an important contribution towards SBM as it provides a very easy interface to the artist where the artist merely draws the 3D models as if they are drawing on a piece of paper, as their approach mimics the way most professional artists draw cartoonish characters. Unfortunately the models created from this method are rough and when modelling complex shapes made up of several sub-shapes, do not maintain smooth topology from one sub-shape to another in an organic manner, and therefore this system alone is not suitable for organic modelling of characters. Figure 8 demonstrates an overview of the process of this approach.

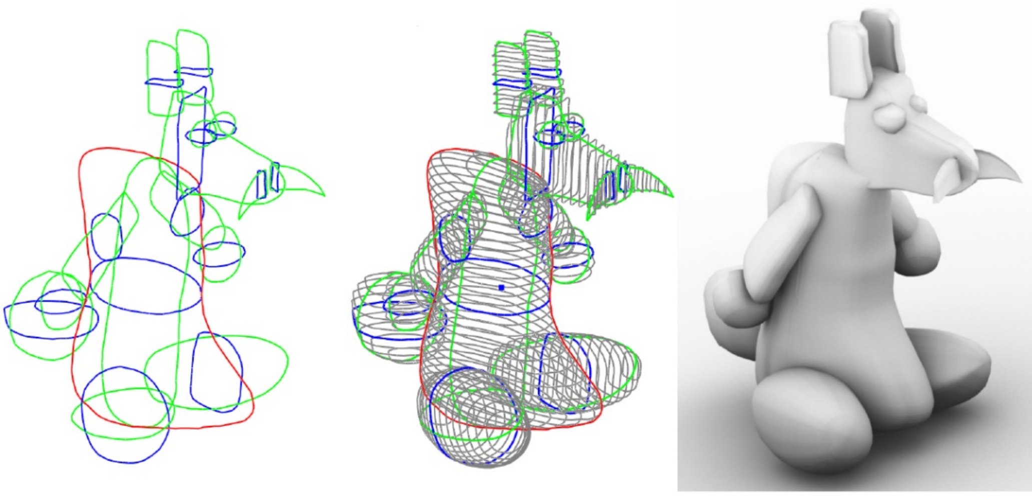


Figure 8: Overview of process in (Andre and Saito, 2011). *Reproduced from (Andre and Saito, 2011).*

2.2.2.8 3-Sweep: Extracting Editable Objects from a Single Photo: In (Chen et al., 2013), the authors have demonstrated a 3D model reconstruction technique from single photos, by automatically fitting the boundaries of a generalized cylinder to the boundaries of the subject in the image. It requires the users to first draw a profile curve of the generalized cylinder, which is essentially an ellipse in 3D space. The user then drags the mouse over the 2D shape and the system renders a generalized cylinder along the path swept by the mouse/pen and snaps it to the outline of the 2D shape. The algorithm involves computing the 2D shape, its projections and its relations to other shapes in the image using energy minimization methods, thus simulating the cognitive ability of humans. This system makes the daunting task of extracting objects simple, and snapping the 3D object to the 2D image outline. Thus this system turns out to be an intelligent and useful system for the research topic of sketch-based modelling of realistic character models. This system is focused towards modelling of objects made up of pipes and cylinder such as vases, telescopes, binoculars, pots and other similar solid body mechanical object, and does not suit organic modelling. This is an important limitation of this approach.

2.2.3 Multi-view systems

Several multi view 3D reconstruction techniques are available which transform a set of images taken from different views of a model into a complete 3D model such as (Lin et al. 2010), and Autodesk 123D catch (www.123dapp.com/catch).

2.2.3.1 3D modelling with silhouettes: Rivers et al. (2010) have proposed a new and simple algorithm towards computing the silhouette cylinders to compute the 3D Constructive Solid Geometry (CSG) by leveraging the special properties of silhouette cylinders. The main idea behind their approach is that the silhouettes of the sub-parts of a 2D drawing from two different views (front and side) can give enough information to create a 3D model. In their interface, a user specifies the silhouettes of a part from front, side, or top views. Once two or more silhouettes have been specified its 3D shape is automatically constructed. Although each part is axis-aligned, parts can be rotated in 3D relative to each other, such that the model as a whole need not be axis-aligned. The system targets the modelling of man-made objects, as they typically can be decomposed into axis-aligned subparts. Organic models are not well suited to this approach. The algorithm combines the parts of the 3D shape using Boolean operations. Figure 9 shows the overview of the process in (Rivers et al., 2010).

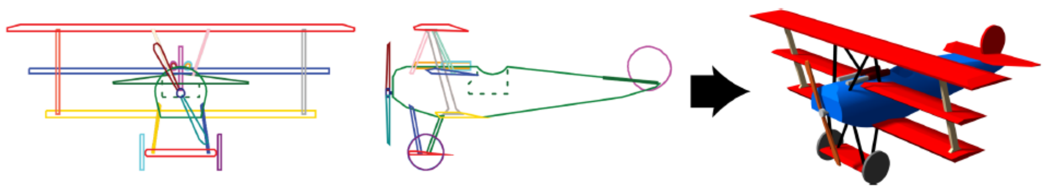


Figure 9: Overview of the process in (Rivers et al., 2010). *Reproduced from (Rivers et al., 2010).*

The algorithm implemented by the authors proceeds in the following steps: 1) Find the intersection of each silhouette cylinder c with the plane. 2) Label these intersection polygons i_c. 3) For each silhouette, also find the intersection of the surface of the silhouette cylinder with the plane (as opposed to the interior), labelled s_c which is a subset of i_c. 4) Apply 2D Boolean operations to the intersection and surface polygons corresponding to a set of silhouette cylinders to compute i and c for the solid resulting from a Boolean operation applied to the corresponding silhouette cylinders (Figure 10). In general, the authors have shown that Boolean operations can be computed for any two solids for which i and s, the interior and surface polygons, are known for every plane, to yield the i and s of the resulting solid for each plane, allowing further operations to be applied to the derived solid. For smoothing of the 3D models, the authors have adopted the smoothing algorithm from FiberMesh. This smoothing operation generates a surface that approximates the surface minimizing the variation of the Laplacian magnitude

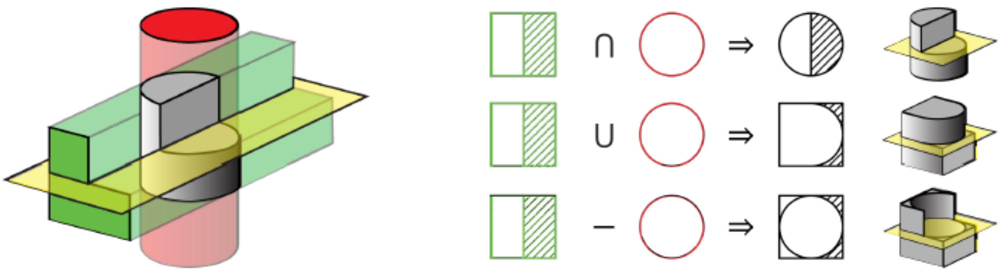


Figure 10: Modelling using Boolean operations. *Reproduced from (Rivers et al., 2010).*

2.2.4 Curve Networks Based Systems

Sketch-based modelling systems have benefited substantially with the 3D space available to the artists to draw strokes. A curve network as the name suggests is a network of strokes that the artist draws on a 2D canvas and they are transformed into 3D curves by underlying algorithms. In this section two systems are discussed that are based heavily on curve networks drawn in 3D space.

2.2.4.1 ILoveSketch: Bae et al. (2008) have developed a robust and feature rich system for 3D sketching. The salient features behind the design of this system are:

- Visually smooth complex curves for design exploration
- Minimal interruption to sketching by GUI and gestures
- Minimal set of gestures with intrinsic affordances
- Immediate and easy access to 2D/3D navigation
- Dynamic information display to assist 2D/3D sketching
- Focus on geometric objects rather than UI components

The basic "feel" of the system borrows from that of a physical paper sketchbook. This system provides designers with a virtual sketchbook with tools for smooth navigation of multiple canvases with interactions such as: tearing, peeling, panning, zooming, and rotation. The system also support automatic dynamic rotation of the virtual sketchbook based on the users' input strokes to make further multi-stroke sketching biomechanically comfortable. However according to the case study mentioned in the paper, one artist did not find the automatic dynamic rotation of view convenient, but found it rather distracting. Figure 11 shows some models created using (Bae et al., 2008).

The authors have provided five different 3D curve sketching methods along with the notion of sketchability – a view dependent scalar measure that helps determine how good a given viewing angle is for a given 3D curve sketching method. Sketchability-based automatic 3D rotation increases a designer's throughput, by reducing the need for explicit 3D navigation to find a suitable view in which to sketch. A minimal gesture set is provided for command input, and audio feedback is used to support gesture confirmation. As a whole, these methods result in a coherent 3D curve sketching workflow that does not rely on menus, icons, or tool palettes that could clutter the screen. Thus, as argued by the authors the user's focus of attention can stay on the artwork at all times. One limitation of this system is that it is not suitable for modelling organic character models as most of the models demonstrated using this system are mechanical in nature such as aircrafts, cars, and spaceships etc.

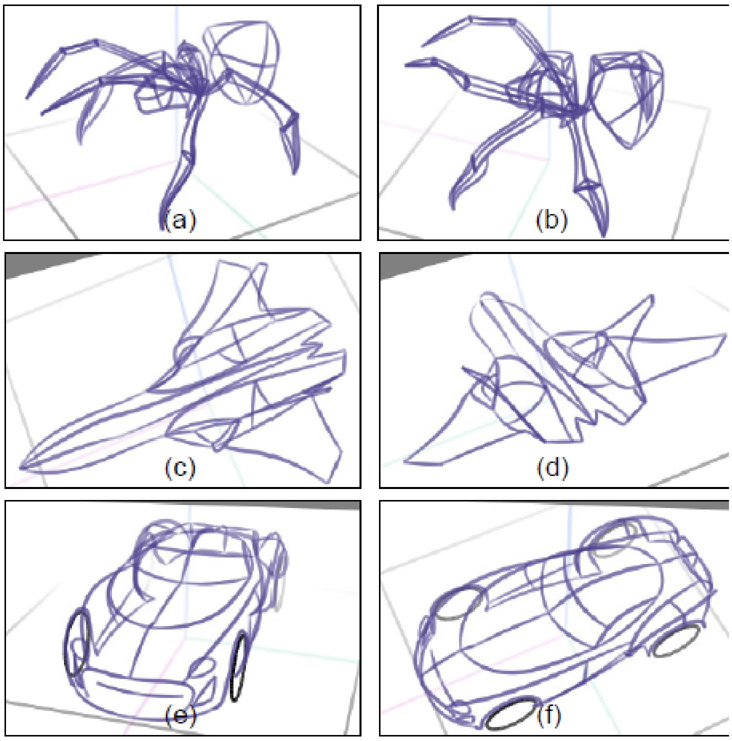


Figure 11: Models created using (Bae et al., 2008). *Reproduced from (Bae et al., 2008).*

2.2.4.2 JustDrawIt: Similar in spirit with the 'ILoveSketch' (Bae et al., 2008) system, Grimm & Joshi (2012) have proposed 'JustDrawIt'. It is a 3D curve sketching system based on existing and novel techniques. JustDrawIt provides an interface to the artists to draw 2D strokes and then the system converts the strokes into 3D curves. The system support "snapping" together curve networks and specifying normal in order to create consistent curves from which surface models can be generated. Figure 12 shows the modelling process of JustDrawIt.

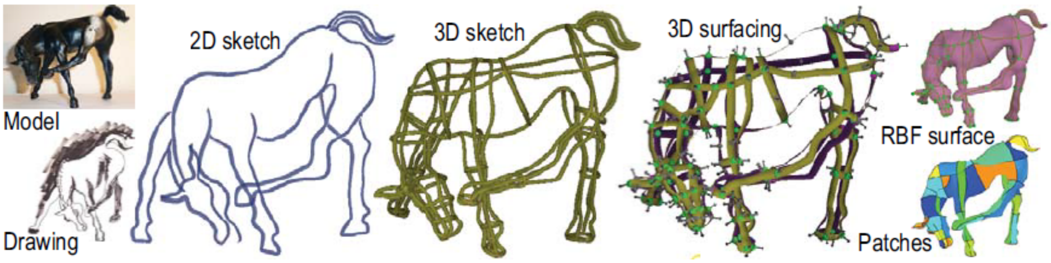


Figure 12: Modelling process of (Grimm and Joshi, 2012). *Reproduced from (Grimm and Joshi, 2012).*

At the core of this system lies the stroke inference engine (Figure 13), which infers the stroke, refines it and transform into curves.

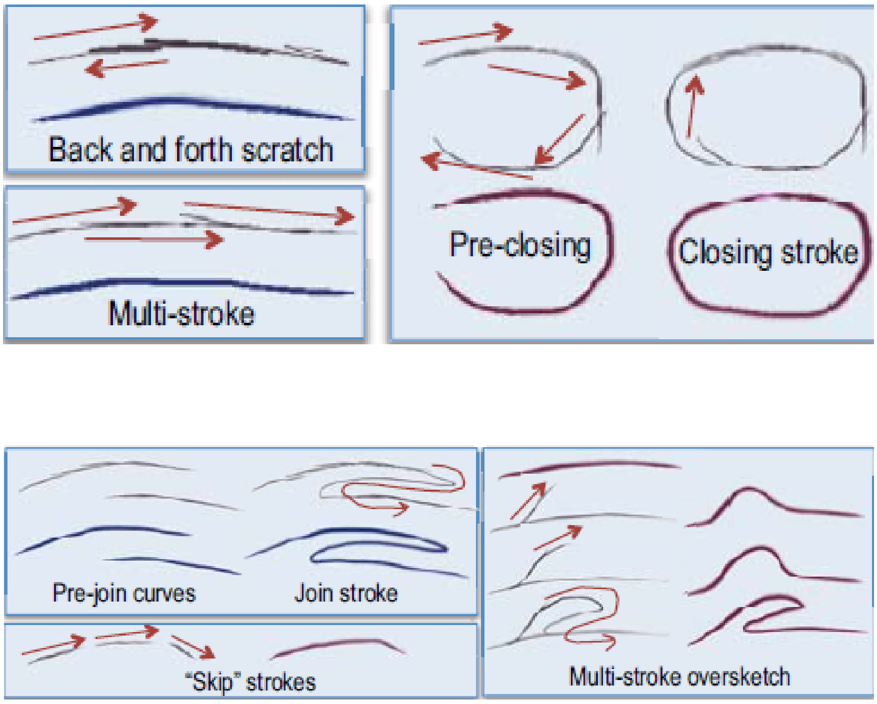


Figure 13: Stroke joined into curves and stroke refinement (Grimm and Joshi, 2012). *Reproduced from (Grimm and Joshi, 2012).*

The system requires the users to roughly input some strokes from the input device such as a pen. Artists when drawing usually draw in a discontinuous fashion. Back and forth, over-sketched, multi-stroking and disjoint strokes are common in traditional drawing practices. The system tolerates these rough user inputs of strokes and converts the strokes into curves. The user is able to perform different manipulation operations on the curves such as (dragging, scaling, rotating, smoothing, erasing some or all) by clicking on a curve, and selecting the desired option from the curve menu.

After the user has drawn a few curves, he/she can proceed towards adding depth (in 3D) to the curve by dragging the curve in the view direction (z-axis).

In contrast with ILoveSketch, this system does not use epipolar constraints to specify depth values along non-planar curves. Instead, treats the problem as one of over-sketching. It is very difficult for a user to envision what a curve would look like from two different views, so instead this system always create a 3D curve. The user can then change the view and over-sketch or continue that curve from the new view. The authors have used a novel depth interpolation and extrapolation technique to make the new stroke consistent (in depth) with the existing curve. For 3D surface creation the system provides visualization and interface support for automatically and semi-automatically snapping curves together and orienting them. In particular, a novel ribbon rendering method is used, which makes visualizing and editing the curve orientation (which direction is “out”) easier.

JustDrawIt benefits from its 2D drawing interface which is easy and intuitive for the artists to learn, however its main disadvantage is in its 3D curve drawing aspect, which has a steep learning curve. Also the artists have also stated that they have not performed a formal user study and only tested the system with four experienced users and allowed them to experiment with it.

2.2.5 Data Driven Systems

Researchers have proposed several novel morphing techniques and found them to be attractive techniques towards generating 3D models from several posed models in a database, and 2D images. These systems are essentially 3D search engines making heavy use of 3D and 2D shape databases to guide the user towards modelling plausible 3D models. These systems employ 2D and 3D shape descriptors to retrieve parts of the models from databases as well as complete models. Several good surveys exist that give an excellent overview of state-of-art shape descriptors such as (Zhang and Lu, 2004), (Zhang et al., 2007), (Yang et al., 2007), and (Kazmi et al., 2013). This section discusses a recent and powerful SBM systems based on 3D search engines.

Figure 14: Suggestive contours algorithm applied to the David statue 3D model (DeCarlo et al., 2003). *Reproduced from (DeCarlo et al., 2003).*

2.2.5.1 BoF+GALIF: Eitz et al. (2012) have developed a 3D search engine which utilizes existing and novel algorithms in its pipeline. This system accepts a rough user sketch as input and matches the sketch with several sketched renderings of the 3D model present in the database. A 3D model in the database is first subject to different camera view captures. Then each view is rendered as a line drawing using Suggestive Contours algorithm introduced by DeCarlo et al. (2003) (Figure 14). Then a shape descriptor is computed using all line drawings using Gabor Filters algorithm. A visual vocabulary is developed from all the models in the databases using the Bag-of-Features approach. The input sketch is also subject of Gabor Filters and then the feature vector is matched with the visual vocabulary to find the closest matches.

2.2.5.2 A Data-Driven Approach to Realistic Shape Morphing: This is a data-driven approach that aims at computing morphed models by using models already present in the database. The models in the database are clustered to form local shape spaces. To find the closeness between the pairs of models in the clusters, a simple distance metric is used. The morphing problem is then solved by solving a global optimization problem of finding a minimal distance path within the local shape spaces. For producing the final models, an extended as rigid as possible interpolation is used. The authors have casted the morphing problem as a global optimization problem of finding a shortest path from the source to the target with intermediate models from the local shape spaces using the distance measure.

2.2.6 Morphing based systems

2.2.6.1 Modelling from Contour Drawings: In this approach by Kraevoy et al. (2009), a contour drawing is used as an input drawing and a correspondence is found between the given drawing and a template model to produce the final model (Figure 15). To produce the final morphed model, an optimization problem is solved using Hidden Markov Model to find the optimal correspondence between a sequence of 2D stroke points and 3D template vertices. The second contribution of the authors is that they have develop an iterative correspond and deform framework which is key to making this sketch-based modelling work. The input drawing is represented as a sequence of points with associated outward pointed normals. The task of finding a correspondence between a sequence of 3D vertices for every sketched contour stroke, i.e. finding a continuous sequence of vertices that best match with the continuous sketched contours. This calls for solving an optimization problem because the aim here is to minimize the transition costs when the distances travelled along the drawn contour and between 3D vertices are equal. To solve this optimization problem, the authors have used the Hidden Markov Model and Viterbi algorithm (dynamic programming solution).

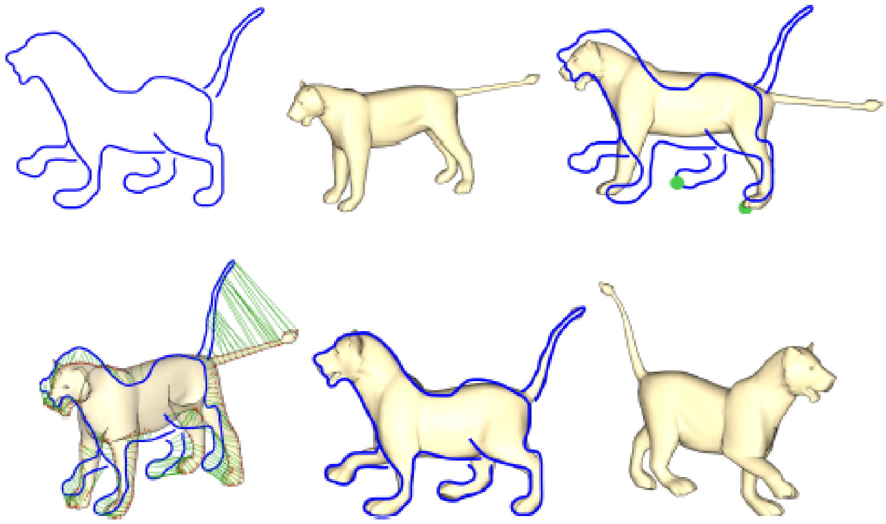


Figure 15: Overview of morphing process of (Kraevoy et al., 2009). *Reproduced from (Kraevoy et al., 2009).*

For the HMM, the points are treated as a set of observation states, while the 3D vertices to be inferred are treated as a set of hidden states. HMM requires emission probabilities and transition probabilities. These are solved using the formulas given in the paper. The deformation of the template model to match the sketched contours is performed iteratively using mean value encoding.

One of the limitations with the proposed technique is the use of HMM technique to solve the optimization problem, which is slow to compute and can take up to 1 to 2 minutes to find the optimal correspondence.

2.2.6.2 What Shape are Dolphins? Building Morphable models from 2D images: In this approach, Cashman et al. (2013) have used several images and a template model as inputs, to compute a morphable model. Given a set of images, the approach first extracts the silhouette of the image and the silhouette is sampled with discrete points and associate normal with these normal. Each image is also accompanied with user specified point constraints, which help to direct the optimization into the correct energy well. Each image is also associated with camera projection parameters to represent each image to lie in the xy plane facing the z-axis. With this setup, the authors have proposed an optimization algorithm, and the formulation of an objective function / energy function. The optimization algorithms match the image silhouette with the template model.

2.2.6.3 Human shape correspondence with automatically predicted: landmarks In (Wuhrer et al., 2012), the authors have proposed a new approach for finding the correspondence between two 3D models in a database by automatically predicting the landmarks. The approach is composed of 3 main steps 1) Predicting landmarks, 2) Finding point-to-point correspondence, and 3) Evaluating the accuracy of the correspondence between the 3D models. At the core of this approach is the use a human body database CEASAR. This database is composed of thousands of scanned high resolution and realistic human models. For the prediction of landmarks, the authors have used a modified version of the algorithm proposed by Azouz et al in (Azouz et al. 2006). This algorithm identifies 73 landmarks on the human body by learning from a pairwise Markov Network. The actual correspondence between a template model and a scanned model is found by using a 2 step alignment process. Firstly the authors have utilized Radial Basis Functions (RBF) and secondly their approach deforms the template model using a non-linear optimization algorithm based on Broyden-Fletcher-Goldfarb-Shanno quasi-Newton approach (Liu & Nocedal. 1989) to solve the optimization problem. The final step of the approach is ‘Evaluation’, in which the accuracy of the proposed approach is conducted over a large database of 500 human body scans. The accuracy is found using 3 measures: compactness, generalization, and specificity. The authors used a modified version of the landmarks prediction algorithm by compromising some amount of accuracy but gaining some amount of speed in computation. Loss of accuracy is one disadvantage the authors have mentioned in paper. One of the advantages of Radial Basis Function is that it is easy to compute and implement as it is a Neural Network that can be solved using simple linear algebra computations.

2.2.7 Conclusion

3D modelling has a very long way to cover to match the convenience of drawing on paper and expressing imagination on paper. Interpreting different contour lines in a sketch by a machine is one of toughest problems in SBM. Same can be said for generating realistic 3D characters from 2D sketches. Curve network system can be a promising future for SBM as they provide a paradigm and interface for 3D sketching, which in nature shares a similar experience with 2D sketching on paper.

This survey has investigated state of the art techniques in Sketch-based Modelling and pointed out their advantages and disadvantages and also classified these techniques into several categories.

3 Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Deformations

3.1 Introduction

Sketch-based modelling (SBM) has a great potential in enhancing and improving conventional 3D modelling and sculpting workflows in the animation industry (Olsen et al., 2009). Over the past three decades the research in SBM has seen substantial advances. The existing SBM approaches can be broadly divided into two categories 1) direct mesh generation, and 2) template-based character creation also called data-driven character modelling in this chapter. Direct mesh generation aims at generating geometrical objects from a user's drawn sketches directly such as Teddy by Igarashi et al. (1999). Its weakness is that it is not suitable for modelling detailed or complex character models. Template-based modelling uses a 3D template model to provide the information about the details and complex shapes representing the character models and it avoids the weakness of the direct mesh generation. It provides a promising solution to create detailed or complex 3D character models from a simple sketch.

The current state of the art in template-based modelling focuses on highly automatic extraction of occluding contours. Due to complex topology and the shapes of character models, the automatic methods lead to incorrect results and have been known to produce inconsistent contours (Bénard et al., 2014). Another problem is the inefficient computing of one-to-one correspondence between the user's drawn sketches and the 3D template models and time-consuming mesh deformation algorithm to deform the 3D template models to fit user's drawn sketches. These problems make current template-based character modelling impractical in real-time interactive modelling applications.

In order to tackle the above problems, this chapter proposes a novel approach to efficiently create detailed and high quality 3D character models from user's drawn sketches. For a sketch-based modelling system to be valuable, combining human's perception and necessary user interactions to steer the algorithm is vitally important to aid the computer in quickly finding accurate correspondences. This hypothesis has served as one of basic ideas behind the proposed approach. The necessary interactions, as mentioned in detailed in Section 3.3 are 1) drawing a rough occluding contours (guided contours) on the template 3D model, and 2) selecting anchor points on the guided occluding contours and the sketched occluded contours. The second idea is to tackle time-consuming iterative calculations caused by using mesh editing algorithms by incorporating skeleton-based large deformation and mesh editing's small deformation to efficiently deform a 3D template model to exactly fit the user's drawn sketches. The third idea is to develop a user interface to integrate all the functions into a complete system to speed up the creation of character models. With the developed user interface of this proposed approach, users first interactively create their drawn sketches with or without a 2D reference image. Then, a 3D template model whose occluding contours best match their sketches is automatically retrieved from the dataset developed in the system. Next, preliminary occluding contours of the 3D template model are automatically generated to guide the users to create the correct and consistent occluding contours on the mesh surface of the 3D template model. After that, with the help of very few anchor points specified by user input, automatic subdivision is performed to help determine correct correspondences between the user's drawn sketches and the 3D template model. Finally, skeleton-based deformation and mesh editing are combined to determine large and small mesh deformations, respectively, and generate new character models from the 3D template model.

The following are the main contributions of the work presented in this chapter.

1. Combining human perception and the necessary interactions with a computer's powerful computing capacity to develop a fast and accurate method of determining correct correspondences between the user's drawn sketches and the 3D template model.
2. Proposing a hybrid mesh deformation technique which employs skeleton-based deformation to tackle large mesh displacements or deformations and mesh editing to deal with small mesh deformations for quickly deforming the 3D template model into a new model exactly matching the user's drawn sketches.
3. Developing a user interface with complete functions to facilitate and speed up creation of new character models from the user's drawn sketches and a template model.

3.2 Related Work

Existing sketch-based modelling systems can be used to model organic and inorganic models from a single view (Igarashi et al., 1999) or multiple views (Rivers et al., 2010). (Cook and Agah, 2009) and (Olsen et al., 2009) have presented thorough surveys on sketch-based modelling techniques. Some notable systems for modelling sketch-based inorganic models are (Chen et al., 2013), (Xu et al., 2014), (Rivers et al., 2010), and (Bae et al., 2008). In (Xu et al., 2014) the authors have introduced a new mathematical framework for inferring 3D curve networks from 2D drawings that are drawn from a perspective view, inspired by design, vision and perception literature. However their technique does not seem to work well for sketches that are drawn from orthographic views and is not suitable for modelling organic characters. Chen et al. (2013) have demonstrated a 3D model reconstruction technique from single photos, by automatically fitting the boundaries of a generalized cylinder to the boundaries of the subject in the image. Rivers et al (Rivers et al., 2010) have proposed a new and simple algorithm towards computing the 3D Constructive Solid Geometry (CSG) by leveraging the special properties of silhouette cylinders. Bae et al. (2008) have developed a robust and feature rich system for 3D sketching. The basic "feel" of the system borrows from that of a physical paper sketchbook.

It has been observed that modelling of organic human characters is a highly challenging task for artists, and some SBM systems have tried to overcome these challenges such as (Buchanan et al., 2013) and (Gingold et al., 2009), but these systems possess several limitations, for example these techniques do not cater for producing highly realistic models. Moreover the modelling of body parts which are hidden in input sketches is highly challenging. To overcome these limitations some researchers have proposed template-based or data-driven systems that use a 3D human character as a template model and deform this model by either morph it or use cage-based deformation techniques (Lipman et al., 2008). The area of template-based modelling for generating organic or human characters particularly from input sketches has been visited very rarely by researchers in the past. Some good examples of template-based SBM systems are (Shtof et al., 2013) and (Kraevoy et al., 2009). Shtof et al. (2013) have introduced a snapping technique which automatically reshapes and snaps simple 3D geometric primitives to 2D sketch primitives, then improves the model globally by inferring geosemantic constraints that link the different parts. This technique leads to incorrect topology or tedious modelling, when it comes to modelling human characters.

An elegant technique for sketch-based modelling has been proposed in (Kraevoy et al., 2009) which involves finding precise correspondences and mesh deformation. For finding one-to-one correspondences, the Hidden Markov Model (HMM) has been harnessed, which is a computationally expensive algorithm (Rabiner, 1989). For deforming the mesh, the Mean Value Coordinates (MVC) has been used which incurs a huge bottleneck in the overall algorithm (Floater, 2003). The method is iterative in nature and solves the energy minimization problem to obtain a deformed mesh in which each iteration includes heavy computations for finding correspondences and mesh deformation. On a mesh of more than 20K polygons, HMM combined with MVC incurs an overall delay of two to ten minutes. This limitation renders their algorithm impractical in an interactive modelling environment.

Template-based modelling is heavily dependent on the extraction of feature curves. Feature curves are classified as silhouette contours, occluding contours, and suggestive contours as shown in (DeCarlo et al., 2003). DeCarlo et al. (2003) have developed a powerful system for extracting feature curves in real time from 3D models. However the occluding contours extracted by their technique produces inconsistent (broken) and non-sequential contours. Bénard et al. (2014) has attempted to solve a long occurring problem of finding inconsistent contours using the method presented by Koenderink (1984) and DeCarlo et al. (2003). However as mentioned by the authors, their algorithm is computationally expensive and does not guarantee 100% contour consistency. Due to this reason their method is infeasible to use within the context of the approach. In this chapter the main focus is on extracting occluding contours as a sketching guide for the user, so that the user can sketch a curve on the 3D mesh as close as possible to the occluding contours.

Several techniques have been proposed for mesh deformation. Nieto and Susín (2013) have discussed and compared several useful deformation algorithms including Mean Value Coordinates, Harmonic functions and Green coordinates in their survey and concluded that Mean Value Coordinates is the best performing algorithm for deformation from the perspective of efficiency. Several techniques exist for Skeleton-based deformation. Chen et al. (2009) have proposed a user friendly interface to generate a control skeleton of a model automatically and a skeleton segment can be adjusted by drawing a new curve. Buchanan et al (Buchanan et al., 2013) have demonstrated a system which generates low-polygon models directly from input character sketches without any interactivity by the artist.

3.3 System Overview

As indicated in Figure 16, the proposed approach consists of an offline phase and an online phase. The offline phase consists of the following steps: 3D templates collection, silhouette + occluding + suggestive contours extraction, shape descriptor formulation, and updating of 3D template dataset. The online phase is composed of the following processes: 1) sketch input/creation, 2) shape descriptor formulation and matching, 3) 3D template model retrieval, 4) occluding contours extraction, 5) correspondences computation, 6) skeleton-based deformation, and 6) final mesh deformation. The following subsections elaborates the steps in the online phase.

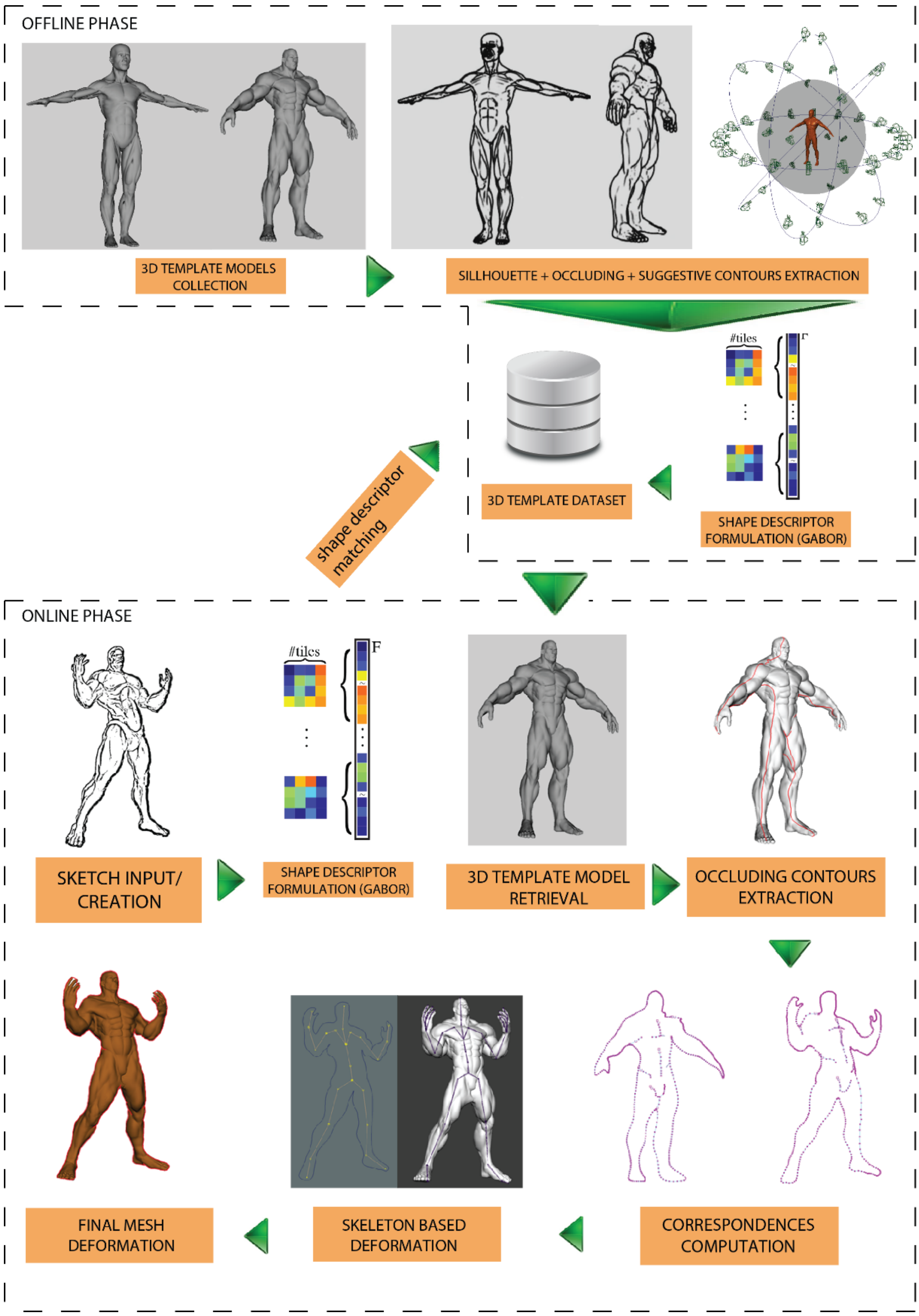


Figure 16: Process flow of the proposed approach.

3.3.1 Sketch Input / Creation

As discussed above, users drawn sketches can be divided into three different types: silhouette contours, occluding contours, and suggestive contours. According to Wuhrer and Shu (2012), suggestive contours are view-dependent features that become contours in nearby views. Suggestive contours were pioneered by DeCarlo et al. (2003) and they are a powerful tool to convey shape information. A comparison among the three different types of contours is illustrated in Figure 17 (Wuhrer and Shu, 2012).

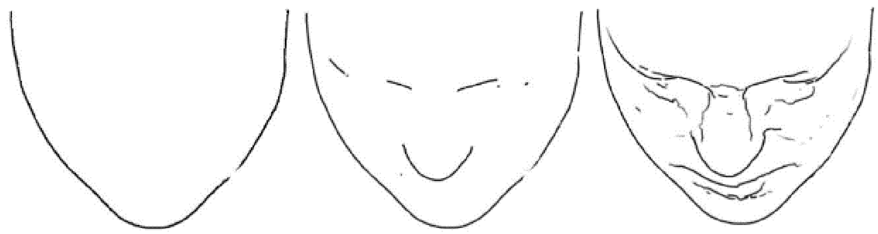
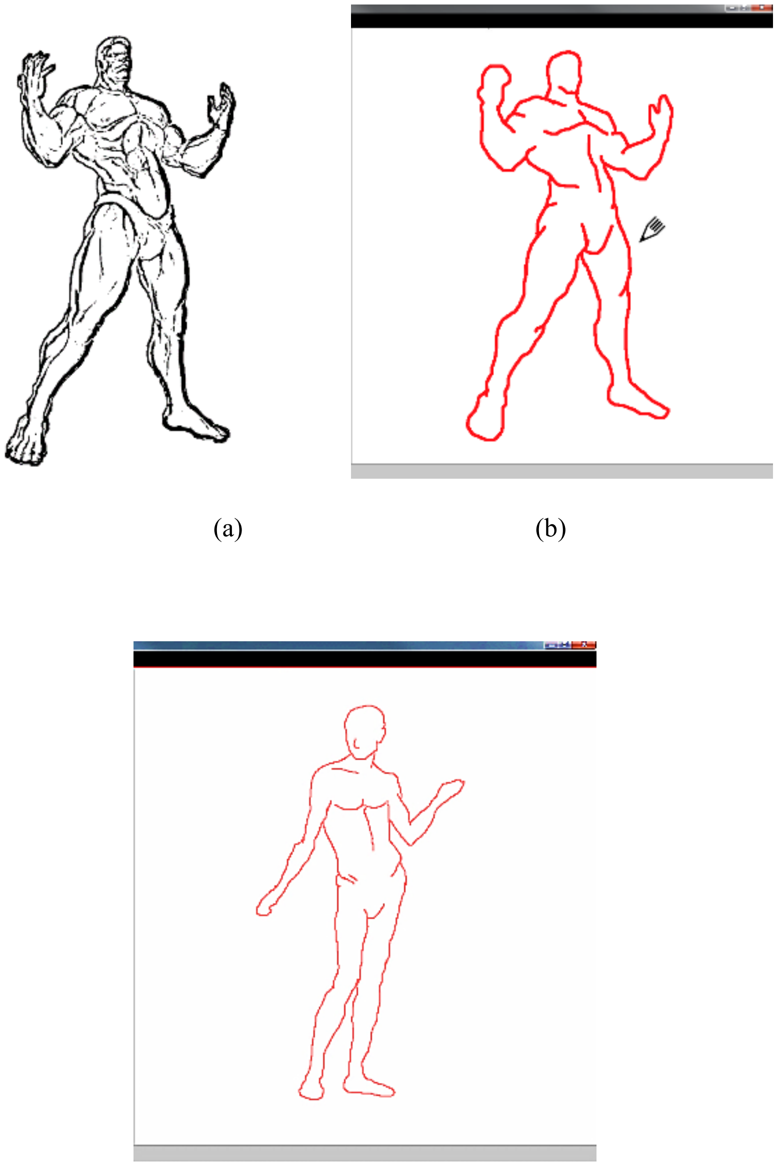


Figure 17: Comparison among contours a) Silhouette contour, b) Silhouette + Occluding contour, and c) Silhouette + Occluding + Suggestive contour of a face model. *Image source: (Wuhrer and Shu, 2012).*

Suggestive contours and occluding contours are both vitally important for the proposed approach to work. The proposed approach needs suggestive contours to compute the shape descriptor and retrieve the 3D template model from the dataset (Section 3.2). However, for template model retrieval the user is provided with the freedom to use any type of contours to draw the input sketch because of the robustness of the 3D model retrieval algorithm incorporated in the approach (Section 3.2.3). Occluding contours are also needed by the proposed system to find correspondences (Section 3.4).



(a) (b)

(c)

Figure 18: Sketching Interface of the proposed approach. (a) Input 2D image, (b) User drawn occluding contours with reference to the input image, and (c) User drawn occluding contours without using the reference image.

In the proposed system, the user can use a reference image as the input sketch (Figure 18(a)). The system also allows the user to trace occluding contours over the input sketch (Figure 18(b)). Alternatively, if a reference image is unavailable, he/she can draw the sketch with the help of the prototypical interface. Figure 18(c) depicts the occluding contour created without referring to any 2D images. The occluding contours drawn by the user are called the created occluding contours and represented within the system as polylines rendered on the XY plane and the value of z-coordinate is set to 0.

3.3.2 Template Model Retrieval

In this step, the input sketch is used as an input query to retrieve a template model from the proposed system's 3D dataset directly. To achieve this goal, the following algorithms are conducted: creation of a dataset of 3D template models, extraction of suggestive contours, and retrieval of 3D template models.

3.2.2.1 Dataset of 3D template models: For the proposed system a dataset of 3D template models was set up through: 1) collecting 3D human character models from the Internet including available 3D databases, and 2) generating them using MakeHuman which is an open source tool for making 3D character models (www.makehuman.org). The created dataset has been included in the developed system. In the dataset, each model is generally composed of a high polygon count (more than 25K polygons). All the models are in either T-pose or A-pose. These T-pose and A-pose models were collected from several freely available online repositories including (www.tf3dm.com), (www.clara.io) and (www.cgtrader.com). The main rationale for only using models in T-pose and A-pose is due to the observation that most of the models used in the animation industry are initially in these poses, and subsequently modified by artists to deform them into desired poses. To enrich the variety of the system's dataset more models can be added to it. The duration for adding a new model depends on the amount of polygons to process for generating the shape descriptors, and averagely takes approximately 2 minutes. In addition to the template models, the dataset also contains 2D sketched renders (generated using suggestive contours, occluding contours, and silhouette contours). The creation of this dataset is an offline process and the process of generating 2D sketched renders is explained in the following sections.

3.2.2.2 Extraction of suggestive contours: The system automatically generates 2D sketched contour representations of the models in the dataset. To acquire 2D sketched renders from 3D models a non-photorealistic rendering (NPR) algorithm proposed by DeCarlo et al (2003) is employed. It generates suggestive contours and silhouette contours of organic 3D models in real time (Figure 19).

User drawn sketches can be from any view angle. In order to obtain 2D sketched renders of 3D models in the dataset which is from the view angle same as or close to that of the created sketches, a camera is positioned at different points for each of models in the dataset to take the images of the model. These points are on four circumferences circling a bounding sphere of the model, and the camera points towards the center of the model as shown in Figure 20.

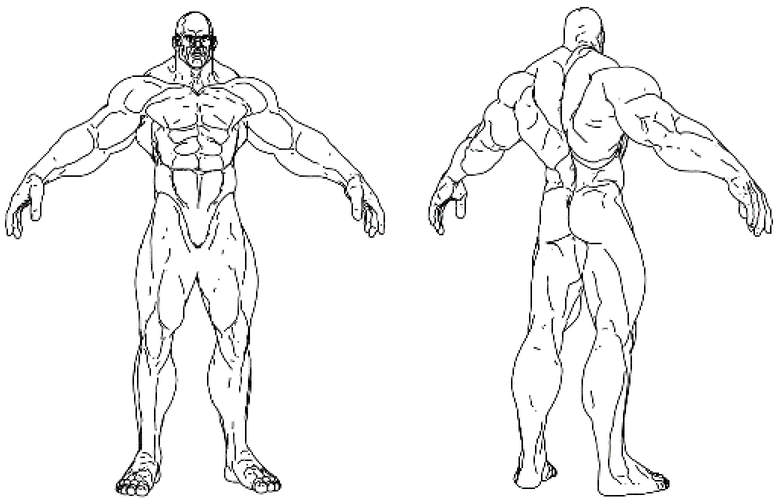


Figure 19: Suggestive contours and silhouette contours generated using (DeCarlo et al., 2003) showing the model as a line drawing.

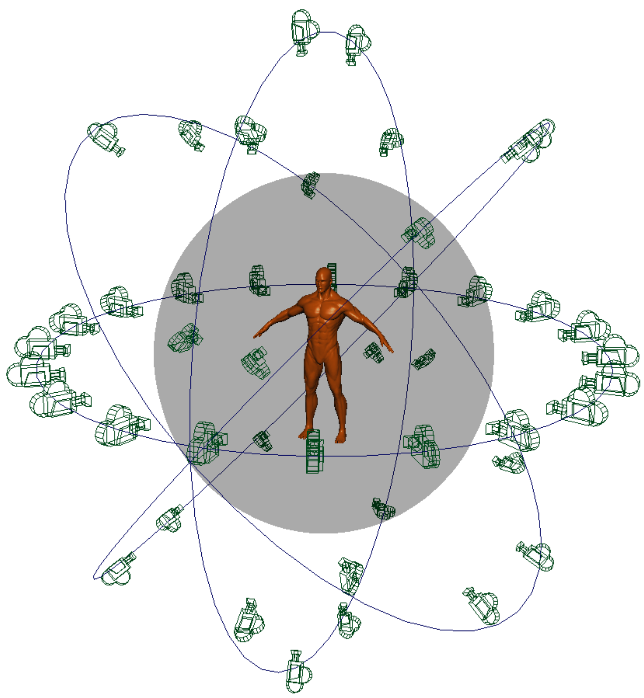


Figure 20: Camera setup for extracting 2D sketched render views of a 3D model in the dataset.

3.2.2.3 Retrieval of 3D template models: After extracting 2D sketched renders, the system computes the feature descriptor vector of each 2D view. This step draws inspiration from Eitz et al. (2012) and uses a powerful shape descriptor based on Gabor Filters to compute the shape descriptors. A Gabor filter is defined in the frequency domain as:

$$g(u,v) = \exp\left(-2\pi^2\left((u_\theta - \omega_0)^2\sigma_x^2 + v_\theta^2\sigma_y^2\right)\right) \tag{3.1}$$

where $(u_\theta, v_\theta) = R_\theta(u,v)^T$ is the standard coordinate system rotated by angle θ A Gabor Filter can be tailored according to the following parameters:

ω_0 : peak response frequency

θ : filter orientation

σ_x : frequency bandwidth

σ_y : angular bandwidth

When the sketch is multiplied with the Gabor Filter in the frequency domain, the filter masks all content that does not contain the right frequency and orientation: the filter responds only to a subset of the lines in a sketch.

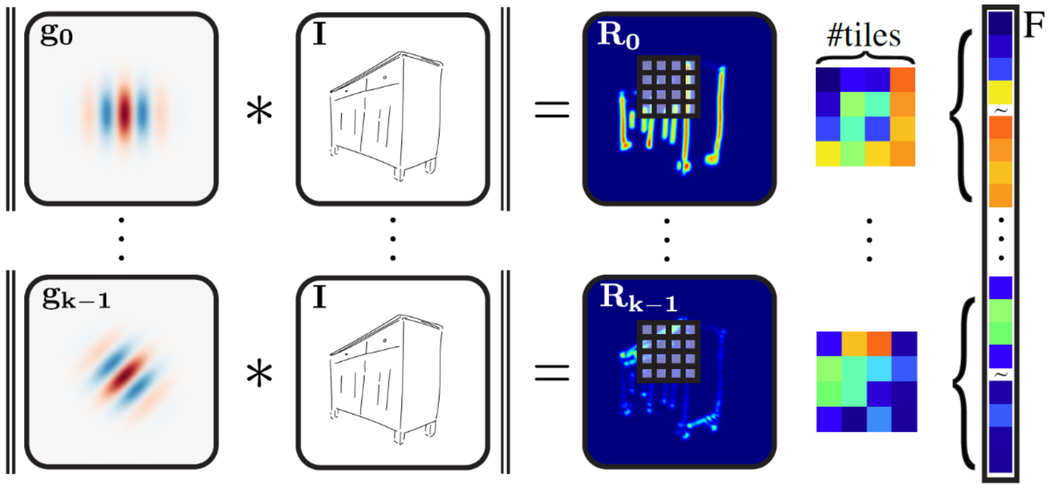


Figure 21: Process demonstrating the formulation of feature vector using Gabor Filter bank. *Reproduced from Eitz et al. (2012).*

To compute the feature vector, the method defines a filter bank of Gabor functions g_i with k different orientations, while keeping all other parameters fixed. The method then convolves the sketch with the Gabor functions from the filter bank to yield a set of filter response images:

$$\boldsymbol{R}_i = \left\| \mathrm{idft}\left(g_i * \mathrm{dft}(\boldsymbol{I})\right) \right\| \tag{3.2}$$

where $\boldsymbol{I}$ is the input sketch, $*$ denotes point-wise multiplication and ‘idft’ and ‘dft’ denote the inverse/forward discrete Fourier Transform (see Figure 21 for a visualization).

σ_y determines the amount of overlap between filters g_i depending on the numbers of orientations in the filter bank. Given the number of orientations k the method defines the 'filter orientations' i.e $\theta's$ used for the filter bank as:

$$\theta \in \left\{0, \frac{\pi}{k}, \dots, \frac{(k-1)\pi}{k}\right\} \quad (3.3)$$

The method then divides the response images into $n \times n$ cells C_{st}, where n is the number of tiles. According the method $(x, y) \in C_{st}$ if the pixel with coordinates x and y is contained in the cell with index (s, t). Finally the method defines the local feature $\boldsymbol{F}$ as a $(k \times n \times n)$ feature vector. In each dimension, $\boldsymbol{F}$ stores the average Gabor filter response within a cell C_{st} for orientation i:

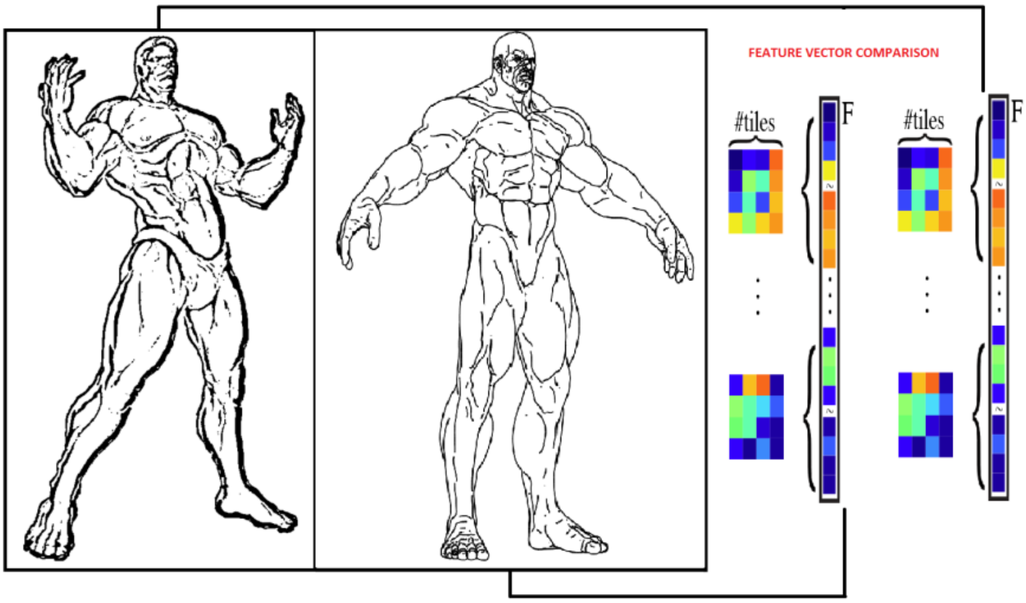
$$\boldsymbol{F}(s, t, i) = \sum_{(x,y) \in C_{st}} \boldsymbol{R}_i(x, y) \quad (3.4)$$

For inserting a value in $\boldsymbol{F}$, the method performs bilinear interpolation in the special domain. The feature vector is stored as a histogram.

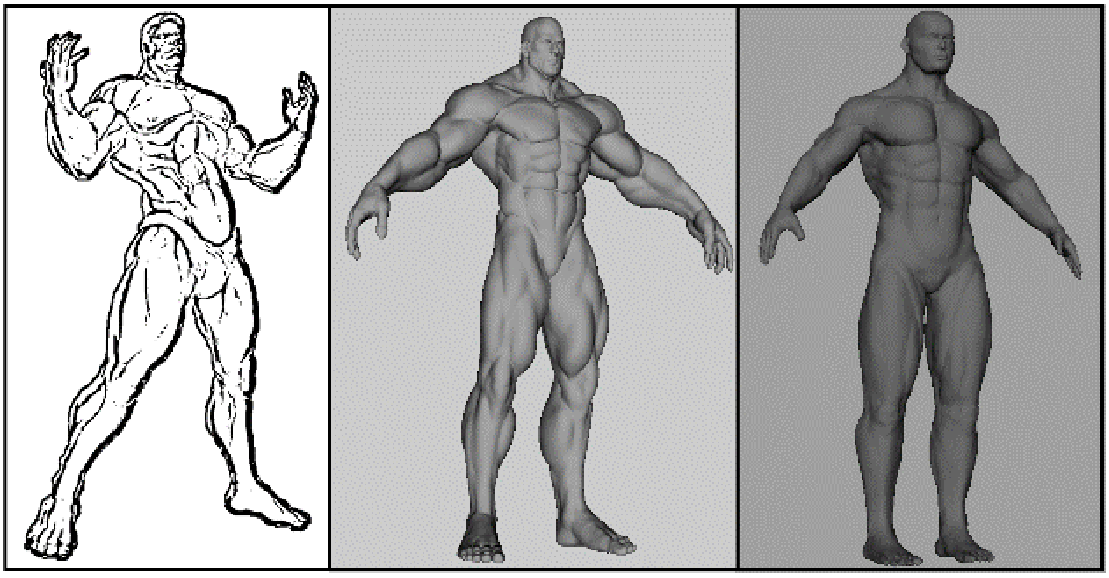
To compare two feature vectors, a vector space model was employed by the method in Eitz et al. (2012) to define similarity between two histograms. Let h and $\bar{h}$ be two histograms representing two sketches. The method defines their similarity as:

$$s(h, \bar{h}) = \langle h, \bar{h} \rangle / \|h\| \|\bar{h}\| \quad (3.5)$$

Two sketches are considered similar if their histograms point into the same direction.



(a)



(b)

Figure 22: Process showing retrieval of 3D template model (a) Comparison between feature vectors of the user's sketch and a 2D sketched render of a model in the dataset, (b) Models similar to input sketch are ranked here.

Based on the feature vector comparison, a template model is retrieved that most closely matches the user's created sketches and present that to the user. Figure 22(a) gives such an example. As shown in the figure, using a user's created sketch (left) the proposed system performs a comparison between feature vectors of the user's sketch and a 2D sketched render of a model in the dataset. In Figure 22(b) the two most similar models found in the dataset are shown, where the middle model is the most similar model.

From the retrieved template model, the coordinates and view direction of the camera angle are stored, which will be used to compute the occluding contours and correspondences in the subsequent steps.

3.3.3 Occluding contours extraction

In order to deform the extracted 3D template model to make it exactly match the user's created sketch, the following research studies were carried out. Firstly, it was investigated how to generate occluding contours of the 3D template model which will be used to provide guidance for the user to sketch the curve on the surface of the template model in this subsection. Secondly, an examination of the correspondence between the user's created sketch and the sketched curve on the template model was carried out.

The proposed approach first requires the user to draw the occluding contours of the input sketch, which can be easily drawn by simply tracing over the silhouette contours and drawing some internal contours which are enough to convey the inner shape. The method only requires this in case the user has not already provided occluding contours at the start of the pipeline. The system then computes the occluding contours of the template model and renders them on the mesh as shown in Figure 23. With the algorithm developed in (DeCarlo et al., 2003), occluding contours are extracted from the 3D model in real-time using the expression defined in Equation (3.6). It was observed that it took approximately 50 milliseconds to generate occluding contours on the model shown in Figure 23, which has approximately 36000 polygons.

Let n be the normal vector at a point p, and v be the vector pointing from the point p to the camera. On a smooth mesh, the occluding contours are made up of vertices where the dot product of the view vector and the normal vector is zero, which is represented as:

$$n(p).\, v(p) = 0 \tag{3.6}$$

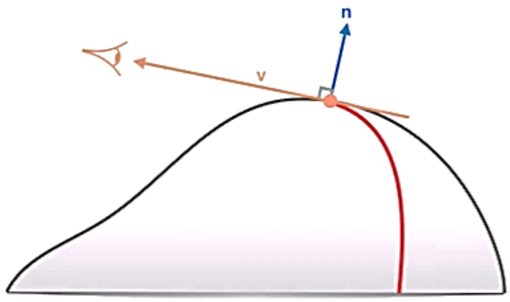


Figure 23: Occluding contour (red curve) on a smooth surface.

To satisfy equation (3.1) exactly, the extracted occluding contours contain too few vertices to be used in the subsequent steps in the proposed system's case. In order to acquire enough vertices to serve guidance for the user, those vertices were selected, which resulted from the modified Equation (3.1), i.e., $-0.3 \le n(p).\, v(p) \le 0.3$. Such a treatment extracts sufficient vertices. However as shown in Figure 24 the extracted vertices do not lead to desired occluding contours, which cannot be used to determine correspondences effectively.

To solve this problem, the approach proposes a user friendly method to generate correct occluding contours in a semi-automatic fashion thus allowing the system to compute correspondences very quick. Such a method uses the extracted occluding contours to guide the user to draw smooth and sequential occluding contours on the template model surface. The user's drawn sketch is represented with a B-spline curve called a *guided occluding contour* which will be used in the correspondence computation discussed in the subsection below.

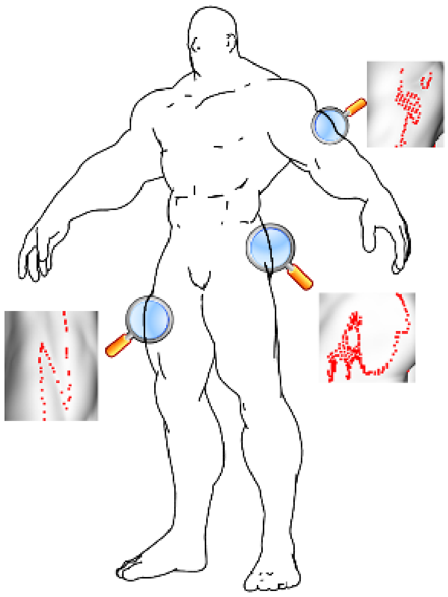


Figure 24: Inconsistent occluding contours magnified on the template model.

3.3.4 Correspondence Computation

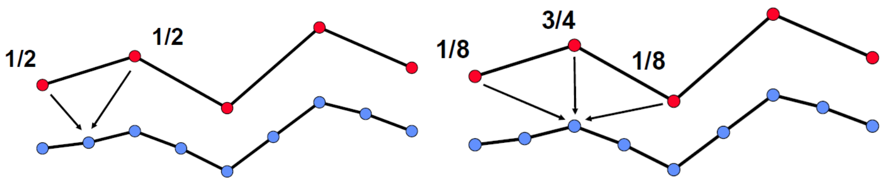
In the proposed approach, correspondence computation is based on three basic ideas. The first is to manually select very few anchor points on the guided occluding contour (B-spline curve) and the corresponding anchor points on the user's created 2D silhouette contour. The second is to generate the same number of points between the adjacent anchor points on both the B-spline curve and the user's created sketch which solves the correspondence problem between the B-spline curve and the user's created sketch. The third is to determine the correspondence vertices on the 3D template model by using the KD-tree algorithm to find the vertices closer to the points on the B-spline curve which solve the correspondence problem between the extracted 3D template model and the user's created sketch. These three basic ideas will be elaborated below.

3.2.4.1 Selection of anchor points: After the user has obtained the guided occluding contour (B-spline curve), he/she can select very few anchor points on the guided silhouette contour and their corresponding anchor points on the 2D user's created sketch to aid the proposed algorithm in finding correct correspondences. This step is very easy for a human to perform as humans are very quick at identifying these apparent correspondences, as compared to a computer. A maximum of 10 anchor points is sufficient. However finding the final one-to-one correspondence is a cumbersome and error prone task for a human. The next subsection explains the proposed algorithm of finding the final correspondences accurately and quickly.

3.2.4.2 Determining correspondence between created and guided occluding contours: To establish a one-to-one correspondence between the user input sketch (created occluding contours) and guided occluding contours, the number of points in the 2D curves must be equal to the number of vertices in the traced 3D occluding contours. The proposed algorithm has achieved this in two steps. First the proposed system decomposes the input and guided occluding contours into a set of sub-curves using the anchor points selected by the user in the previous step, as boundaries of the sub-curves. The system then subdivide the sub-curves to insert new curve points thus resizing two corresponding sub-curves to contain equal number of points. The proposed approach uses Chaikin's algorithm (Chaikin, 1974) to add new curve points between the original curve points and updates positions of the original curve points thus making it smoother. This subdivision technique is defined by two rules, 1) odd rule (equation 3.7), and 2) even rule (equation 3.8). The algorithm first resizes the curves using the odd rule by inserting more points in the curve which contains fewer points, and then evenly distributes the points in the two curves using the even rule (Figure 25).

$$p_{2i+1}^{j+1} = 1/8(4p_i^j + 4p_{i+1}^j) \quad (3.7)$$

$$p_{2i}^{j+1} = 1/8(p_{i-1}^j + 6p_i^j + p_{i+1}^j) \quad (3.8)$$

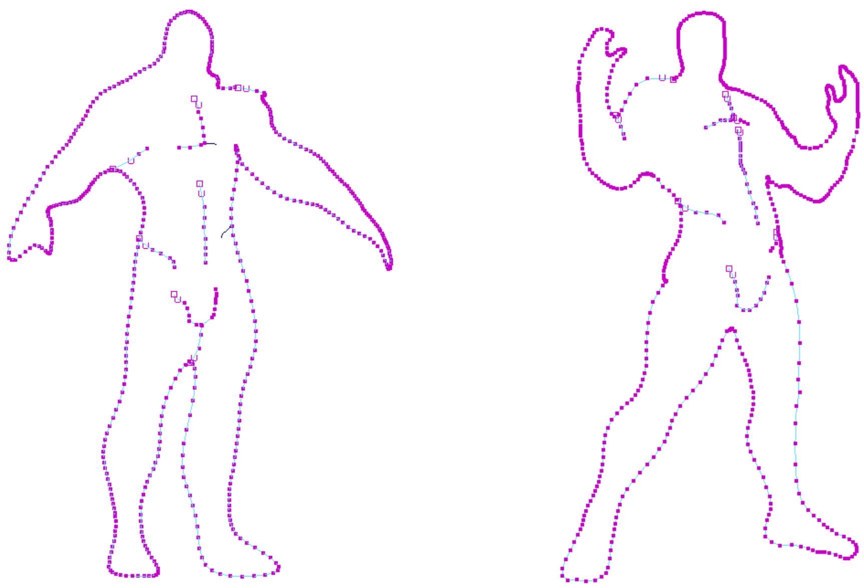


(a) (b)

Figure 25: stroke subdivision (a) Odd rule inserts new points in the curve, (b) Even rule repositions the old points.

In equations (3.7) and (3.8), i is the current vertex to be processed, while j is the total number of vertices in each iteration of subdivision. p_{2i+1}^{j+1} is the newly inserted vertex in equation (3.7), while p_{2i}^{j+1} is the repositioned vertex i.

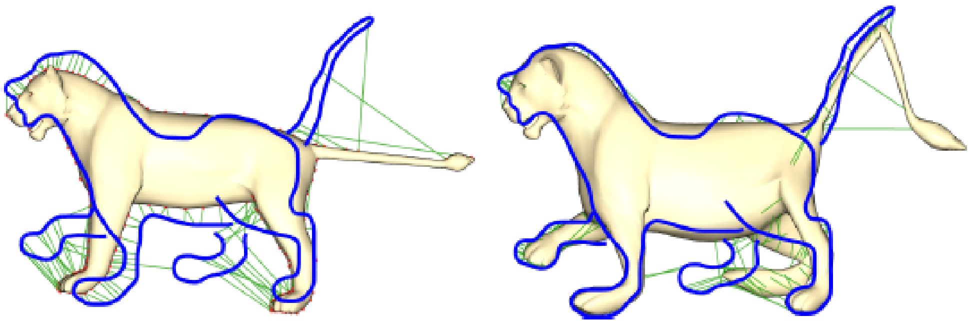
Once the two curves are resized to contain equal number of points, the system computes a one-to-one correspondence between these guided occluding contours (B-spline curve) and the input sketched contours (Figure 26).



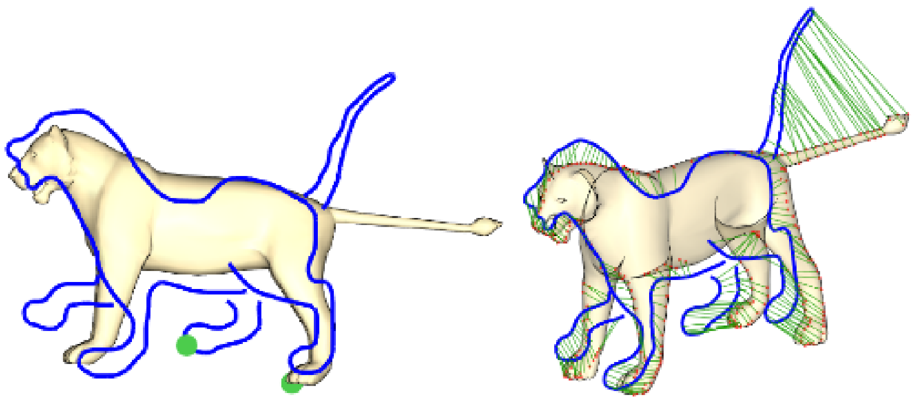
(a) (b)

Figure 26: Contour correspondence. The contours on the left show guided occluding contours that will be corresponded with the input sketch contours on the right.

3.2.4.3 Determining correspondence between created occluding contours and the 3D template model: The last work of correspondence computation is to find the vertices on the 3D template model which correspond to the points on the created occluding contours. This can be achieved with the following algorithm using the KD-tree data structure. The KD-tree spatial decomposition data structure finds the vertices on the 3D mesh that are closest to points on the guided occluding contour. The obtained vertices are naturally in sequence and give us the accurate correspondences with the points on the created occluding contours. An alternative to the KD-tree data structure is the BSP (Binary Space Partitioning). BSP is a generic process of recursively dividing a scene into two until the partitioning satisfies one or more requirements. Since BSP algorithm has not been used in existing research in the context of finding nearest vertices on a 3D mesh, the KD-tree data structure is employed instead. The KD-tree is an elegant data structure for range searching and nearest neighbour searching. It was first proposed by Bentley (1975) and is designed to handle special data in a simple way. Utilizing KD-tree data structure makes the proposed approach very fast as compared to the heavy computations performed by the Hidden Markov Model (HMM) in Kraevoy's method (Kraevoy et al., 2009). In an HMM, the input is a sequential series of observed states and the goal is to infer the corresponding sequence of hidden states that is most likely to have generated these observations. Their method for computing correspondences is a semi-automatic method as it requires the user to select a few anchor points to assist the algorithm to establish the right correspondences such as the case in which the two rear legs of the lioness are incorrectly deformed unless the user manually selects some corresponding vertices to accomplish correct correspondences (Figure 27).



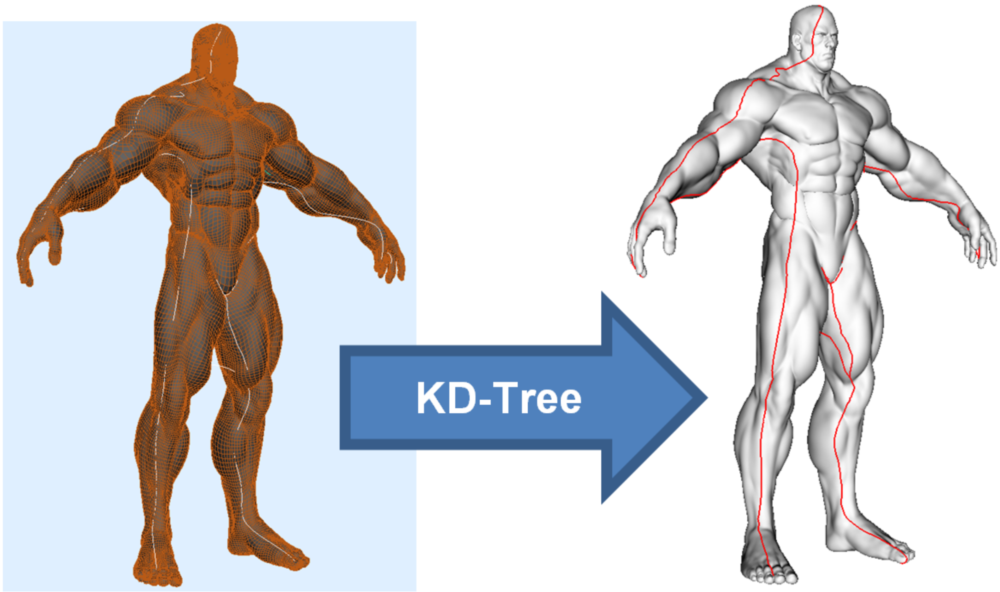
(a)



(b)

Figure 27: Correspondence computation method (Kraevoy et al., 2009) (a) Incorrect correspondence and deformation without specifying anchor vertices, (b) Correct correspondence and deformation with specifying anchor points for correspondences. *Image source: Kraevoy et al (Kraevoy et al., 2009).*

With the above treatment using KD-tree data structure, the system obtains the vertices on the 3D template model which form the read curve in Figure 28(b) where the white curve in Figure 28(a) is the guided occluding contour.



(a) (b)

Figure 28: Occluding Contours computation by the proposed method (a) Guided occluding contours drawn by the user, (b) Accurate occluding contours achieved by the proposed approach after applying KD-tree algorithm.

It was found that with some necessary user interaction in the proposed approach, the overall performance of finding correspondences has been boosted.

3.3.5 Mesh Deformation

When deforming a 3D template model to fit the input sketch contours, large mesh displacements or deformations may be involved as indicated in Figure 29. If there is a 3D template model in the dataset whose occluding contours are close to the user's drawn contours, the mesh editing algorithms described in Subsections 3.5.2 and 3.5.3 can be applied directly. However, if such a template model does not exist in the dataset, the pose of the user's created sketches might be quite different from the pose of the 3D template model. One such example is to deform the 3D template model whose occluding contours shown in Figure 26(a) exactly match the user's created occluding contour shown in

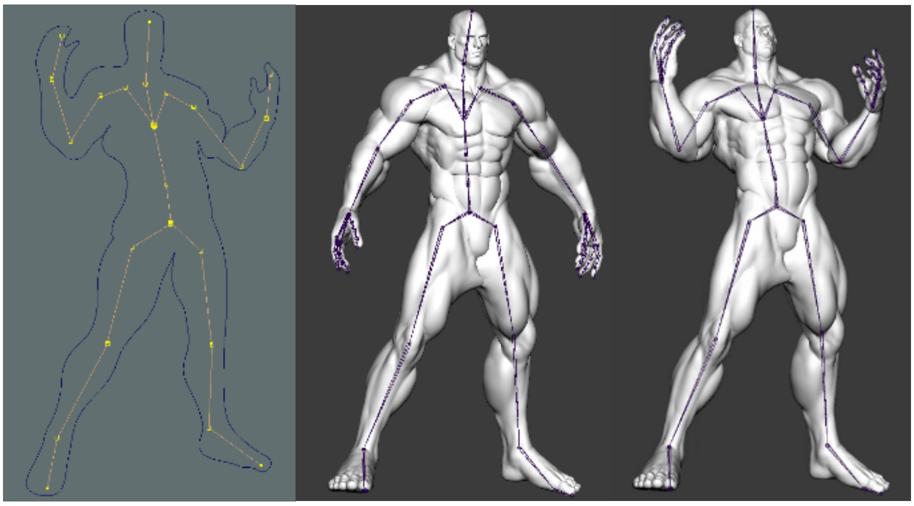
Figure 26(b). For this case, direct use of mesh editing algorithms will have to face large mesh displacements or deformations and involve time-consuming iterative calculations.

As reported in (Kraevoy et al., 2009), when using the Mean Value Coordinates, an iterative match-and-deform algorithm is involved. This typically requires a total of ten to twenty outer iterations of matching and deforming to obtain the final result leading to an entire process of two to ten minutes, which makes the algorithm unsuitable in an interactive modelling environment. In order to tackle this problem, a hybrid skeleton-based deformation and mesh editing scheme are proposed. Skeleton-based deformation deals with large displacements or deformations, while mesh editing removes the small deformation errors between the input sketch contours and the 3D template model after skeleton-based deformation. Such a treatment avoids iterative match-and-deform calculations and achieves efficient and accurate mesh deformation. In the existing references, mainly four mesh editing algorithms exist. They are: Poisson mesh editing (Yu et al., 2004), Harmonics coordinates (Joshi et al., 2007), Mean Value Coordinates (Floater, 2003), and Laplacian Mesh Editing (Sorkine et al., 2004). Poisson mesh editing is mostly used to create a transition surface to smoothly connect two separate surfaces or models together. Harmonic coordinates have been known to be used in cage-based deformation. In contrast, both Mean Value Coordinates (MVC) and Laplace coordinates have been used to tackle mesh editing problems similar to the one in this chapter. Here both deformation methods are used in Subsections 3.5.2 and 3.5.3 to remove the deformation errors between the created occluding contours and those of the 3D template model after skeleton-based deformations.

3.2.5.1 Skeleton-based deformation: As discussed above, when the pose of the created occluding contours is quite different from the pose of the 3D template model, skeleton-based deformation will be applied first. This involves: 1) creating skeleton of the created occluding contours and the 3D template model, 2) automatically applying geometric transformations such as translation, rotation and scale to align the skeleton of the 3D template model exactly with the skeleton of the created occluding contours, 3) using skeleton-driven skin deformation algorithm to obtain a new mesh shape deformation caused by skeleton movements. For the created occluding contours shown in Figure 26(b) and the 3D template model indicated in the middle image of Figure 26(b), the created skeletons are depicted in Figure 29. Figure 29(a) indicates the skeleton of the created occluding contours, while Figure 29(b) demonstrates the skeleton of the 3D template model. After automatically applying geometric transformations, the skeleton of the 3D template model in Figure 29(b) is changed into the skeleton in Figure 29(c) which is well aligned with the skeleton of the created occluding contours in Figure 29(a). A powerful skeleton-based skin deformation technique by Lewis et al. (2000) is employed to determine the new mesh shape of the 3D template model. With this algorithm, the new position of a deformed vertex is determined by transforming the vertex at the initial pose through Equation (3.9) below. At a particular skeletal configuration c, a deformed vertex V_c, can be computed by

$$V_c = \sum_{i=1}^{n} w_i M_{i,c} M_{i,d}^{-1} V_d \tag{3.9}$$

where w_i are the weights, V_d is the location of a vertex at its initial pose, $M_{i,c}$ denotes the transformation matrix associated with the ith joint in configuration c i.e. it is the transform from the surface containing V_d to the world coordinate system. $M_{i,d}^{-1}$ is the inverse of the transformation matrix associated with the ith influencing joint in configuration d. Configuration d pertains to the original joints positions and weights as shown in Figure 29(b), while configuration c pertains to the final joints positions as shown in Figure 29(c). The weights were determined based on the number of polygons and the efficiency of this algorithm directly proportional to the polygon count. The deformation is controlled by the user through the weights w_i. It was observed that weights lying in the region of 0.05 to 0.10 produces satisfactory results.



(a) (b) (c)

Figure 29: Skeleton-based deformation: a) Skeleton of the input sketch, b) Skeleton of the 3D template model, c) The new mesh shape of the 3D template model after geometric transformations.

Using Equation (3.9), the new mesh shape of the 3D template model is obtain whose skeleton has been aligned with the skeleton of the input sketch contours in Figure 29(a), and depicted it in Figure 29(c) as well. In order to align the template skeleton with the input sketch skeleton, the system requires the user to select only the joints in the input sketch skeleton and its corresponding joint in the template skeleton. This manual selection of corresponding joints is the only constraint for skeleton-driven deformation process. It can be seen that the new mesh shape of the 3D template model is very close to the input sketch contours, leaving very small deformation errors to be removed by mesh editing to be examined in the following two subsections.

3.2.5.2 Final Deformation using Mean Value Coordinates: The proposed approach has used the algorithm presented in (Floater, 2003) to compute the final mesh deformation. For a mesh with vertices V and edges E the Mean-Value Coordinates for each vertex $v_i \in$

V is computed from the Euclidean coordinates of the vertex and its m neighbour vertices v_j , where $(i, j) \in E$.

$$v_i = F_i(V) = \sum_{(i.j)\in E} w_{ij} \left[v_j - \left(d_i + v_j \bullet n_i\right)n_i\right] + h_i n_i \qquad (3.10)$$

Here $F_i(V)$ is the function over the set of all vertices and is subsequently used in Eq (3.10) for energy minimization. n_i is the estimated normal at vertex v_i and is computed as a function of the neighbour vertices. h_i is the vertex offset above the projection plane. w_{ij} are the weight functions and normalizing each weight function by the sum of all weight functions gives us the Mean Value Coordinates. The weight functions determine the amount of deformation to be achieved taking into account the distance between the sketch and the skeletally deformed model. d_i is the average distance from the origin. To achieve final deformation of the mesh, the following energy minimization deformation functional is solved using Gauss-Newton algorithm:

$$\arg\min_V G(V) = \frac{1}{2}\sum_{v_i\in V}(v_i - F_i(V))^2 \qquad (3.11)$$

The computation of MVC based deformation is iterative. For the mesh containing between 35,000 and 45,000 polygons such as the one shown in Figure 28(a) it takes a maximum of 5 iterations to achieve reasonable deformations with each iteration taking between 0.01 to 0.03 seconds. The value of h_i is fixed as 0.01 in each iteration. The mesh shape of the 3D template model after applying Mean Value Coordinates is given in Figure 30(c) where Figure 30(a) is the input sketch, and Figure 30(b) indicates the skeleton of the input sketch.

3.2.5.3 Final Deformation using Laplacian Mesh Editing: The Laplacian coordinates preserve the local geometry of the mesh after deforming it. The Laplacian coordinates of each vertex are defined as a displacement vector between the average of the neighbour vertices and the actual 3D position of the vertex. Using such Coordinates, editing of meshes is highly efficient and fast, since Laplacian Mesh Editing (LME) only requires solving a simple linear system (Sorkine et al., 2004).

For each vertex its differential coordinate is represented by the difference between its position x_i and the average position of its neighbours

$$d_i = L(x_i) = x_i - \frac{1}{n}\sum_{j \in N_i} x_j \qquad (3.12)$$

A mesh can be described by a vector of differential coordinates of all its vertices $D = \{d_i\}$. D can be calculated by multiplying a constant coefficient sparse matrix L (each vertex only interacts with its local neighbors) with a position vector X (a vector of positions of all the vertices $X = \{x_i\}$), i.e., $D = LX$.

The selection of curve points $x_i, i = \{1, \dots, n\}$ in the region of interest (ROI) is automatically done and the algorithm sets the target position p_i for each curve point. The mesh is then deformed according to this setup along with preserving its geometry. The mesh shape of the 3D template model after applying Laplacian Mesh editing is shown in Figure 30(d). Laplacian Mesh Editing will use the sketch points as the handle to deform the mesh.

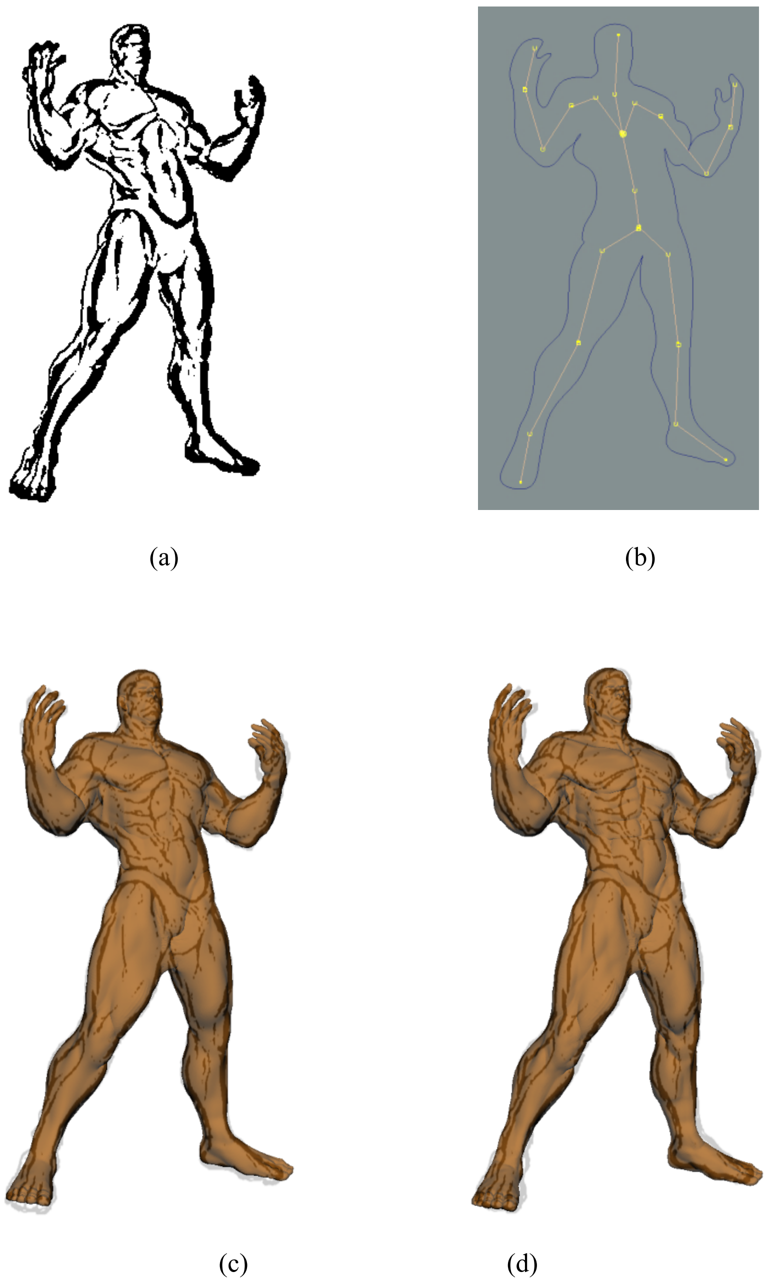
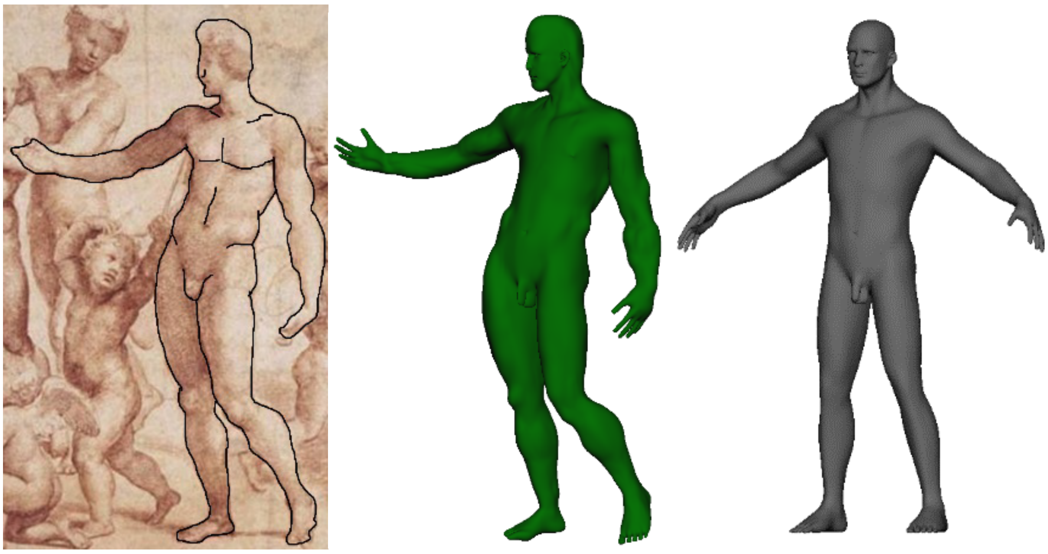


Figure 30: Results showing comparison between Mean Value Coordinates vs Laplacian Mesh Editing (a) Input sketch, (b) Skeleton of the input sketch, (c) Final deformed model using Mean Value Coordinates showing exact matching, and (d) Final deformed mesh using Laplacian Mesh Editing showing exact matching.

With the proposed approach, the choice of final deformed model depends on the requirements of the artist. If the focus is more on the speed of generation of character models, then MVC is the preferred technique for final mesh deformation and thus the model in Figure 30(c) is the preferred model. However aesthetically speaking, the final model in Figure 30(d) is a better model.

3.4 Results and Comparison

In this section some more results are presented that were obtained from using MVC and LME on a mesh of more than 30K polygons (Figure 31 and Figure 32).

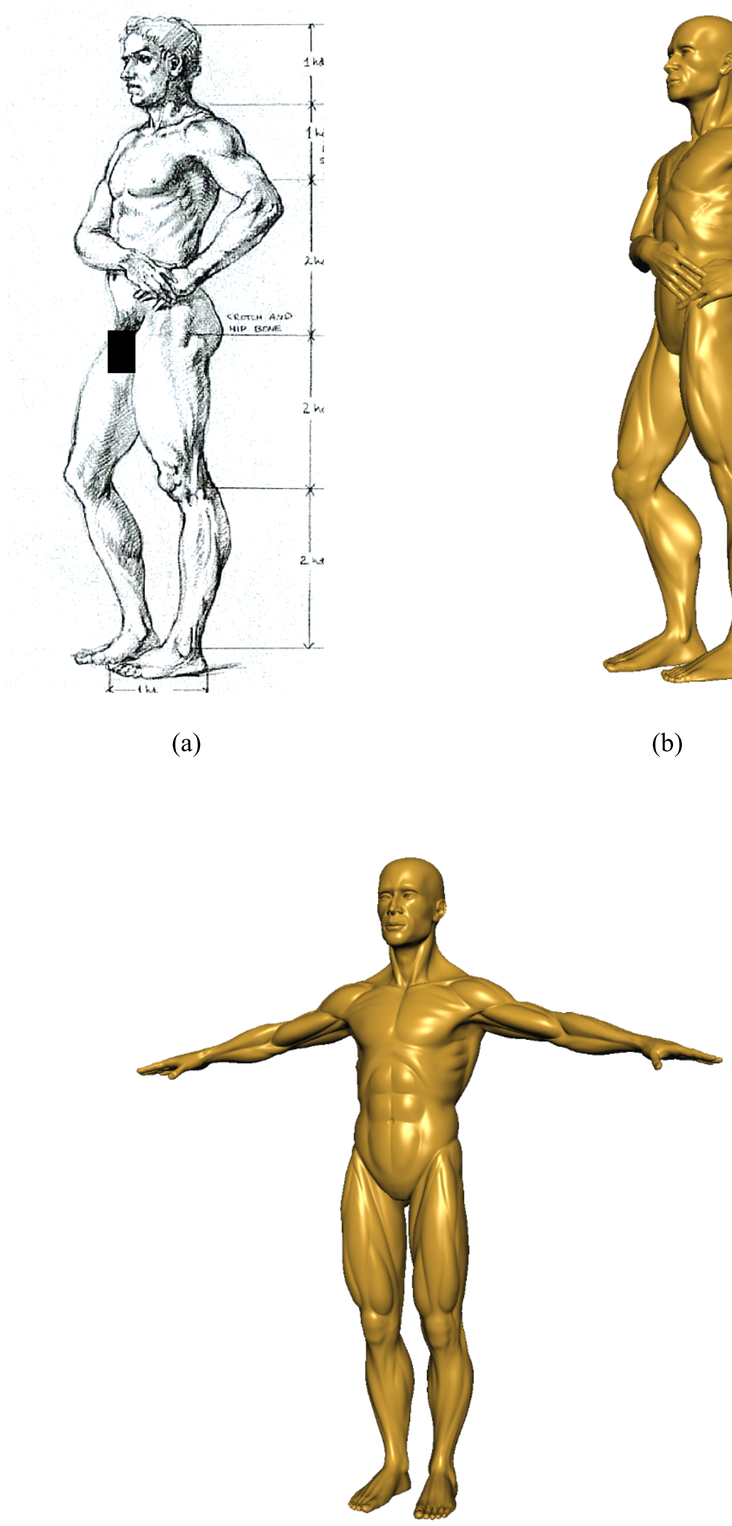


(a) (b) (c)

Figure 31: Another result of the proposed approach (a) Input sketch, (b) Final deformed model using Laplacian Mesh Editing, and (c) Template model.

A comparison of the total time durations of computing the final deformed mesh (Figure 30) using Kraevoy's method (Kraevoy et al., 2009) and the proposed approach is presented in Table 3.

As mentioned by the authors in (Kraevoy et al., 2009), the HMM computation takes up to two seconds on a mesh of more than 20K polygons to compute correspondences. The proposed method takes much less than a second to compute the correspondences on a mesh of more than 30K polygons. The bottleneck of Kraevoy's method is the actual deformation which incurs a total delay of two to ten minutes on a full-resolution mesh. The proposed method on the other hand is considerably faster in deforming the mesh as demonstrated in Table 3.



(a) (b) (c)

Figure 32: Another result of the proposed approach (a) Input sketch, (b) Final deformed model using Laplacian Mesh Editing, and (c) Template model.

Table 3: Performance comparison between the proposed approach and Kraevoy's method (Kraevoy et al., 2009).

Method/Approach	Polygons	Maximum time taken (in milliseconds)
Kraevoy's method (Kraevoy et al., 2009)	28,000 approx	600000
Proposed approach	39,000 approx	2000 (correspondence+ deformation using MVC)

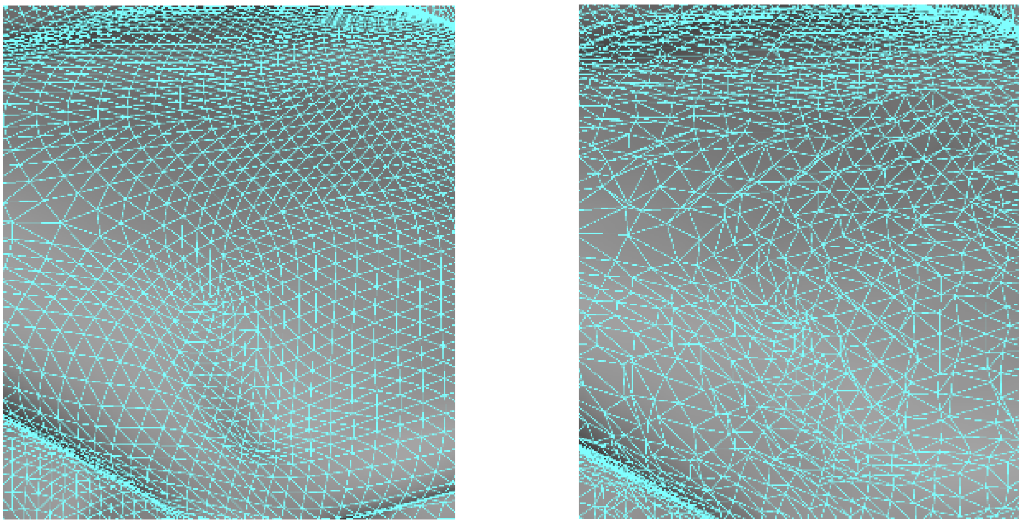
Table 3 shows the time taken to obtain the final correspondence for the models in Figures 30-32, followed by a comparison of the times taken to achieve the final deformed meshes in Figures 30-32 after performing MVC and LME respectively.

Figure 33 demonstrates the difference between the geometry achieved after final deformation of the model in Figure 30, obtained from using Mean Value Coordinates and Laplacian Mesh Editing (Sorkine et al., 2004). Laplacian Mesh Editing was found to preserve the local geometry of the model while Mean Value Coordinates slightly re-arranges the topology and reduces the muscle bulge on the surface. While both algorithms computed final deformation at interactive rates, Mean Value Coordinates was found to be slightly faster in computation than Laplacian Mesh Editing, as shown in Table 4.

Table 4: Correspondence computation and final deformation performance comparison.

Correspondence computation (milliseconds)	**Final deformation using MVC (milliseconds)**	**Final deformation using LME (milliseconds)**	**Model**
836 approx	541 approx	769 approx	 Polygon count: 39,000 approx
810 approx	522 approx	637 approx	 Polygon count: 32,000 approx
1108 approx	825 approx	1095 approx	 Polygon count: 80,000 approx

As apparent from the table above, MVC takes on average less time as compared to LME.



(a) (b)

Figure 33: Close-up of results achieved from LME and MVC (a) Mesh close-up from LME shows it preserves the mesh topology, (b) Mesh close-up from MVC showing re-arranged topology.

Figures 31 and 32 give two more results, which deforms a 3D template model to exactly match the created 2D occluding contours.

3.5 Conclusion and future work

This chapter has proposed a new approach and an interface to create new character models from user's drawn sketches and a 3D template model. This new approach combines human vision, intelligence and interactions with computer's powerful computing capacity to achieve accurate and quick correspondences between the 3D template model and user's drawn sketches, and presents a hybrid mesh deformation technique to change the 3D template model into new character models efficiently. The presented hybrid mesh deformation technique maximizes the strength of skeleton-based deformation in dealing with large mesh displacements and deformations globally and that of mesh editing algorithms in tackling small mesh deformations locally. The developed user interface integrates all the functions ranging from creating a character dataset, interactive sketch creation, retrieval of a most similar 3D template model, fast and accurate correspondences, hybrid mesh deformation. Figures 34-37 shows some interfaces of the proposed system. The system is an extension of the Real Time Suggested Contours system developed by DeCarlo et al. (2003) available at http://gfx.cs.princeton.edu/proj/sugcon/

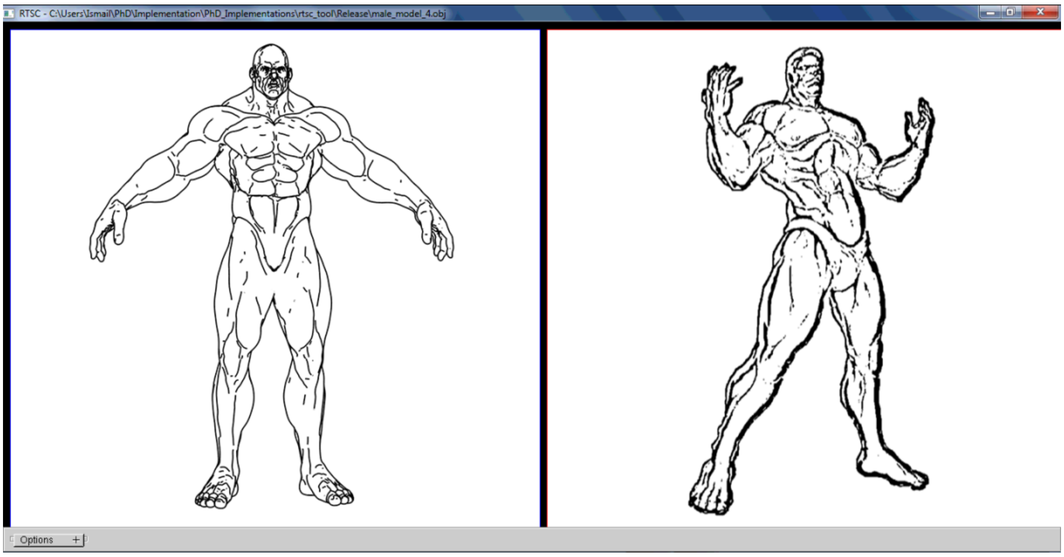


Figure 34: The interface showing the input sketch (right panel) and the most closely matching 3D character with contours renderings (left panel).

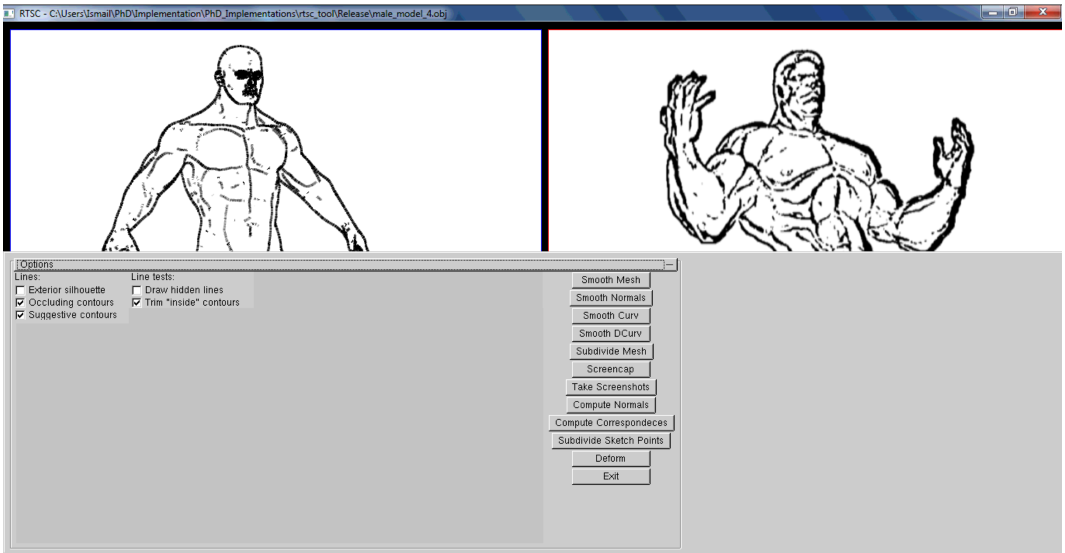


Figure 35: Tools provided by the system.

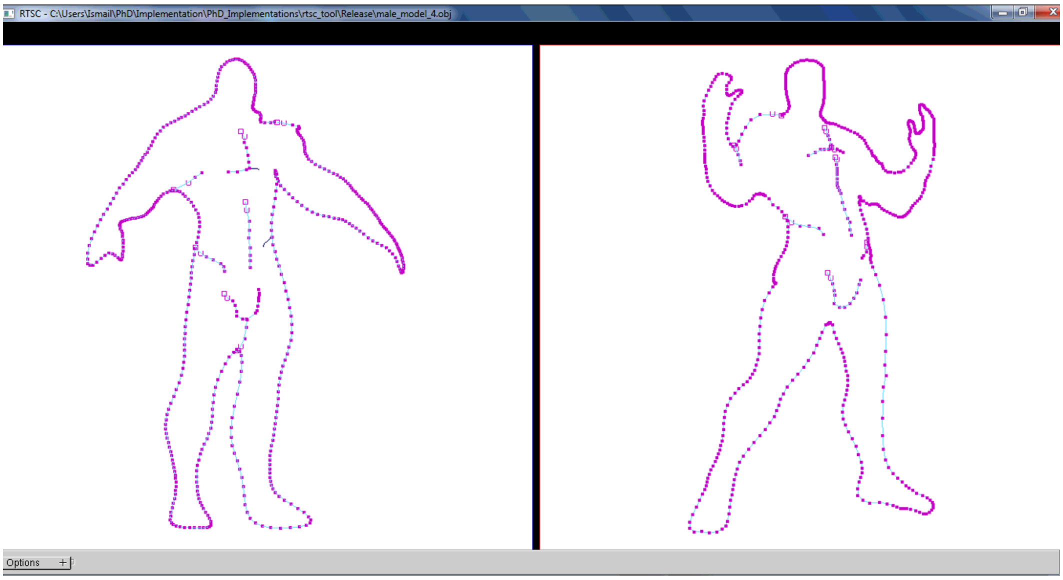


Figure 36: Interface showing the occluding contours correspondence computation.

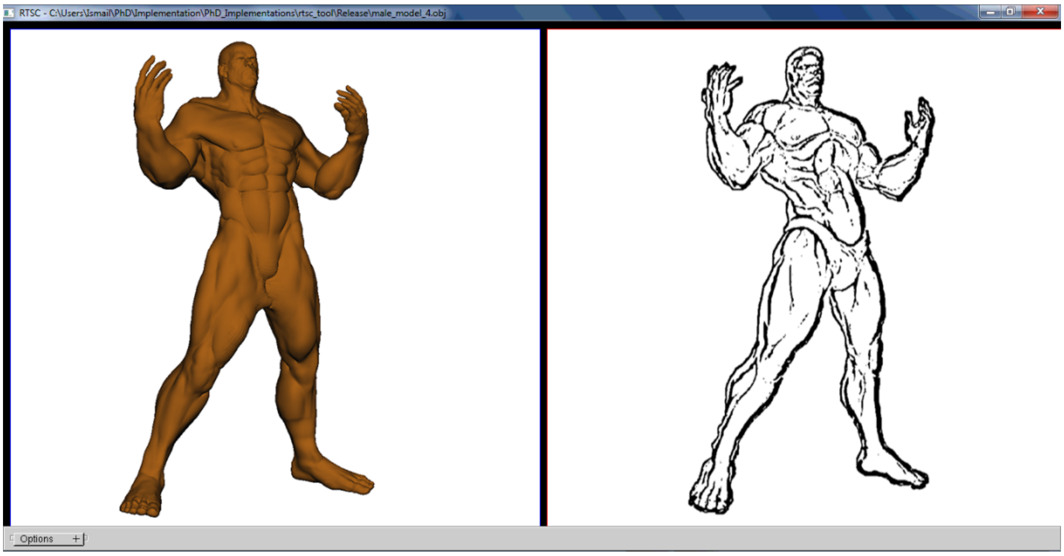


Figure 37: Interface showing the final deformed model alongside the artist's drawn sketch.

The proposed technique has some limitations. First, the current dataset only includes human character models. In future, there is a possibility to evolve the database into a large version involving various human and non-human character models. Second, manually drawing occluding contours is slightly difficult for the first time users. This can be avoided by developing an automatic algorithm to generate the occluding contours.

4 A Hybrid Approach for Character Modelling Using Geometric Primitives and a Shape-from-Shading Algorithm

4.1 Introduction

Organic modelling of 3D characters is a challenging task when it comes to correctly modelling the anatomy of the human body. An artist views the human body in the form of several parts which can be modelled using simple primitive objects. For examples limbs (arms and legs) can be approximated with cylinders, head can be approximated with spheres or ellipsoids. However some artists make character sketches in which the characters are wearing armour and clothes. A new primitive 'cube' is proposed in this work, which can be used to model armour and clothes for characters. Figure 38 shows an example of a female warrior model created by the proposed system in which the armour of the model was created using a cube primitive.

Although the deformed 3D models achieved in the previous chapter can possess high number of polygons and look aesthetically detailed, the approach described in the previous chapter is not suitable to generating models with even finer surface details such as skin wrinkles etc. The technique proposed in this chapter can be augmented to the approach in the previous chapter thus resulting in even better quality character models.

Figure 38: A female warrior model with her armour modelled using a cube primitive.

Most sketch-based modelling approaches generate base mesh models only without details. Realistic models require details on their surface. To create more realistic looking models, artists tend to use sculpting packages such as Pixologic ZBrush (ZBrush), or Autodesk Mudbox (Mudbox). These packages provide tools which are often difficult to learn and master by novice artists, and are not designed to be used in a more natural way. Thus the artist is required to give a keen attention to detail, often making the entire process very laborious and time consuming. These packages also require computers with high specifications or a GPU with a considerable amount of memory, as most of the features are designed to be executed on the GPU, thus making them less favorable for 3D artists focusing on low resolution 3D graphics for indie games.

Shape-from-Shading (SFS) provides a promising and efficient method to create details from images. The idea for reconstructing 3D surface details from images and combining it to a low-polygon base mesh comes from the philosophical work by

Koenderink (1998). In the paper, the author discusses in detail the philosophy behind generating 3D surfaces from shaded / lighted images. Several SFS techniques have taken inspiration from Koenderink's work such as (Wu et al., 2007), (Hahn et al., 2013), and (Wu et al., 2008).

The proposed hybrid approach solves the problem of producing better quality character models than the previous sketch-based modelling (SBM) approaches (Gingold et al., 2009), through easy inputs by integrating SBM approaches given in (Gingold et al., 2009) with the shape-from-shading algorithm in (Yang and Han, 2007). The work presented in (Gingold et al., 2009) and (Yang and Han, 2007) has inspired me to combine them together to produce a hybrid technique leading to better results.

The main contributions in this chapter are as follows:

1. A novel and hybrid approach is proposed, which harnesses some of the techniques of Gingold et al. (2009), and Yang and Han (2007) to generate good quality character models through sketching. The previous SBM techniques that rely on creating models directly from sketches without using a template model from database (Gingold et al., 2009) generate simple base mesh models only, thus limiting the artist to create detailed models.

2. In addition to the generalized cylinder, the approach provides artists two additional tools, i.e., cube and ellipsoid as geometric primitives to create base mesh models quickly and easily.

3. Instead of requiring the artist to provide a template source mesh as input as in (Takayama et al., 2011), the proposed approach instantly generates relief mesh using shape-from-shading algorithms, which can be easily transferred to the target base mesh.

4. Utilizing the technique in (Yang and Han, 2007), surface details are generated from a single picture or sketch, and provide tools to interactively add the details to the base mesh.

4.2 Related Work

Sketch-based modelling systems can be roughly categorized in hard surface modelling, and organic modelling systems. Hard surface modelling systems were developed to aid architectural or mechanical 3D models. Notable examples of sketch-based hard surface modelling systems are Trimble SketchUp (Sketchup), (Schmidt et al., 2009), and (Rivers et al., 2010). On the other hand, organic sketch-based modelling systems provide tools to create character models mainly using feature curves, and suggestive contours (DeCarlo et al., 2003) etc. Organic modelling systems provide tools to the artists to create 3D models from simple primitives such as ellipsoids and using inflation techniques to inflate a closed 2D region / sketch (e.g. a circle, oval etc.) such as Teddy (Igarashi et al., 1999), and FiberMesh (Nealen et al., 2007). In (Gingold et al., 2009), the authers used generalized cylinders and ellipsoids for modelling of organic models directly from a single view model, and provided annotations to artists so they can manipulate the models easily and quickly. However their system provides a limited set of tools to the artist for modelling and only provides two primitives (cylinders and ellipsoids). Most artists make use of more primitives to model a human character. One important primitive used by artists is the box primitive as demonstrated in the tutorial video by the leading comic artist Stan Lee (webfox100, 2009). In this tutorial, he tells us how easily a human body can be decomposed into simple primitive geometric shapes.

In FiberMesh (Nealen et al., 2007), the authors have proposed a system to add details to simple 3D models. However, using this system to add finer details is not easy and requires great attention to detail for the novice artist. Moreover very subtle details are not very easy to add.

Shape-from-Shading (SFS) has undergone considerable amount of research in the past decades. Several excellent surveys exist on SFS such as (Zhang et al., 1999) and (Durou et al., 2008). SFS algorithms have immense applications not only in the animation/gaming industry but also in archeological research where scientists are reconstructing ancient artifacts and base reliefs to preserve them in a digital form such as Project Mosul (http://projectmosul.org/). Hahn et al. (2013) has proposed a two-step method for surface reconstruction using 2D strokes and a vector field on the strokes. They have used TV (Total Variation) and H1 regularization with a curl-free constraint for obtaining a dense vector field and using this dense vector field to obtain the final height map. Total variation regularization is a process, most often used in digital image processing, that has applications in noise removal, and it has been found to be more effective than linear smoothing or median filtering (Rudin et al., 1997). However this method involves solving energy minimization functionals, is very computation intensive, and requires complex GPU implementation to speed up the computation. This method also involves solving energy minimization functionals.

Lee and Kuo (1993) used the brightness constraint and the smoothness constraint. Surfaces were approximated by the union of triangular surface patches. The vertices of the triangles were called nodal points and only nodal depths were recovered. They have used interpolation to recover depth at the pixels. For each triangular patch, the intensity of the triangle was taken as the average intensity of all pixels in the triangle and the surface gradient of the triangle was approximated by the cross product of any two adjacent edges of the triangle. This established a relationship between the triangle's intensity and the depth at its three nodal points.

Tsai and Shah (1994) presented a very fast and simple algorithm for computing the height map from a single greyscale image. Their approach uses a linear approximation of reflectance in z axis, and the results are very convincing. In (Sun et al., 2009) the authors have presented a new algorithm for base relief reconstruction using Adaptive Histogram Equalization, which optionally uses a template model to compute the height fields via orthogonal or perspective projection. The shape features of the base relief are enhanced by using gradient scaling factors. The results of this approach are in general better looking than other techniques. However as pointed out by the authors, the basic drawback of this technique is without optimization it is time consuming taking around an hour to process a high resolution photograph.

Several researchers have addressed the problem of stitching/transferring details from one mesh to another mesh. To stitch the details onto the base mesh, the proposed hybrid approach utilizes the Discrete Exponential Map (DEM) algorithm first proposed by Schmidt et al. (2006) and then generalized by Takayama et al. (2011). In (Qian et al., 2015) the authors have proposed a novel approach for mesh cloning approach based on pyramid spherical coordinates driven by boundary loop, which extends an existing algorithm for computing offset membrane on mesh. The source and target meshes are mapped onto a 2D parametric domain using geodesic polar maps. During cloning, the boundary loop of the region of interest (ROI) on the target mesh is fitted in real time by B-spline curve to register the boundary loop of the source ROI. Via the reconstructed boundary loop, the ROI is deformed to register the target mesh by pyramid spherical coordinates to ensure that the clone result is seamless and natural.

For modelling a base mesh model, the proposed system mainly draws inspiration from Gingold et al (Gingold et al., 2009). In addition to this, the proposed system aims at enhancing the system by providing a new shape primitive for modelling i.e. cube. It uses the same definition for creating the generalized cylinder as well as the ellipsoid.

Moreover the proposed approach uses the technique in (Yang and Han, 2007) to generate a 3D surface from purely 2D images by computing the height values at each pixel. The authors in (Yang and Han, 2007) have proposed a novel SFS method based on hybrid reflection model which contains both diffuse reflectance and specular reflectance. According to the authors, when discrete characteristic of digital images is considered, finite difference approximates differential operator. The reflectance map equation described by a partial differential equation (PDE) turns into an algebraic equation about the unknown surface height. Then the proposed approach interactively adds details to the bash mesh created using geometric primitives to generate plausible character models. Thus the system presented in this chapter makes some improvements by providing base mesh modelling, detail generation, and detail integration in the single interface.

4.3 System Overview

By combining the techniques presented in (Gingold et al., 2009) and (Yang and Han, 2007), a modelling system has been developed and presented in this chapter, which is composed of two main modules: 1) Base Mesh Modeller, and 2) Surface Detail Modeller. This section explains all the modelling primitives and the controls the proposed system provides to manipulate/modify the primitive models to depict different parts of a knight model.

4.3.1 Base Mesh Modeller

This module provides three primitive shapes to allow artists to quickly model a base mesh from a single view. The proposed system provides the tools which are simple to use and quickly models the primitives using sketching input/gestures. The primitive shapes provided are 1) Generalized Cylinders, 2) Cubes, and 3) Ellipsoids. The artist can choose from these primitives to model different parts of the character. To create a primitive inside the window, the artist first selects the primitive from the toolbox, and draws a stroke in the main window. The system automatically creates a primitive with default size. The system also automatically connects two primitives together. To manipulate the shape of a primitive, the system provides several controls to the artists as discussed in Section 3.1.4. Figure 39 demonstrates a bodybuilder character modelled using the base mesh modeller of the proposed system.

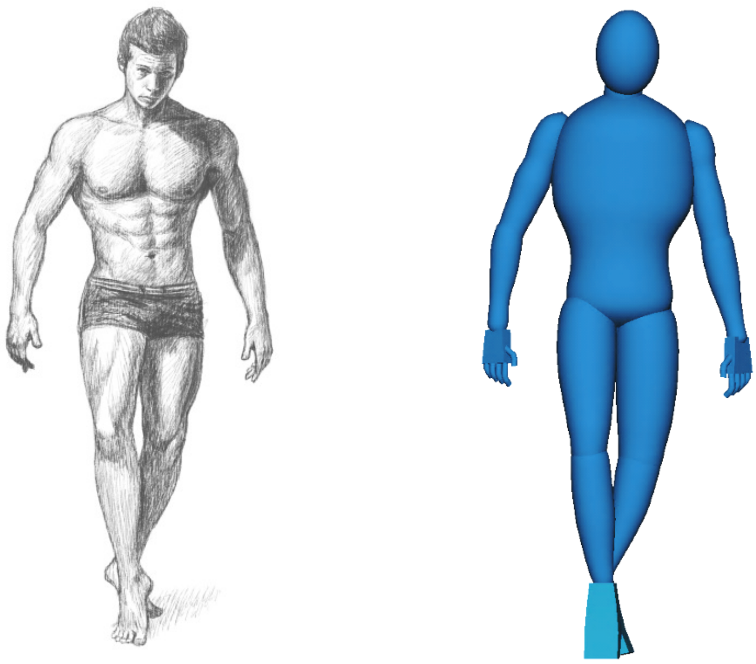


Figure 39: Bodybuilder character created using the proposed system's base mesh modeller from a single sketch. *Sketch retrieved from:* *http://luneder.deviantart.com/art/Male-Figure-sketches-1-410274631*.

4.3.1.1 Generalized Cylinder: A parametric definition of the generalized cylinder is used here as proposed in (Gingold et al., 2009). A generalized cylinder is also known in research literature as a 'sweep surface' (Chen et al., 2013), (Shtof et al., 2013). It is generated when a cross section shape is swept across a spine curve. A spine curve is a B-spline curve defined by a set of control points. The equation of the surface of the generalized cylinder is defined as:

$$P(t,r) = p(t) + v^t(r)n(t) + u^t(r)b(t) \tag{4.1}$$

where $p(t)$is the control point on the curve at the parameter t. $n(t)$ is a unit normal to the curve at the point $p(t)$. $b(t)$ is a vector perpendicular to the tangent vector and the unit normal vector at the point $p(t)$ acquired by the cross product. $v^t(r)$ and $u^t(r)$ are the components of the point on the cross section at the control point $p(t)$.

An ellipse has been used as a cross section curve. A cross section is thus a 2D B-spline curve drawn on a 2D plane perpendicular to every B-spline curve point $p(t)$. The cross section curve is defined at the point $p(t)$ as:

$$u^t(r) = cos\theta(t)s^u(t)u(r) - sin\theta(t)s^v(t)v(r) \tag{4.2}$$

$$v^t(r) = sin\theta(t)s^u(t)u(r) + cos\theta(t)s^v(t)v(r) \tag{4.3}$$

where $s^u(t)$ and $s^v(t)$ are the scaling factors in the u and v coordinates of the cross section curve. θ is the angle of rotation of the cross section.

The end caps of the cylinder are created in tangent continuous fashion, and the cross section scale keeps decreasing as it approaches the end point of the cylinder. The smoothness of the cylinder can be controlled by the user by specifying the number of subdivision steps as explained in Section 4.

Figure 40 shows a Generalized Cylinder. In the figure, the cross section is highlighted in orange color, the end caps are shown with red outlines, while green curve shows the spine curve.

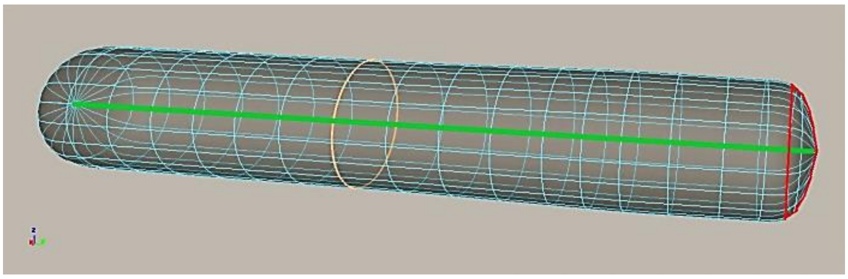
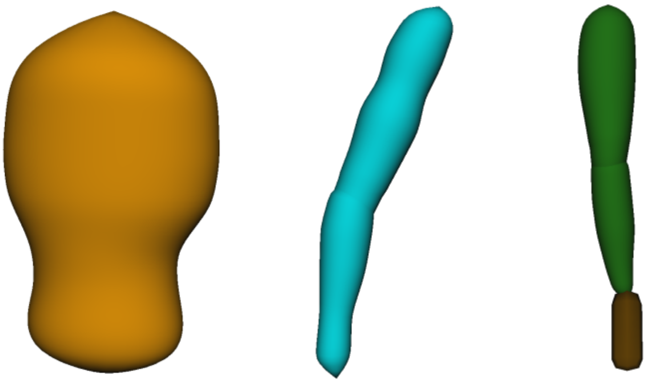


Figure 40: A generalized cylinder.

In Figure 41 the torso, legs and arms of the male bodybuilder character were modelled using a generalized cylinder.



(a) (b) (c)

Figure 41: Body parts modelled using generalized cylinders. (a) Torso, (b) Arm, and (c) Leg.

4.3.1.2 Ellipsoid: A general ellipsoid also known as the triaxial ellipsoid has been used here. Mathematically, it is defined as a quadratic surface given in Cartesian coordinates by:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{4.4}$$

where the semi-axes are of lengths a, b, and c. An ellipsoid can be used to model the head and breasts of a character. An ellipsoid can be selected from the tools bar and dragged onto the canvas area. Here the artist can scale, rotate or translate the ellipsoid to better align with the sketch. An ellipsoid is geometrically akin to the generalized cylinder except that its spine curve length is equal to the cross section radius of the central cross section.

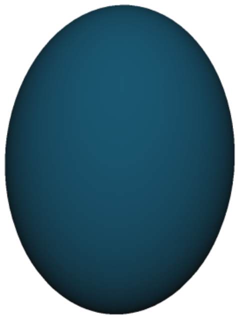


Figure 42: The head of the bodybuilder character was modelled using an ellipsoid.

4.3.1.3 Cube: A cube is a simple geometric structure which can be used to model armour and suits of characters. It can also be used to quickly model shoes of the character. A cube's width can be scaled, and then it can be rotated to form different parts of the suit / armour of character models such as knee pads, belts, shoulder guards etc. Figure 43 shows the shoe of the bodybuilder modelled using a cube.

Figure 43: Shoe of bodybuilder character modelled using a cube.

4.3.1.4 Mesh manipulation/modification controls: The system provides simple controls to the artists to manipulate and modify the mesh geometry. Mesh manipulation controls in the system presented here have been inspired by Gingold et al. (2009), who provided annotations and controls to the user to manipulate the primitives, such as connection curve annotation for connecting two or more primitives, mirror annotation to create a copy of the primitive and reflects it across the symmetry plane of another primitive. A subset of these controls has been in order to match the mesh manipulation requirements.

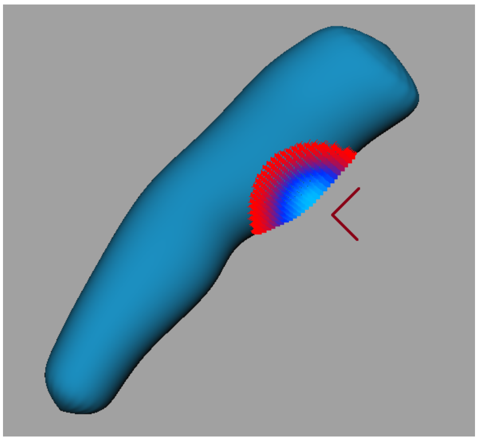


Figure 44: Mesh manipulation. The axes allow the users to pull or push the mesh in the desired direction.

As shown in Figure 44, a user can select one or more points on the surface of a primitive and move them to modify the shape of the primitive. The system allows the user to rotate, translate and scale a set of vertices. When a user selects a vertex on the base mesh, the selected point along with its neighboring vertices is selected to allow the user to modify the mesh in a smooth manner.

4.3.2 Surface Detail Modeller

This is the second module of this system, which allows an artist to generate finer details onto the base mesh using a shape-from-shading algorithm and detail transfer techniques. This module is responsible for producing the final detailed character model.

4.3.2.1 Relief mesh generation via a Shape-from-Shading algorithm: Once the base mesh is created, the system allows the artist to integrate details to the base mesh to give it a more detailed and finished look. To attach surface details to the base mesh, the algorithm first generates a relief mesh from a photograph/sketch. In this step the artist can either provide the photograph/snapshot directly from the input sketch, or retrieve an image from the web which looks similar to the portion of the input sketch the artist needs to use for shape-from-shading. The system utilizes the Shape-from-Shading algorithm proposed by Yang and Han (2007) to reconstruct the surface details separately. The reason for choosing to implement the algorithm in (Yang and Han, 2007) is because it avoids the complex and computationally heavy energy minimization computations and is relatively simple to implement. The SFS algorithm is not meant to separate details from the sketched character but to generate 3D relief details directly from a photograph. Thus this is very easy to perform as it only requires the artist to select a photo from the web that can closely match the details of a particular part of the input sketched character such as armour details and suit details etc. For integrating the details onto the base mesh surface, the system employs the Discrete Exponential Map algorithm (Takayama et al., 2011). This is a semi-automatic method to incorporate the details onto the base mesh, which requires the artist to manually select a region of interest.

The input to this module is a single image or sketch which can be coloured or in grey scale. The idea is to calculate the height values at each pixel using the hybrid reflectance map equation, and iteratively refine the height values.

The hybrid reflectance equation presented in (Yang and Han, 2007) has the form of

$$R(p,q) = (1-\omega)R_d(p.q) + \omega R_s(p,q) \tag{4.5}$$

where $p(x,y) = -\frac{\partial z(x,y)}{\partial x}$ and $q(x,y) = -\frac{\partial z(x,y)}{\partial y}$ denote the x and y partial derivatives of reconstructed 3D surface height $z = z(x,y)$ with respect to the image coordinates x and y. $\omega \in [0,1]$ is the factor of specular component. From this equation, the reflectance map equation is derived, which is given by Yang and Han (2007):

$$(1-\omega)\frac{pp_0+qq_0+1}{\sqrt{p^2+q^2+1}\sqrt{p_0^2+q_0^2+1}} + \omega\left(\frac{pp_h+qq_h+1}{\sqrt{p^2+q^2+1}\sqrt{p_h^2+q_h^2+1}}\right)^K = \frac{I(x,y)-I_{min}}{I_{max}-I_{min}} \tag{4.6}$$

where I_{max} and I_{min} are the acquired maximum and minimum intensities of image $I(x,y)$. This SFS algorithm is fast to compute. The authors have used a hybrid reflectance model which is more prone to real reflectance than Lambertian shading model or Torrance-Sparrow shading model (Torrance and Sparrow, 1967), which was used in previous SFS algorithms. The intensity gradient is in the direction that the shape of surface changes most, so the directional derivative of an image is used as parts of an objective function in the algorithm. The reflectance map equation described by PDE turns into an algebraic equation about the unknown surface height, which is easy to compute.

The aim is to compute the z values at each pixel. The iterative formula used by Yang and Han (2007) is given as:

$$
\begin{aligned}
&z_{i,j}^{(k+1)} = z_{i,j}^{(k)} \\
&+ \mu \times F(z_{i-2,j}^{(k)}, z_{i-1,j}^{(k)}, z_{i+1,j}^{(k)}, z_{i+2,j}^{(k)}, z_{i,j-2}^{(k)}, z_{i,j-1}^{(k)}, z_{i,j}^{(k)}, \\
&z_{i,j+1}^{(k)}, z_{i,j+2}^{(k)}, z_{i-1,j-1}^{(k)}, z_{i-1,j+1}^{(k)}, z_{i+1,j-1}^{(k)}, z_{i+1,j+1}^{(k)})
\end{aligned} \quad (4.7)
$$

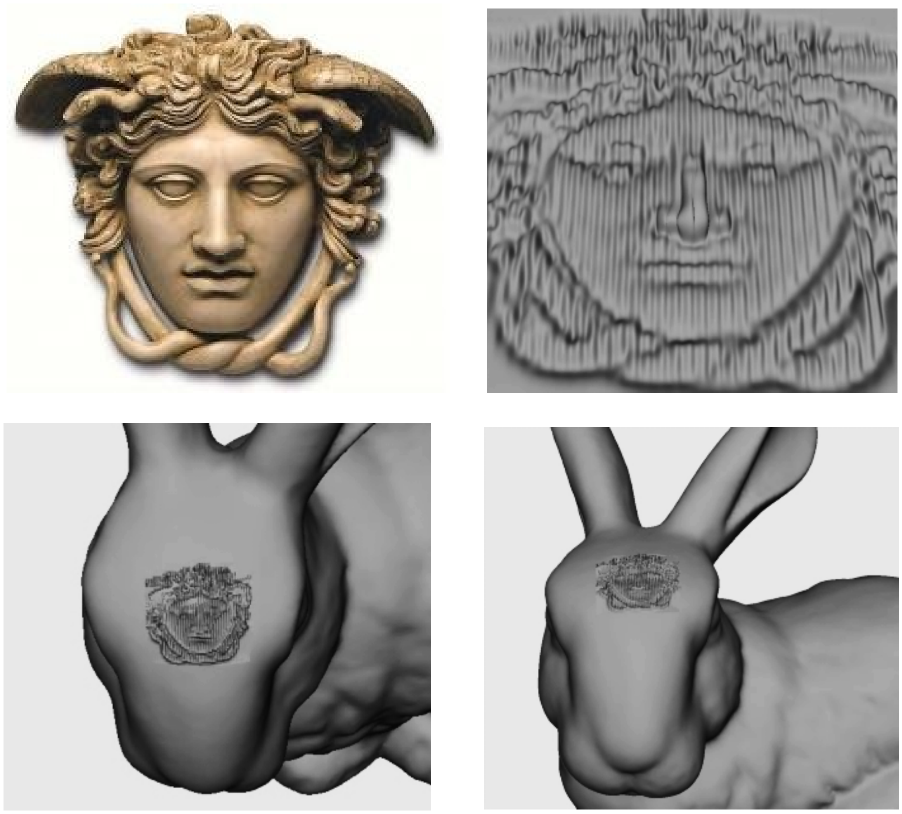


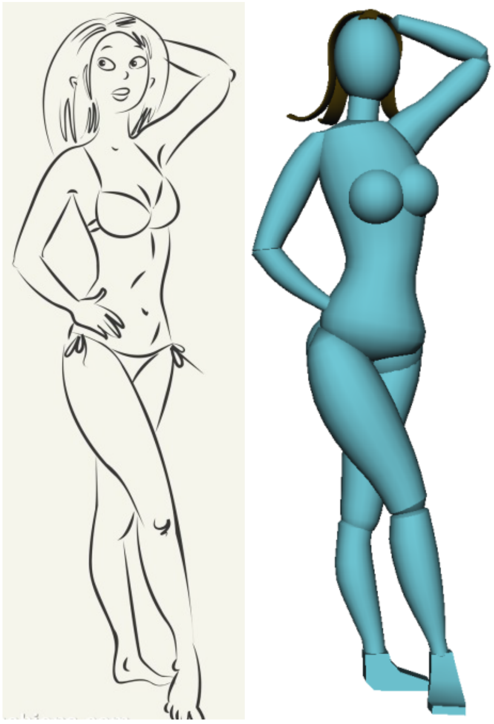
Figure 45: Medusa relief (top left). A detailed surface of medusa relief (top right). Show detailed surface added to the head of the bunny (bottom images). *Image retrieved from Medusa image: http://www.belladonna.de/haupt-medusa-p-2678.html.*

Here $k = 1,2,\ldots$ is taken as the iterative time, and μ is the iterative rate. The authors have suggested to stop the computation when an iterative time limitation is arrived or error criteria are satisfied. However the experiments for the system have shown that the height map achieved after the 5th iteration is acceptable enough. The detailed formulae for the second term in the above equations are given in the paper by Yang and Han (2007).

Figure 45 demonstrates a 3D surface generated from a Medusa relief and then the details transferred to the bunny model.

4.3.2.2 Detail transfer from relief mesh to target base mesh: For this part of the system, the GeoBrush algorithm as presented in (Takayama et al., 2011) was utilized to transfer the details from a source mesh to the target base mesh in a semi-automatic manner. In this semi-automatic and interactive step, the system first increases the number of polygons of the target base mesh to approximately match with that of the source relief mesh, by subdividing the target mesh. The system then requires the artist to select a region on both the reconstructed mesh and the target mesh (obtained from shape-from-shading) using painting. Then a rough correspondence between the source and target meshes is established, as the topology of both the surfaces can vary. The system then computes the parameterization and mapping from the source mesh to the target mesh using the Generalized Discrete Exponential Map algorithm as introduced in (Takayama et al., 2011), which is a generalized version of the DEM algorithm presented by Schmidt et al. (2006). For any surface, DEM is a parameterization of the surface centered at a point p on it. The system finds the geodesic distance curves originating from p following the same origin as the tangent plane at p, using Djikstra's shortest path algorithm.

Figure 46 shows a female character modelled using the base mesh modeller and then detailed using the SFS algorithm. In Figure 46(b), the entire base mesh of the female character was modelled using geometric primitives. The torso, arms, hair and legs were modelled using generalized cylinders. The head and breasts were modelled using a spheroid. The feet were modelled using a cube.



(a) (b)

Figure 46: Female character base mesh (a) Female character sketch, and (b) Completed model created using geometric primitives. *sketch retrieved from www.sketchesfashions.com.*

After the artist has completed the base mesh of the model, he/she can proceed with adding the details to this model. Figure 47 shows the details of the bra generated using an image and transferred onto the breasts base mesh of the female character in Figure 46.

(a) (b)

Figure 47: (a) Details of the bra generated using an image, and (b) Transferred onto the breasts of the female character above.

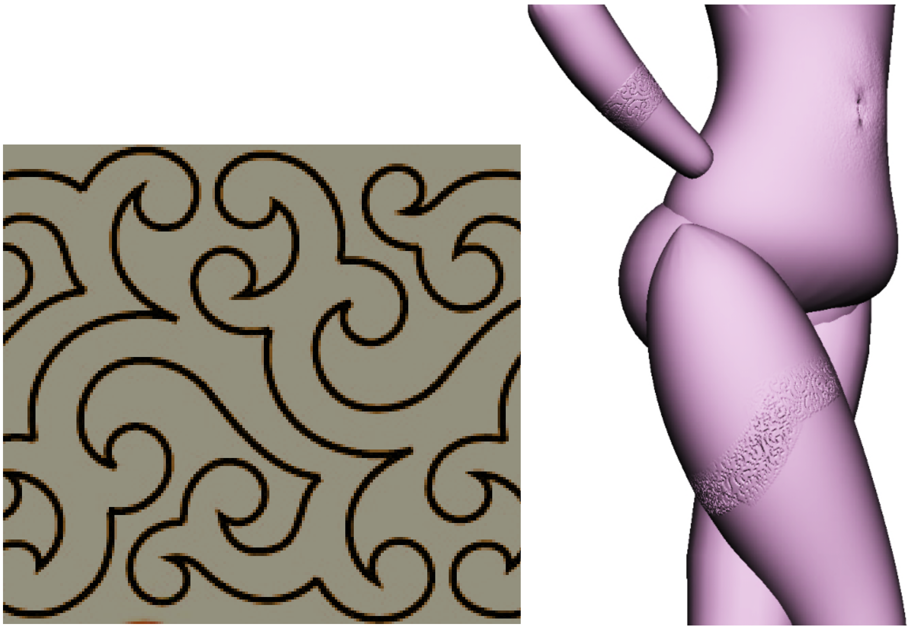
Figure 48 shows the details of the abdomen generated using an image and transferred onto the lower torso base mesh of the female character in Figure 46.

(a) (b)

Figure 48: (a) Details of the abdomen generated using an image, and (b) Transferred onto the base mesh of the female character base mesh in Figure 46.

Figure 49 shows the details on the right arm and leg of the female generated using an image and transferred onto the base mesh of the right arm and leg of the female character in Figure 46. As shown in the figure, the texture pattern repeats in a tiled fashion. The algorithm tiles the texture pattern over the area selected by the user on the target mesh, however it was observed that some texture overlaps occurred over the area selected, which is a limitation of the proposed approach. Figure 50 shows the final female character with necessary details added to the base mesh.

The geometric detail transfer algorithm discussed in this section can also be used to enhance the approach presented in the previous chapter in order to create more realistic models by transferring the geometric details to the surface of a final deformed model.



(a) (b)

Figure 49: (a) Details of the right arm and leg of the female generated using an image pattern, and (b) Details transferred onto the right arm and leg base mesh of the female character.

Figure 50: Detailed model using SFS algorithm.

4.4 Implementation

The proposed system has been implemented in C++ on an HP Z400 workstation with Intel Xeon CPU 3.33Ghz, Nvidia GeForce GTX 560Ti, 8GB RAM and running Windows 7 64-bit.

Figures 51-55 shows some interfaces that were developed to test the proposed system. As can be seen from the images, a model is being constructed using the proposed system.

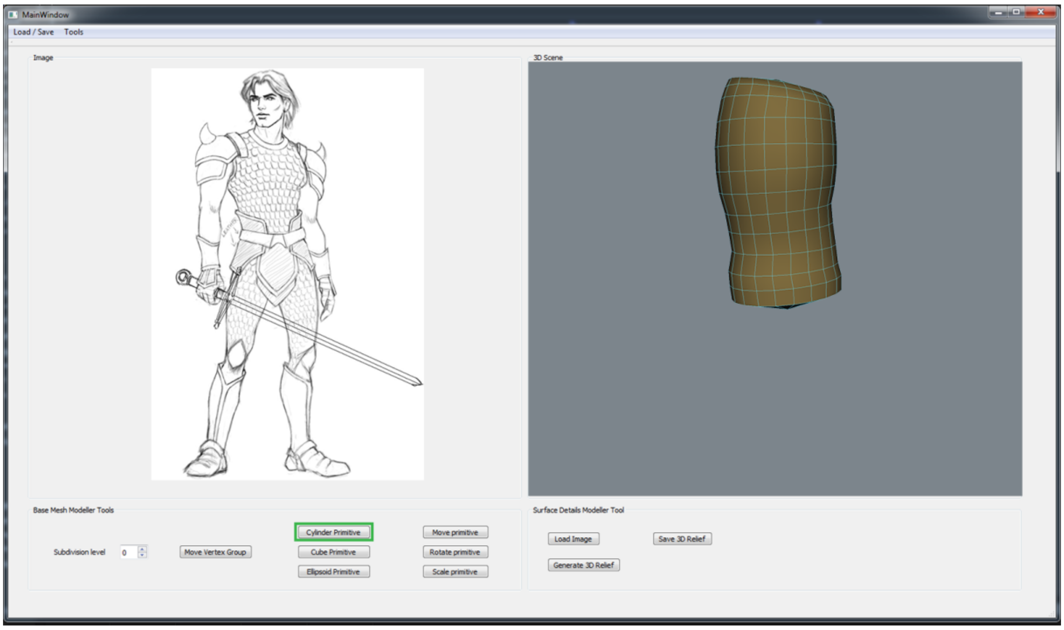


Figure 51: Cylinder primitive being used to model the torso of the human character.

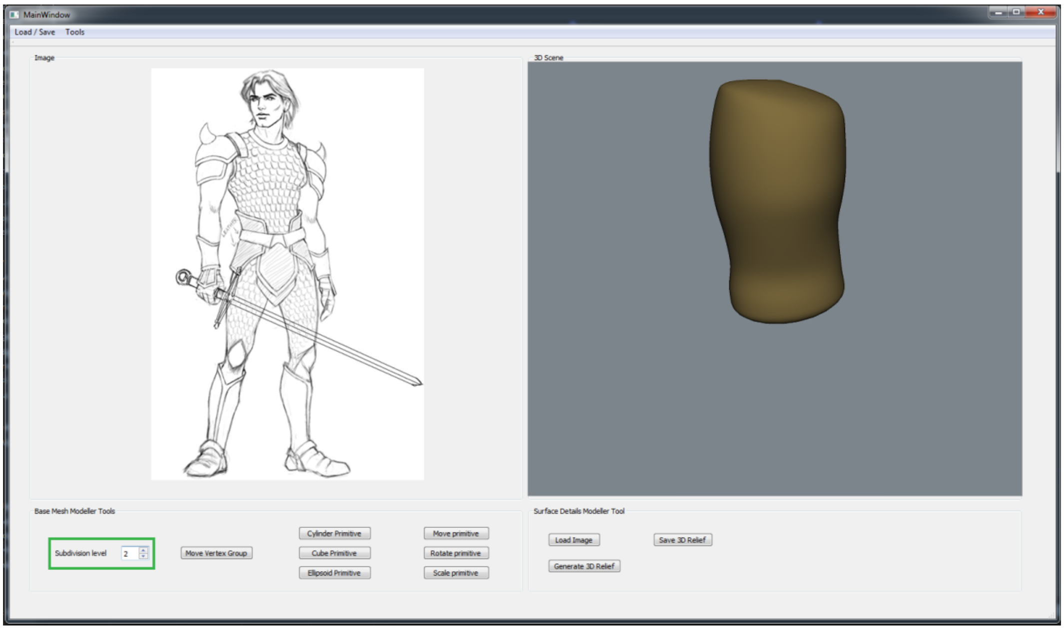


Figure 52: Setting the subdivision level to achieve a smooth looking torso.

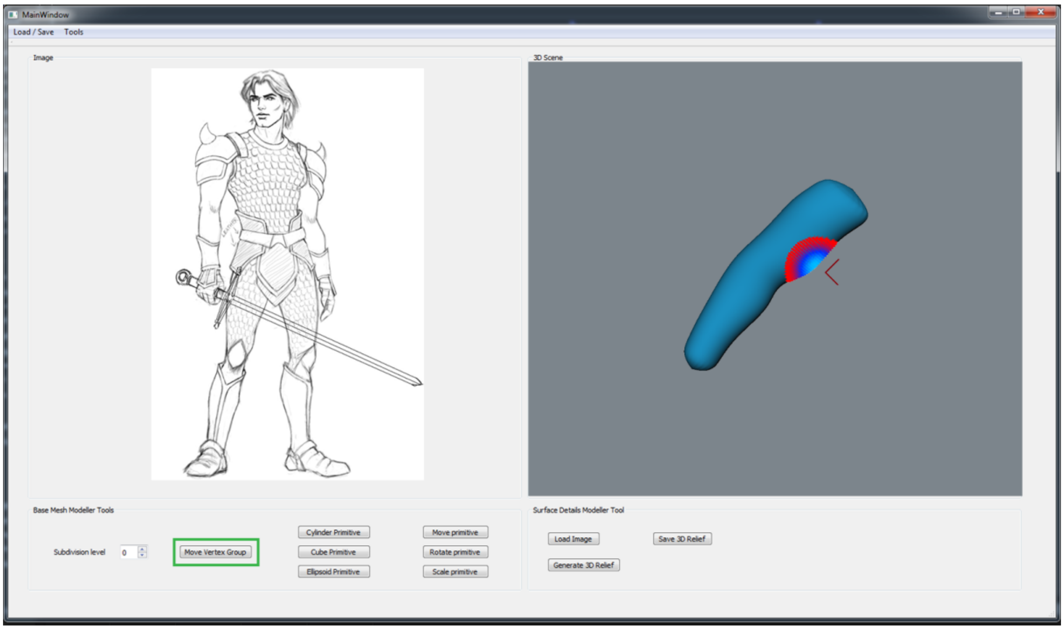


Figure 53: Using the mesh modification tools to model the arm of the character.

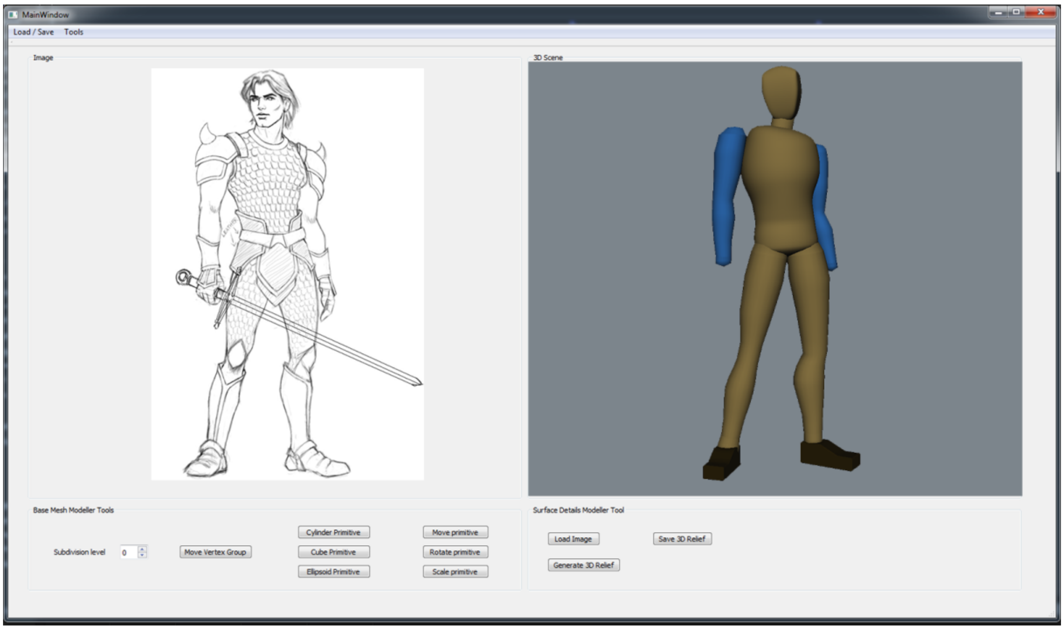


Figure 54: Right panel (3D Scene) showing the base mesh of the character.

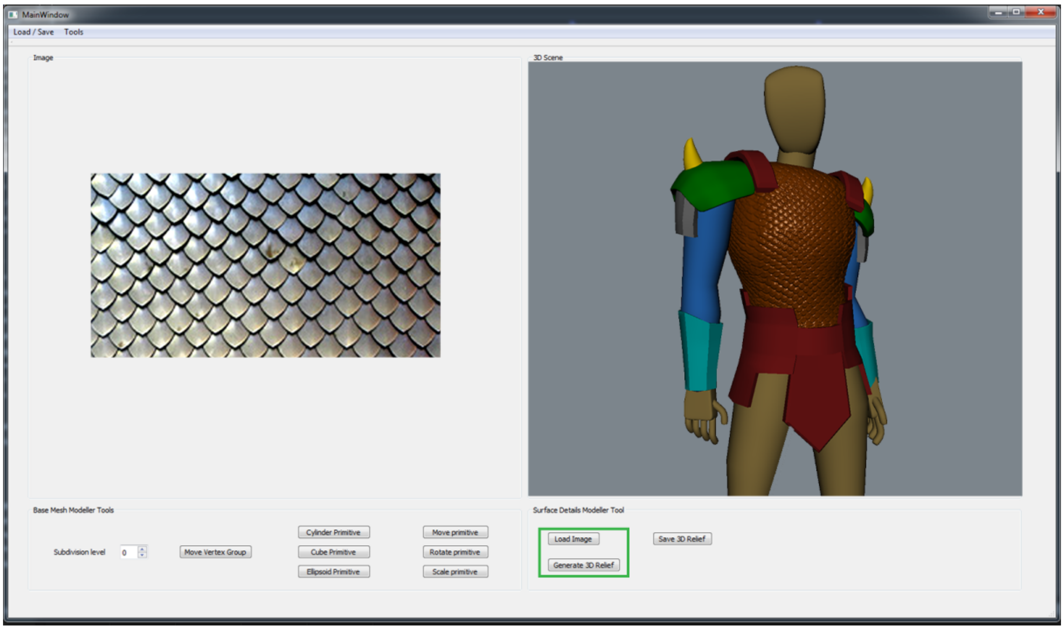
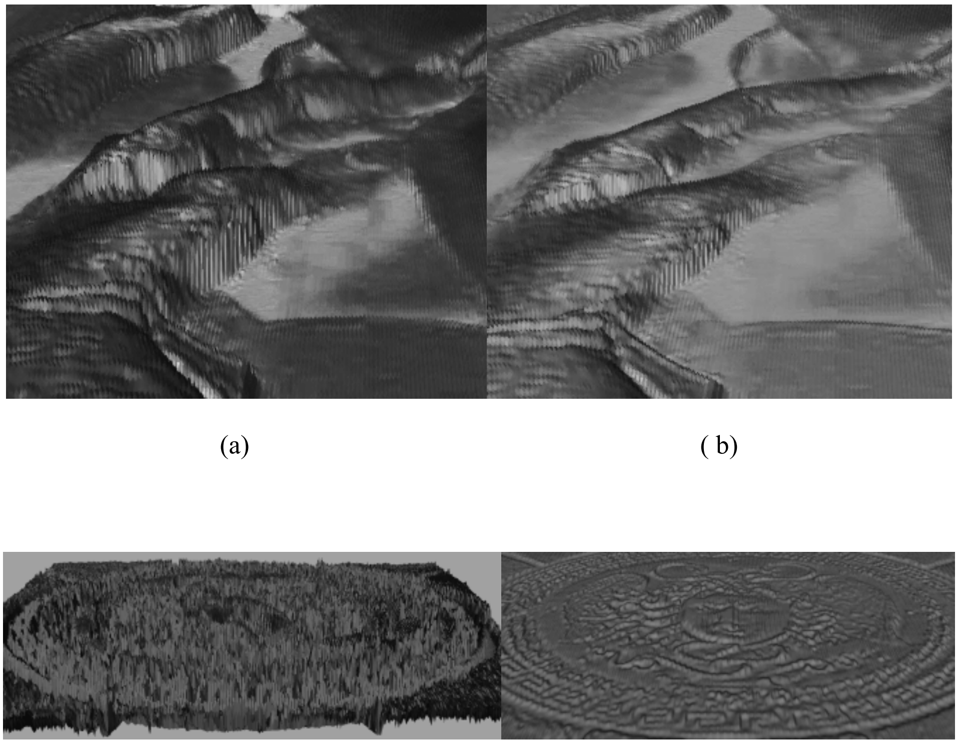


Figure 55: Surface details modeled using the SFS algorithm and transferred onto the base mesh.

For SFS algorithm, k was set to 10 as the maximum iterative time criteria, and μ was set as 0.02, as the higher values tend to distort the overall depth of the image, as shown in the Figure 56. The value of μ as 0.02 produced the most desirable results as higher values produced unwanted distortions while lower values compromised the desired level of detailed by making the 3D relief too flat.



(a) (b)

(c) (d)

Figure 56: Two examples of relief meshes degenerated using two separate values of μ a,c) μ = 0.5 b,d) μ = 0.02.

The spine curve of the generalized cylinder is composed of multiple control points, which is approximated by a B-spline curve. The spline curve was initialized to contain a minimum of three control points. The user has the option to specify the number of subdivisions of the spine curve to set the desired smoothness as shown in Figure 52. A subdivision algorithm was used by DeRose et al. (1998) for smoothing the spine curve according to the number of subdivision steps specified by the user. In this way a relatively rough curve can be approximated with a smooth B-spline curve. This is demonstrated in Figure 57. The cylinder drawn on the left has been subdivided to 2 levels, while the cylinder on the right has been subdivided to 5 levels.

Subdivisions = 2 Subdivisions = 5

Figure 57: Subdivision steps.

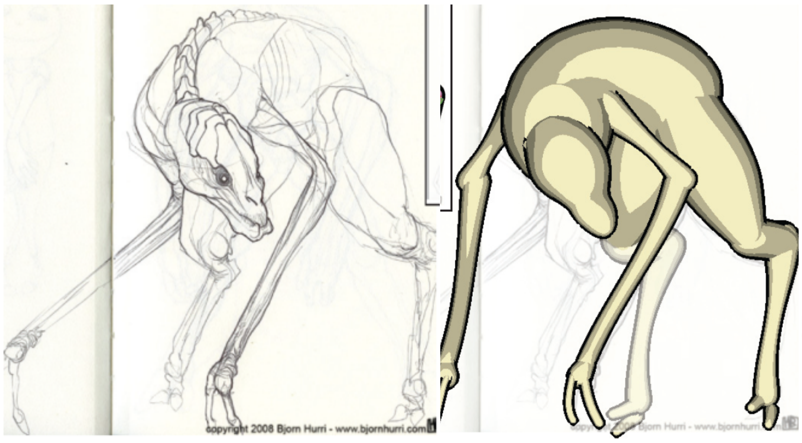
For transferring details from the source mesh onto the target base mesh, the GeoBrush demo program provided by the authors at (http://igl.ethz.ch/projects/geobrush/) has been utilized. Here the artist can very easily perform the operation of selecting the source mesh and the target mesh and the system performs the appropriate details transfer.

4.5 Results and Discussion

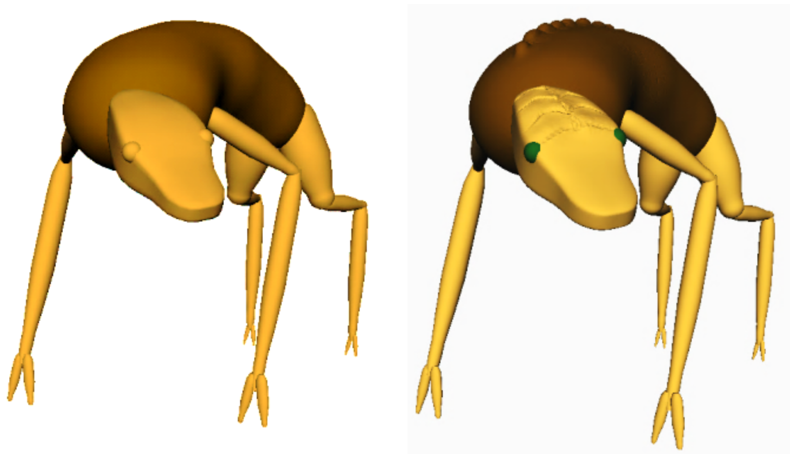
The proposed system has been tested with research students. The students found it easy and intuitive to quickly model the base mesh from a single picture. The users also found it very intuitive to generate a 3D surface from a photo and add it to the base mesh in order to show details on a mesh.

The proposed system can be used for rapid prototyping of character models and can find applications in game development studios, animation studios, online gaming etc. The proposed system has been targeted towards novice animators and artists.

The proposed approach has been designed to create prototypical base mesh models in less time as compared to the ones created via Autodesk Maya (Maya). With normal Maya modelling, it is very time consuming to modify the primitive shapes to resemble the different parts of the input sketch. In contrast, the proposed approach enables an artist to quickly modify the primitive shapes using sketching gestures and vertex manipulation tool. It indicates that creating different parts of a human body from primitives is much easier and faster than modelling with Autodesk Maya (Maya) etc. The proposed system is targeted towards unskilled 3D artists with little experience of professional tools like Autodesk Maya (Maya).



(a) (b)



(c) (d)

Figure 58: (a) Input sketch of the monster model (b) Monster model created using (Gingold et al., 2009), (c) Base mesh of monster model created using the proposed approach, and (d) Details added to the base mesh.

Figure 58 shows a comparison of a monster model created using the approach of Gingold et al. (2009) and the proposed approach. The proposed approach took nearly 15 minutes to create the base mesh model in Figure 58(c), which is almost the same time taken by the Gingold et al. (2009), however as compared to the models created by Gingold et al. (2009) in Figure 58(b), Figure 58(c, d) shows that the proposed approach can create more detailed prototypical models.

In Figure 58(d), the details were incorporated using the detail transfer technique explained in Section 4.3.6. For this task, a sample of the image from Figure 58(a) was extracted to be used to generate the detail for the head of the monster. Similarly a sample of the details at the back of the monster were extracted from Figure 58(a), which was then treated with SFS method explained in Section 4.3.6.

4.6 Conclusion

In this chapter, a hybrid sketch-based modelling system is presented, which combines the features from existing systems such as (Gingold et al., 2009) and shape-from-shading (Yang and Han, 2007) to allow artists to quickly model the base mesh of the character as well as add details to the base mesh. The system provides intuitive tools to add details to the base mesh. In contrast to the popular commercial animation packages such as Autodesk Maya (Maya), and Autodesk 3Ds Max (3Ds Max), which have a steep learning curve, the system allows the artists to quickly create a base mesh from a single view via simple strokes, and utilizing the power of primitives (generalized cylinders, cubes, and ellipsoids). Moreover, the proposed system allows the artists to quickly add details to the surface of the base mesh by generating a 3D surface from a photo with minimum effort. It is believed that the proposed system makes a fresh attempt at pushing the sketch-based modelling systems beyond their boundaries.

4.7 Limitations

There are several limitations of the proposed system including:

1. The mesh manipulation controls are not easier to learn and can be improved in future by simplifying the controls.
2. In contrast with Gingold et al. (2009), the system does not make use of Laplacian mesh editing (Sorkine et al., 2004) to modify meshes in a more artist friendly manner. This is an important feature that is now being embedded in popular modelling packages such as Blender (Blender), and this feature is planned to be added to the system in future.

In Chapter 5, possible future research directions are discussed to address the limitations mentioned above.

5 Sketch-to-Box: A New Character Modelling Technique

5.1 Introduction

Anthropomorphic 3D character models are indispensable assets in the video game and VFX industry. As the technology is advancing, so is the need to render highly realistic human characters in movies and games. This realism is crucial for entertaining the audience and making them believe in things that are virtual. Some examples of highly realistic 3D human characters in movies are 1) cyclops in 'Wrath of the Titans' 2) Navi in Avatar 3) The Hulk in 'Avengers Assemble'. One of the most important attributes these characters share is the anatomical correctness in their appearances. The muscles and body proportions in these characters are very close to the actual human body, which are carefully modelled and sculpted.

Unfortunately, modelling of anatomically correct anthropomorphic characters is a painstaking skill to learn, and to master it is considered as one of the holy grails of character modelling. Character artists who learn to draw a human figure from life spend a considerable amount of time understanding the curvatures and muscular flow of a human body. This is because the curvatures of the human figure change with different poses. A human body is made up of hundreds of muscles. Most of the muscles are visible through the curvatures on the surface of the human body. When a human body is changed to a new pose, a certain amount of muscles change their size and shape. This is because human body muscles are stretchable and contractible. These muscles transformation are visible through the skin when the human body changes its pose. In artistic terminology, this is known as Surface Anatomy (Sheppard, 1991).

It is considered to be very difficult for an artist to master the surface anatomy due to its shape varying nature. Professional animation packages such as Autodesk Maya (Maya) and Pixologic ZBrush (ZBrush) provide tools to create surface anatomy in 3D. However they are difficult to master for beginners and thus result in the withdrawal of many beginners. Thus there is a serious scarcity of character modelling tools which allow novice artists to improve their character modelling skills and smoothly graduate towards professional tools. For character modelling, professional tools Autodesk Maya (Maya) revolve around the idea of careful planning of mesh topology and good knowledge of NURBS modelling which are difficult to master. Moreover, in an animation production pipeline 3D character assets go through multiple levels of anatomical refinement in order to look as close as possible to the real humans.

Generally speaking, box modelling is a traditional character modelling approach used by 3D artists to model an entire character model from a single box, which serves as a starting point. An example is shown in Figure 59. This approach has been widely adopted by 3D artists. However to model realistic human characters, box modelling requires the artists to have good knowledge of anatomy. More than just having good knowledge of anatomy, the artists must have a substantial amount of experience of making the mesh topology that should follow the correct muscular curvatures, and bring out the maximum details in the human figure being modelled. Using this knowledge and substantial amount of practice an artist creates a topology that adheres to the muscular structure of the human body. However, if the topology is incorrect, the artist has to redo the topology using topology refinement tools / techniques. This requires the artist to be extra careful during the modelling process, incurring long delays in the overall modelling process. Thus the learning curve of traditional box modelling is highly steep. Another big limitation is that it is not suitable for modelling characters in different poses or extreme poses, since it is mostly used to model half of the model in a T-pose or A-pose and duplicate the other half.

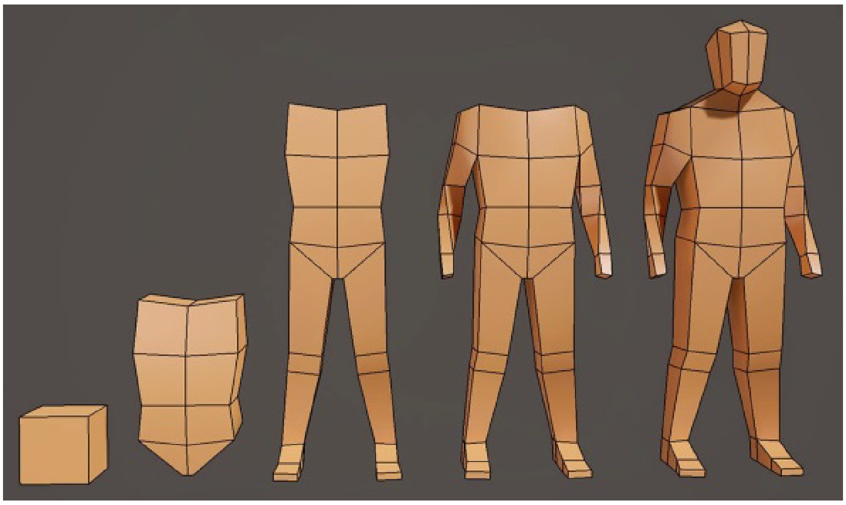


Figure 59: The process of traditional character modelling from a single box. *Image source: unicointeraction.blogspot.co.uk*.

Another approach for character modelling is organic modelling using tools such as Autodesk Mudbox (Mudbox). This approach mimics the process of real life clay sculpting in 3D. Although these tools provide a free hand sculpting input, one of the major drawbacks of these tools is they require immense computing power to process millions of polygons in order to mimic clay sculpting in 3D. In these tools, an artist needs to first make the primitive models smooth enough to easily sculpt the muscular details. Another drawback is that these tools also require artists to plan well ahead of starting to model a human character in a particular pose because of the flow and arrangement of muscles resulting from the pose. Moreover these tools are designed mainly for professional artists having knowledge of clay sculpting, and they are known to be unfriendly for beginners.

Many professional artists in the past such as the 20th century artist George Bridgman, have claimed that the complexity of the human body can be most easily broken down into box shaped forms. Modelling using smooth surfaces such as cylinders and spheroids requires more pre-planning and attention to detail, and more computational power due to the smoothness of these surfaces. In his seminal book for artists *Constructive Anatomy*, Bridgman (1920) has taught with great depth about how each muscle can be represented using a box/block that gives it a feeling of mass, and how an artist can construct the entire human figure and surface anatomy using simple geometric shapes mainly a box. An excerpt of his book is shown in Figure 60. His techniques have been adopted and used by character artists for decades and most modern comic artists use his techniques to draw comic characters. He also talks about how these boxes can be oriented and arranged in a particular manner in order to give a sense of depth to the human figure and a sense of mass. According to him, human muscles can be broken down into a number of WEDGE like shapes, which provide intuitive approximations of connections of the actual muscles. Modern artists such as Zarins and Kondrats have also adopted Bridgman’s approach by teaching art students how to master drawing human figures (Zarins and Kondrats, 2014). In the proposed approach an attempt has been made to transform the classical ideas of Bridgman from 2D to 3D and aimed at mimicking his techniques in a sketch-based modelling environment in order to provide a novel human body modelling system.

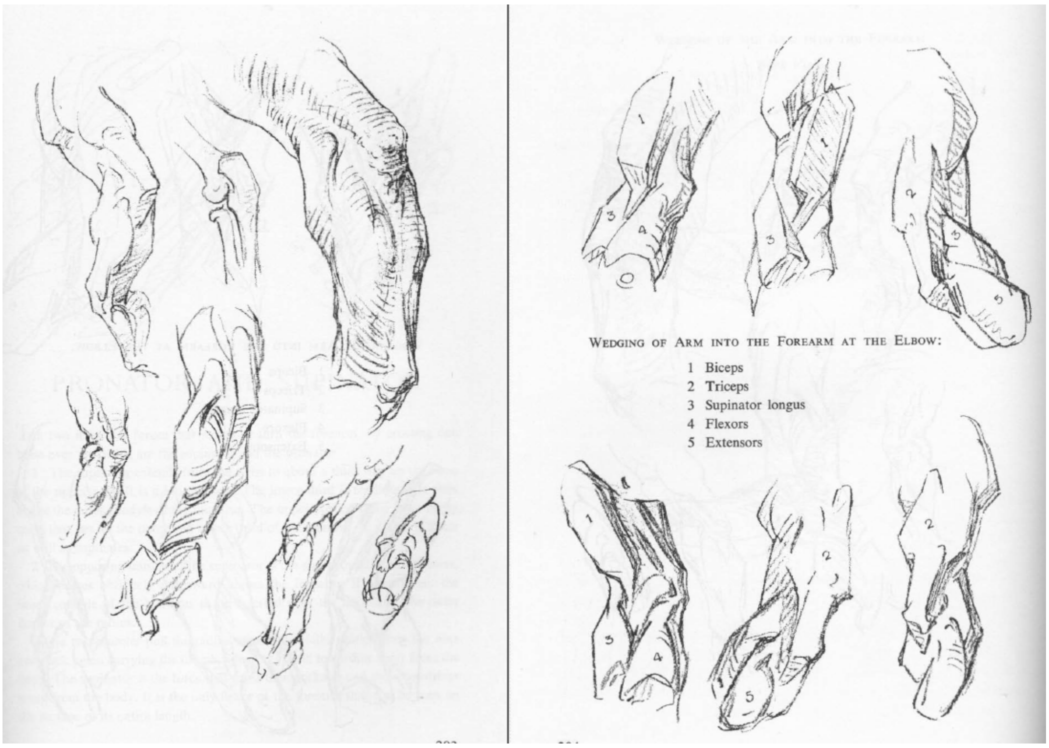


Figure 60: An excerpt from George Bridgman's book 'Constructive Anatomy' demonstrating the anatomical construction of arm muscles.

This chapter proposes a novel approach for modelling of plausible human characters using 2D/3D boxes with a focus on developing a good knowledge of modelling surface anatomy. Resting its foundations on the pioneering techniques of human figure drawing by George Bridgman, the idea presented here aims to transform the classical technique into a modern sketch-based 3D modelling system, thus proposing a unique and novel approach to character modelling. The proposed approach is also data driven in nature as it provides cues to the novice artists to model muscles. The proposed system provides an intuitive sketching interface to the life drawing artists and students to create 3D character models with greater ease and correct anatomical details. Moreover the system provides tools to the artists to create and modify muscles of the human body by easily sculpting 3D boxes and refining them to match the shape of the human muscles, while conforming to the drawing principles set out by George Bridgman.

The following contributions has been made in the proposed system:

1. Based on the knowledge gained in this research, the proposed system is the first of its kind that provides a unique sketch-based modelling experience for novice as well as professional artists while improving their skillset for creating impressive human character models that follow *surface anatomy* well.

2. The proposed approach looks at human figure modelling from the perspective of modelling from boxes to depict body parts/muscles following the highly intuitive drawing technique of George Bridgman. There is no evidence of past techniques that construct the anatomically correct human figure from boxes using George Bridgman's approach in a sketch-based modelling context.

3. The proposed approach involves efficient data-driven and cage-based morphing techniques that use OBB tree representations for 3D muscles, to provide effective cues to the novice artists to place and model anatomically correct human muscles.

4. Finally, a novel algorithm has been designed for inferring 2D boxes into 3D boxes. Using this algorithm, the system has made it possible to infer correct orientation, scale and position of boxes in 3D and generates the boxy character almost instantaneously (Figure 61). Hence the system provides a useful tool for character artists to visualize their pose in 3D.

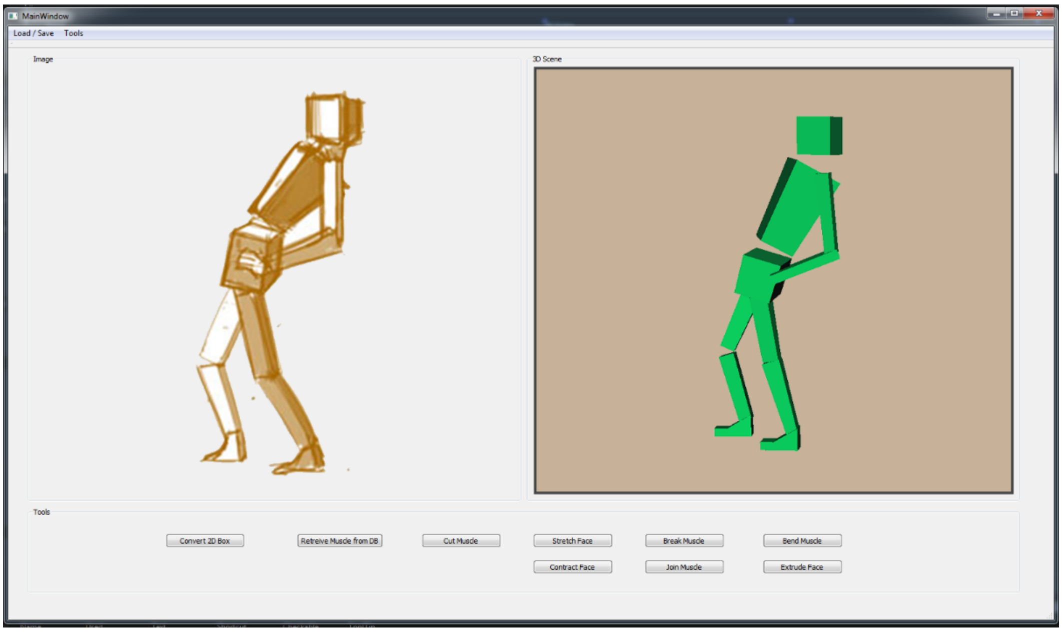


Figure 61: Prototypical interface showing the boxy character generated instantaneously.

One of the purposes of the proposed system is to be used in the initial design phase in animation studios. In animation studios, prior to the modelling phase, a very important phase is to block out the scene with simple 3D objects to depict the final models to be shown in the final scene. Often the blocking phase there is a requirement for characters to be depicted in different poses giving the director a sense of what the final scene will look like. In this scenario, the system can act as a great tool to allow the artists to very quickly create articulated posed characters with plausible surface anatomy. Please note that the proposed system has not been targeted to be used in the actual modelling phase of the animation pipeline, as the user solely creates muscles separately using boxes which is not suitable for animating characters.

The system can also be used as a training tool for beginner artists. Since the proposed approach is founded on the drawing paradigm proposed by George Bridgman, and there was a scarcity of tools that can allow the artist to mimic this artistic in 3D, the system can act as a bridge to fill this gap.

Another important application of the system presented here is in 3D printing. The proposed system can be used to create stylized characters very quickly with low polygon count and thus can be used to decorate interiors.

Finally the proposed system can be used to create 3D prints of planar character models for art students and professionals. One such example is of the Male Planar Statue being sold by the artist David Richardson which shows muscles in boxy fashion (Figure 62).

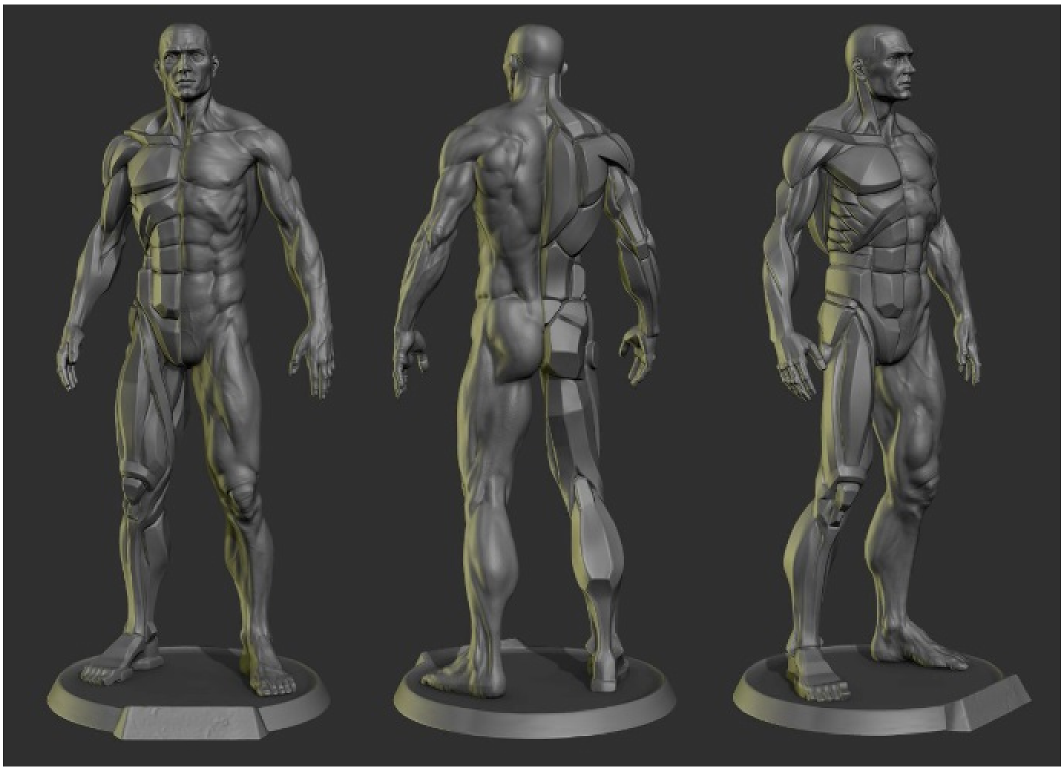


Figure 62: Male planar model by © David Richardson.

5.2 Related Work

Traditional figure drawing has been pursued from two directions. First is a classical approach which is referred to as *Rhythm Life Drawing* (Mattesi, 2012). This approach emphasizes on representing the bodily curvatures and muscles with smooth curved strokes as shown in Figure 63.

Figure 63: A human character sketch highlighting smooth curved stroked typically used in 'rhythm life drawing' practices. I*mage source: (Mattesi, 2012).*

The second and a more refined approach is to draw the human body as a set of boxes depicting all the major parts of the body (Figure 64) such sa head, torso, pelvis, arms, legs and feet. In the second approach, the muscles of the human body are iteratively refined in boxy fashion in order to approximate the curvature of these muscles. This approach was pioneered by George Bridgman. Later on, several other artists adopted, refined and taught this technique of figure drawing and claimed that the said technique is the easiest to learn and master (Bridgman, 1920). As argued in Bridgman's work, box based refinement of muscular body parts is an effective approach towards improving one's skills in character modelling.

Figure 64: Characters made out of boxes depicting major body parts. *Image courtesy of conceptart.org user 'BlindLynx'.*

The proposed system is a hybrid SBM (Sketch-based Modelling) system for human character modelling, which includes retrieval of 3D muscles from the database, and direct construction and modification of new muscles. From box modelling point of view, the proposed system is closer in spirit to Zeleznik et al. (2007), which is a construction based SBM system and provides a novel interface for 3D drawing. It is one of the earliest systems to leverage traditional pen and paper sketching of engineering drawings. This system was designed to allow CAD artists to translate their ideas in a more natural manner. Inside the system is employed a novel inference algorithm, which is gesture driven for generating 3D boxes from 2D sketched boxes. The system also allows the artist to modify the boxes via extruding. However the controls provided for modification are too limited to be used for comprehensive modelling of human characters, and are mostly suited for engineering drawings instead of organic modelling.

Most recently another construction based system called Second Skin by Paoli and Singh (2015) have proposed an intuitive SBM tool to model the costumes and suits on top of naked 3D character models. The authors have addressed the long standing problem of quickly modelling layered geometrical structures with volume, which is largely absent in virtual clothing systems. Their main contribution is a novel interactive 3D curve inference algorithm which is based on curve classification. However there are a few limitations in their system. Firstly, this system tends to restrict the artist to draw the layered armours and suits on top of existing 3D characters. Secondly, the system has no support for modelling the surface anatomy of the human body. Finally this system is a tool for very quickly modelling layered suits but trades away the accuracy of the armour shapes.

In (Ku et al., 2006), the authors have proposed an intuitive interface to generating 3D objects from 2D polyhedrons of irregular symmetry. Their system produces a vertex-edge graph to interpret the input line drawings, and also provides the functionality to amend the final drawing by adding more polyhedral 3D objects to the original model. This technique is highly automatic which makes it less suitable for figure drawing artists who tend to stay engaged with the system.

Takayama et al. (2013) have presented an attractive sketch-based system which proposes some useful tools for directly editing of quad meshes via sketching operations. The good interactivity of the system make it suitable for handling manual quad re-meshing that provides the user with a high degree of control while avoiding the tediousness involved in existing manual tools, thus refining the topology of the final mesh. In a similar spirit to their system, the system provides a high degree of control for the user to sculpt human anatomical models. Contrary to their approach, the proposed system provides extensive operations / tools to cover the sculpting needs of 3D artists, however in a more natural manner as opposed to using Autodesk Mudbox (Mudbox).

The proposed system being a highly interactive system includes operations for modifying the box-shaped muscles by manually modifying the faces of the muscles. Earlier systems which offered tools for face modifications (extrusion and intrusion of faces) had several limitations and were prone to inconsistent topology and self-intersections. Moreover these systems were vulnerable towards unwanted holes in the geometries. To solve these problems, recently Lipp et al. (2014) have presented a novel tool to freely add or subtract faces to an existing model just by pulling or pushing a face while avoiding face self-intersections. The proposed system takes several practical cues from this system (Lipp et al, 2014) as the operations designed for the proposed system happen to involve extrusion of faces.

Bae et al. (2008) have developed a robust and feature rich system for 3D sketching. The basic “feel” of the system borrows from that of a physical paper sketchbook. This system provides designers with a virtual sketchbook with tools for smooth navigation of multiple canvases with interactions such as: tearing, peeling, panning, zooming, and rotation. The system also supports automatic dynamic rotation of the virtual sketchbook based on the users’ input strokes to make further multi-stroke sketching biomechanically comfortable. However according to the case study mentioned in the paper, one artist did not find the automatic dynamic rotation of view convenient and rather distracting. One limitation of this system is that it is not suitable for modelling organic or human character models as most of the models demonstrated using this system are mechanical in nature such as aircrafts, cars, and spaceships etc.

5.3 System Overview

To fulfil the aim of providing a sketch-based modelling tool to create convincing 3D human models from scratch in the design stage, the proposed system has been divided entirely into two modelling components:

1. Box Modeller
2. Muscle Sculptor

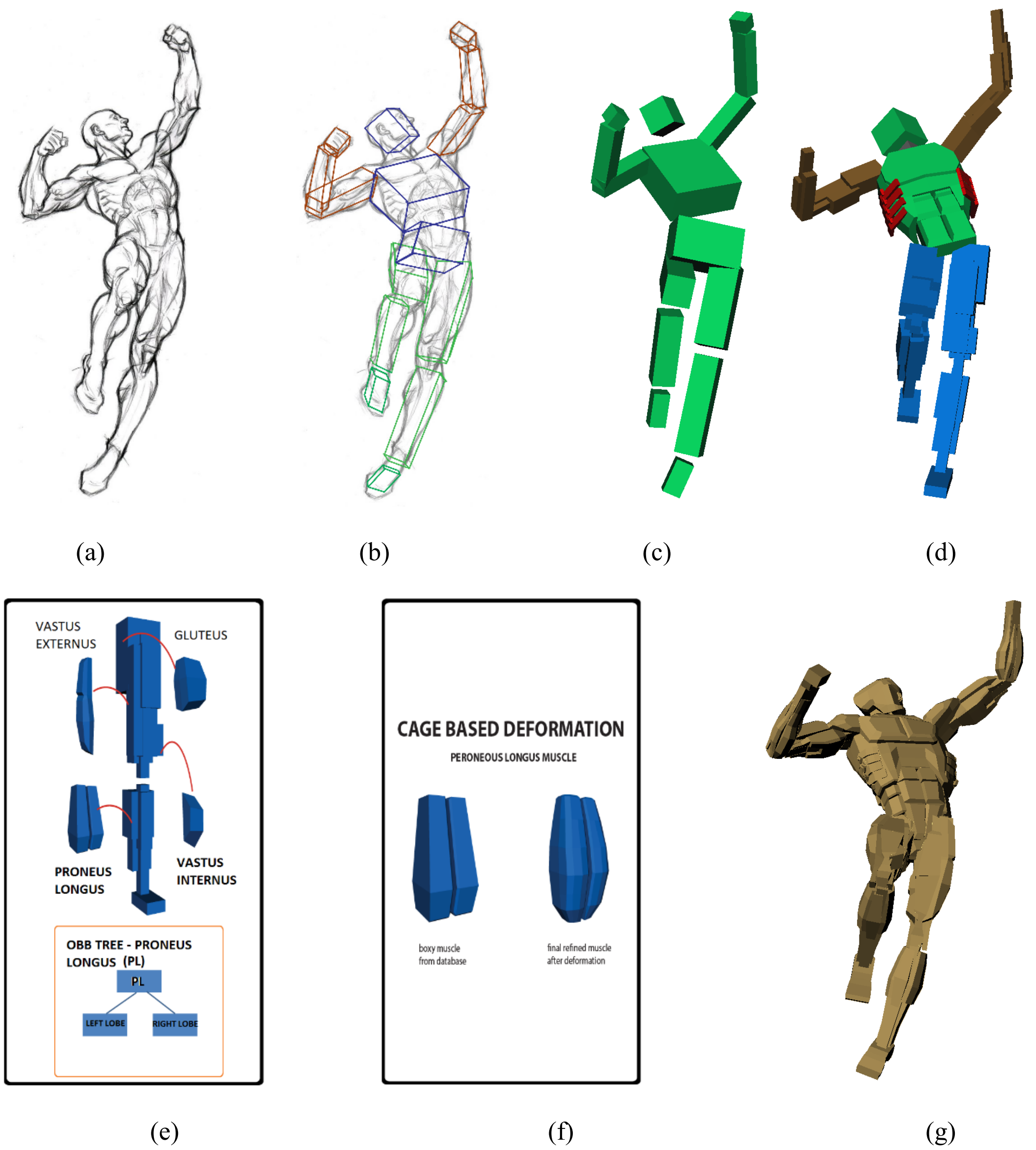


Figure 65: Pipeline of the proposed approach. (a) Source image loaded into the system by the artist, (b) boxes drawn by the artist on the source image to represent the major parts, (c) Boxy character: 3D boxes inferred by the system, (d) Boxes added by the artist to the boxy character to represent muscles, (e) Boxy muscles retrieved by the system from the database corresponding to the boxes added in the previous step using OBB tree representation, (f) Cage-based deformation of boxy muscles to further refine the muscles' shapes, and (g) Final human character achieved after several iterations of proposed sculpting operations.

Box Modeller allows the artist to quickly create a 3D human figure made up of 3D boxes by sketching 2D boxes in the 'sketch pane'. Thus this component is used to depict the desired pose of the human figure in a very quick and easy way using boxes, which can be viewed in a 3D space. Muscle Sculptor extends the boxy model created from the first component and provides tools to iteratively refine the boxes to resemble human muscles and also add more boxes to the structure for further refinement.

The need for these two tools is backed by the observation of the process character artists go through in order to create anthropomorphic characters. The very first step done by the artists is to quickly create an articulate pose entirely from boxes. The artists then proceeds with adding details to the boxes. He/she refines the topology to resemble the shape of muscles in a human body. This step is done interactively until the point the artist achieves the desired character model. The proposed system has been divided into two components as these mimic the entire artistic process, by delivering a SBM tool to the artists to mimic this traditional artistic process.

The proposed approach is sequential in nature, which is demonstrated in Figure 65. The first step is to load an image of a human character in the proposed system, which the artist uses to draw the boxes on top of the character. The system infers the boxes in real-time and presents a character made entirely of boxes. The artist then proceeds with drawing more boxes to roughly represent muscles. The system then retrieves the boxy muscles and refines them further by deforming them using cage-based deformation. Alternatively the artists can use the operations provided by the system to directly sculpt the boxes (representing muscles) iteratively to achieve realism in the surface anatomy of the human character.

5.3.1 Box Modeller

Approximating a human character sketch using boxes is a technique widely used and appreciated by professional artists. This technique, which is commonly known as ‘human gesture / pose drawing’ has proved to be highly pedagogic in order to understand not only the pose but also the camera perspective and proportions of the human body. In the past, famous renaissance artists such as Luca Cambiasi have used boxes/cubes to quickly conceptualize the human poses and postures. From the works of Luca Cambiasi it has been revealed that these artists were accustomed to creating prototypical drawings of human figures entirely of boxes, for the purpose of creating gesture drawings as shown in Figure 66. These initial drawings helped artists to understand the volumes and proportions of the different human body parts and subsequently add details with greater ease. Early 19th century artist George Bridgman has used this technique considerably to teach figure drawing in his professional career. Modern artists have also advocated in favour of this technique in order to accelerate the learning process of sketching human figures in a life drawing and fine arts education (deMartin, 2016). The literature review conducted for this research project has revealed that ever since sketch-based modelling has been pursued as a research area, there has been a surprising scarcity of tools for artists that can help them instantaneously visualise the ‘human gesture / pose drawings’ in 3D.

Figure 66: A sketch by renaissance artist Luca Cambiasi.

For a novice artist, the system provides support to load an image and allow the artist to draw boxes on the image to represent the major body parts (torso, arms, legs, head, and pelvis). This is shown in Figure 67.

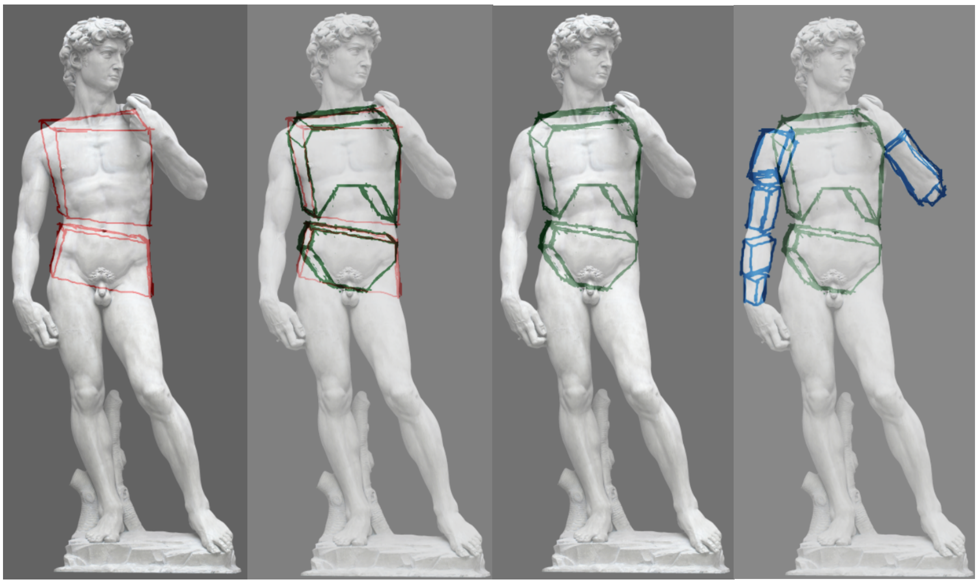


Figure 67: Artist's box drawings on an image using the system's interface.

For the professional artist, the box modeller component of the system allows an artist to quickly create a pose of the human model using 3D boxes, based on an input sketch. The system requires the artist to sketch 2D boxes in the 2D panel and then automatically infers the input boxes using the system's novel algorithm (box inference algorithm) which infers the orientation, scaling and translation of the input sketched boxes and very quickly creates a boxy model using 3D boxes, which matches the pose of the input sketch.

The interface of the propoesd system is divided into two panels laid out adjacent to each other. 1) The left 2D panel, where the artist can either load an image of a human figure and draw 2D boxes on top of that image, or directly draw 2D box sketches without using an image. 2) The right 3D panel where the system generates the 3D boxes after inferring the 2D sketched boxes in the 2D panel.

Figure 68 shows a model created using the prototypical interface of the ‘Box Modeller’ component.

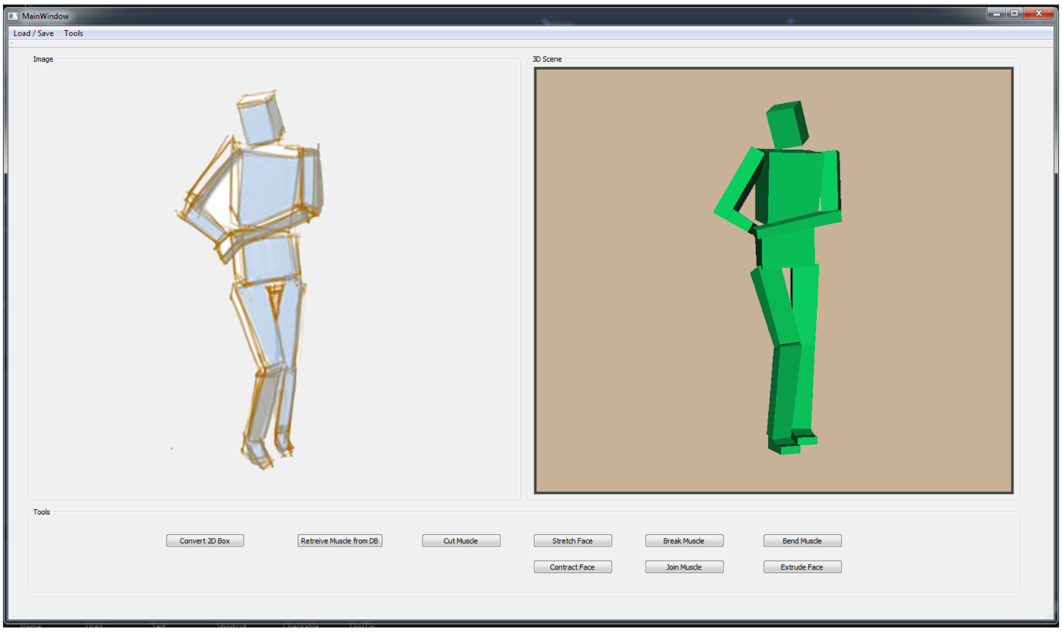


Figure 68: Boxy character model created using the prototypical interface.

In Sections 5.3.1.1 to 5.3.1.9, the thesis discusses about how the inference of a 3D box from its 2D counterpart works, and also discusses about how the system automatically aligns the two 3D boxes to form a boxy character such as the one shown in Figure 68.

5.3.1.1 Box Inference Algorithm: In three-dimensional Euclidean space (Figure 69), suppose we rotate an axis-aligned cuboid ABCDEFGH to A’B’C’D’E’F’G’H’ and project it onto a plane to obtain a projected cuboid A”B”C”D”E”F”G”H”. A-H define the vertices of the cuboid. If we have already known the 2D coordinates of the eight points of the projected cuboid, can we deduce the 3D coordinates of the eight vertices of the original cuboid?

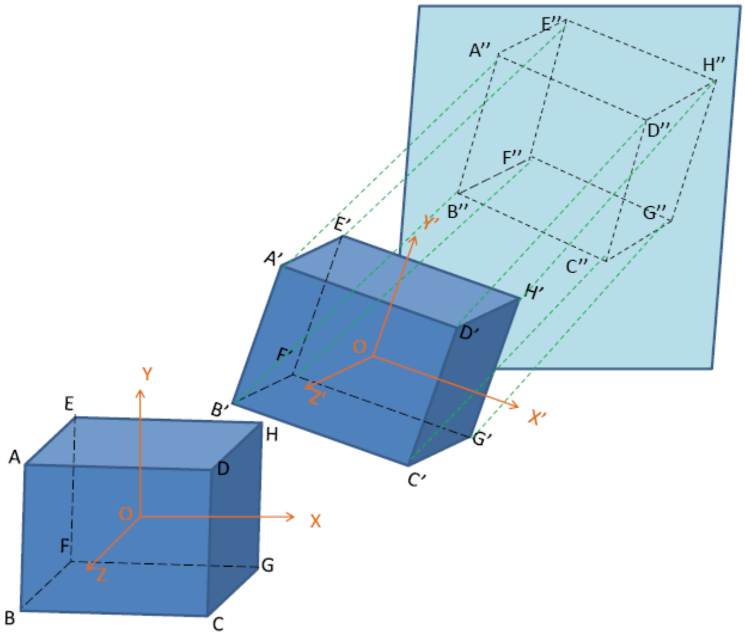


Figure 69: Inference of 3D coordinates of the eight vertices from 2D sketch input.

In order to answer the above question we need to find the length of the edges of cuboid, as well as the rotation axis and rotation angle needed to rotate the cube in the 3D scene. These algorithms are detailed below.

5.3.1.2 Solution to the length of the edges of the cuboid: First, we build a three-dimension Cartesian coordinate system OXY Z in which the center of the cuboid ABCDEFGH is the origin O, and use $\overrightarrow{FG}$ as the positive direction of the X axis, $\overrightarrow{FE}$ as the positive direction of the Y axis, $\overrightarrow{FB}$ as the positive direction of the Z axis. We know $\overrightarrow{FG} \perp \overrightarrow{FE}$, $\overrightarrow{FG} \perp \overrightarrow{FB}$, $\overrightarrow{FE} \perp \overrightarrow{FB}$. Because rotation does not change the perpendicular relation, thus we get $\overrightarrow{F'G'} \perp \overrightarrow{F'E'}$, $\overrightarrow{F'G'} \perp \overrightarrow{F'B'}$, $\overrightarrow{F'E'} \perp \overrightarrow{F'B'}$. In order to describe briefly we write those vectors in a matrix as below:

$$\begin{bmatrix} G'-F' & E'-F' & B'-F' \end{bmatrix} = \begin{bmatrix} a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \\ b_1 & b_2 & b_3 \end{bmatrix} \tag{5.1}$$

where $a_1, a_2, a_3, a_4, a_5, a_6$ are six known real numbers that represent the $x, y,$ and z components of the vectors $G' - F', E' - F', B' - F',$ while b_1, b_2, b_3 are three unknown parameters that need calculating. Utilizing the perpendicular relation mentioned above yields:

$$a_1a_3 + a_2a_4 + b_1b_2 = 0$$

$$a_1a_5 + a_2a_6 + b_1b_3 = 0$$

$$a_3a_5 + a_4a_6 + b_2b_3 = 0$$

(5.2)

From the above Equation, b_1, b_2, b_3 can be solved. So from Equation (5.1) we know the length, height and width of the cuboid as well.

5.3.1.3 Solution to three axes of rotated cuboid: After evaluating b_1, b_2, b_3, the orientation of the three axes of rotated cuboid may be expressed like:

$$\begin{bmatrix} OX' & OY' & OZ' \end{bmatrix} = \begin{bmatrix} a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \\ b_1 & b_2 & b_3 \end{bmatrix} \tag{5.1}$$

5.3.1.4 Solution to rotation matrix: According to the relationship of coordinates before and after rotation, we can get:

$$R_A(\theta)\underset{p}{\left[G-F \quad E-F \quad B-F\right]}=\underset{P'}{[G'-F' \quad E'-F' \quad B'-F']} \tag{5.2}$$

Following the solution of Section 5.3.1.2, we know all the entries of the left side matrix P and the right side matrix P' in Equation (5.4), thus the rotation matrix can be written as below:

$$R_A(\theta)=P'P^{-1} \tag{5.3}$$

5.3.1.5 Solution to rotation axis and rotation angle: Rotation matrix $R_A(\theta)\in R^{3\times 3}$ in equation (5.4) rotates a vector through an angle θ about the unit axis A:

$$R_A(\theta)=\begin{bmatrix} c+(1-c)A_x^2 & (1-c)A_xA_y-sA_z & (1-c)A_xA_z+sA_y \\ (1-c)A_xA_y+sA_z & c+(a-c)A_y^2 & (1-c)A_yA_z-sA_x \\ (1-c)A_xA_z-sA_y & (1-c)A_yA_z+sA_x & c+(1-c)A_z^2 \end{bmatrix} \tag{5.6}$$

where A_x, A_y, A_z are the x-, y-, z- coordinates of the Axis A and c, s stand for the cosine and sine of θ respectively.

From Section 5.3.1.4 we know all the entries of $R_A(\theta)$. Correspondingly we can obtain an equation set encompassing nine equations. By using Levenberg-Marquardt method, A_x, A_y, A_z, c and s can be evaluated after several iterations.

5.3.1.6 Solvability: The equation set can be solved only when the right side matrix of Equation (5.1) is full rank. In other words, if there is no plane of the cuboid become a single line after projection, then the equation set is solvable.

As the original cuboid is axis aligned and the length of each edge is not zero, the matrix P on the left side of Equation (5.4) is invertible. Furthermore, following Section 5.3.1.2 can be solved we can always solve Section 5.3.1.4.

5.3.1.7 Box Alignment: Without automatic alignment of boxes, the boxes created to represent legs and hands would all be created on a single plane regardless of the pose of the concept human model, which will create an incorrect final pose. The proposed system provides an interactive interface where the artist can create a spline curve in an orthographic view and then the box automatically aligns itself according to the created spline curve.

The line of action (LOA) is widely used in 2D animation design. Inspired by Guay et al. (2013), a sketch-based editing method has been developed that matches the shape of the 3D character boxy model to the shape of the 2D stroke.

5.3.1.8 Matching boxy character to LOA: Since the character model is made up of boxes, the principle axis of each box would serve as a bone for the given body part. The connection relationship would be generated from the adjacent boxes, where each connection would serve as a joint, which has rotation freedoms that allow the model to create the pose. Therefore, the box model would finally be posed by the connections. For a selected series of m boxes $x_b(i), i = 1,2,\ldots,m$, assume $x_c(s_i)$ represents the $m+1$ corresponding connections/terminals of piecewise-rigid m boxes, and x_r is the single root position, which is the center of the box standing for waist and P_v is the current view matrix. A given 2D stroke is denoted as $x_{loa}(w_i), i = 1,2,\ldots,m+1$. The input stroke is firstly parameterized as a cubic Hermite Curve, where (w_i, w_{i+1}) is the segment that correspond to the box (s_i, s_{i+1}). To minimize the difference between the 2D stroke and the 3D box chains in the current view plane determined by the view matrix P_v, the warping function is defined as:

$$\min_{Q,x_r} \sum_{i=1}^{m+1} E_i^P + E_i^T$$

$$E_i^P = \lambda^P \|P_v x_c(s_i) - x_{loa}(w_i)\|^2$$

$$E_i^T = \lambda^T \left\|T_c(s_i) - \check{T}_{loa}(w_i)\right\|^2$$

(5.4)

where $\{\lambda^P, E^P\}$ is to chose the closest point on the line while the term $\{\lambda^T, E^T\}$ is used to emphasize rigidity of the boxes. The T_c and $\check{T}_{loa}$ denote the tangent of the principle axis of the 3D box and the 2D stroke at corresponding point i, respectively. The Q is the rotation angle of each connection. In order to solve the depth ambiguities problem, additional constraint is applied: we constrain the models transformations to lie in the viewing plane, i.e. along the 2D dimension of the viewing plane. In other words, the rotation of each box is parameterized to a single axis-angle component θ_i, the user can separately sketch and rotate the camera to edit in depth. Hence the rotation $Q = \{q_1, q_2, \ldots, q_3 + 1\}$ will be solved with respect to θ_i.

5.3.1.9 Correspondence between LOA and Bodyparts: For a given input 2D stroke, in order to use it to manipulate the character, a key problem is to identify the correspondence of the line segments and each box. Since the warping function is piecewise-defined, one segment of the stroke is corresponding to one box, we denote the segments as $\Omega = \{(w_1^0, w_1^1), (w_2^0, w_2^1), \ldots, (w_m^0, w_m^1)\}$ for the m boxes. The sample of the curve at a node i would be calculated as $w_i = 0.5\left(w_i^0 + w_{i-1}^1\right)$. The segment identification problem is then translated as a minimization problem:

$$\min_{W} \sum_{i=1}^{m+1} E_i^P + E_i^T + E_i^C$$

$$E_i^P = \lambda^P \| P_v x_c(s_i) - x_{loa}(w_i) \|^2$$

$$E_i^T = \lambda^T \left\| T_c(s_i) - \check{T}_{loa}(w_i) \right\|^2$$

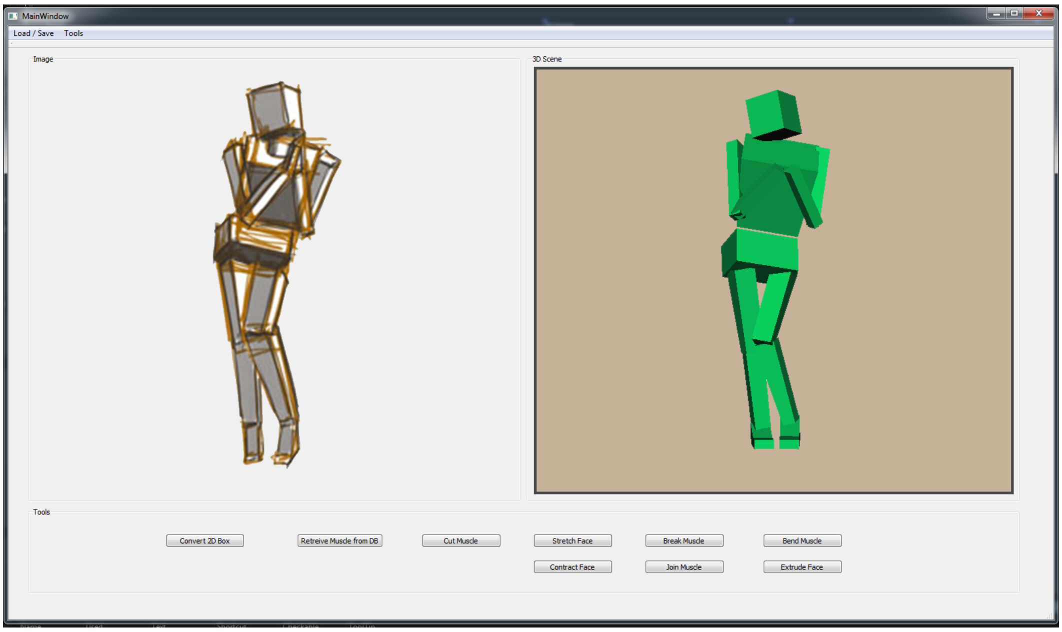
$$E_i^C = \lambda^T \| w(i+1)^0 - w(i)^1 \|^2$$

(5.5)

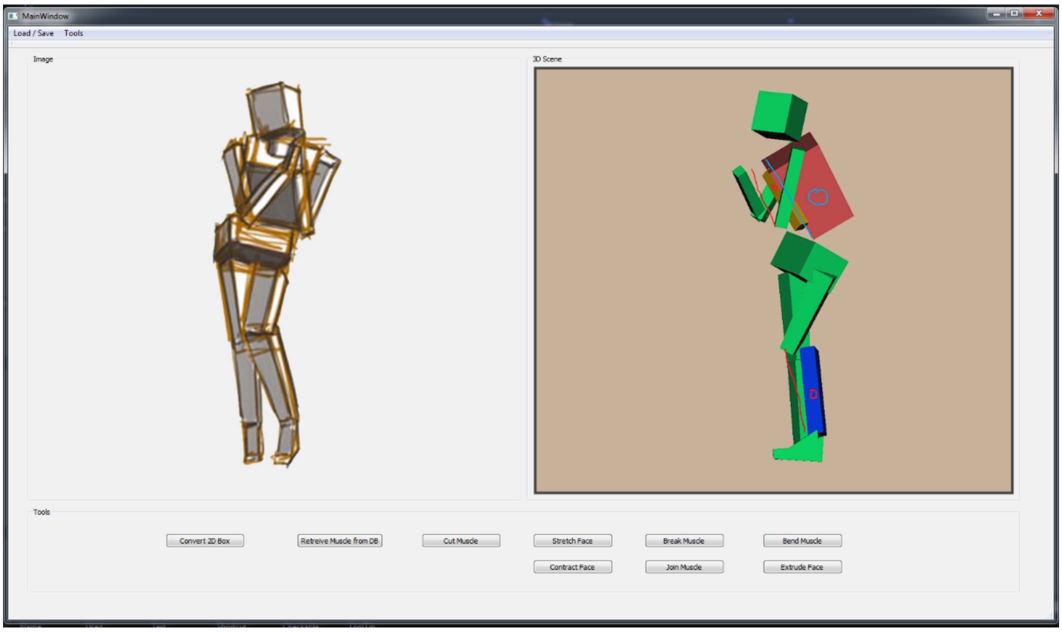
where $\{\lambda^C, E^C\}$ is used for the connectivity.

In practice, we solve the equation sets (5.7) and (5.8) iteratively until convergence to achieve the desired pose.

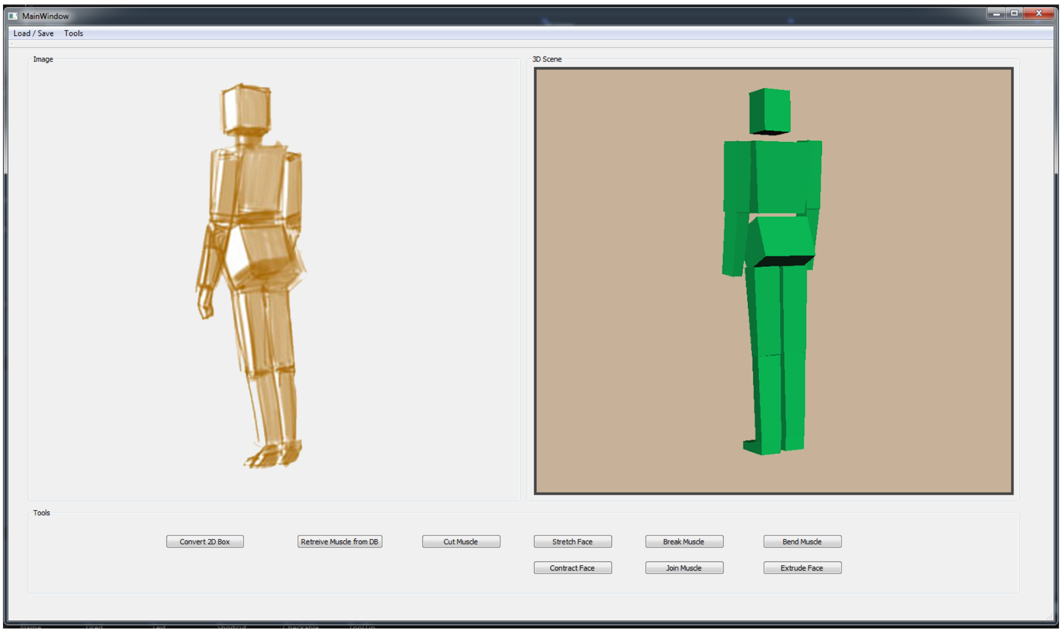
Figures 70(a-d) demonstrate several characters being created using the *box modeller* component. Figure 70(b) shows how a user can provide additional gestures to allow the system to automatically align the boxes.



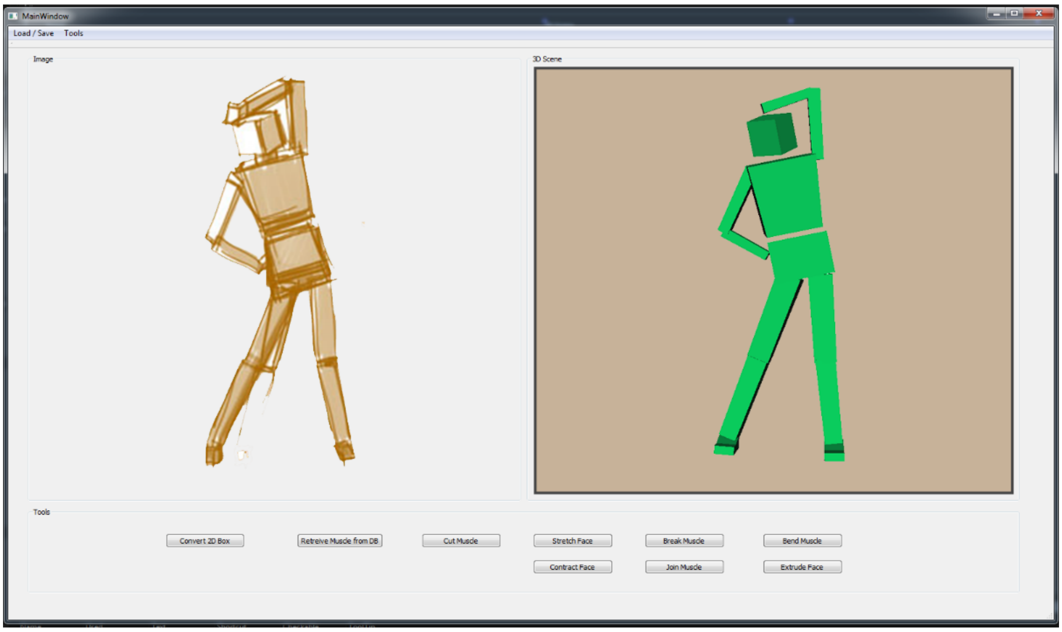
(a)


MainWindow
Load / Save
Tools
Image
3D Scene
Tools
Convert 2D Box
Retreive Muscle from DB
Cut Muscle
Stretch Face
Break Muscle
Bend Muscle
Contract Face
Join Muscle
Extrude Face

(b)


MainWindow
Load / Save
Tools
Image
3D Scene
Tools
Convert 2D Box
Retreive Muscle from DB
Cut Muscle
Stretch Face
Break Muscle
Bend Muscle
Contract Face
Join Muscle
Extrude Face

(c)



(d)

Figure 70(a-d): Some results showing boxy characters modelled using the Box Modeller component.

5.3.2 Muscle Sculptor

In order to make the human figure more detailed and realistic, the second component of the system allows the artist to refine / modify the 3D boxes created with the box modeller to resemble the actual human body parts and muscles.

The muscle sculptor component introduces two modelling paradigms. 1) Sketch-based and data-driven muscle modelling, and 2) Sketch-based muscle sculpting. The first paradigm is developed to aid novice artists / users who are new to human figure modelling, while the latter is focused at professional artists who possess prior knowledge of human anatomy and human figure sculpting.

5.3.2.1 Sketch-based and data-driven muscle modelling: For the novice artist, the system provides tools to aid in creating plausible boxy muscles on relevant parts of the human body. Figure 71 shows the steps performed by the muscle sculptor to aid the novice artist in achieving a plausible model for the leg. To this aim, the system allows the artist to roughly draw the muscles on top of the image and then use the automated tools to retrieve a closely matching muscle from the database. The system then deforms the retrieved boxy muscle into a more refined boxy muscle to adhere the surface anatomy.

The database created for the system contains the set of muscles that most commonly visible in surface anatomy drawings. The muscles were pre-modeled using Autodesk Maya (Maya) and then stored in the database instantaneously using the OBJ file format. Thus the duration of storing pre-modeled muscles solely depends on the time it takes to model the muscle. For each muscle three different variants were stored to allow the user to choose from amongst the stored variants. For instance there are three different versions for the 'deltoid' muscle. One of the future works in the proposed system is to evaluate the database on the basis of the number of models it can hold to maintain overall system performance. It was observed that the system did not respond as expected if the artist provided a sketch that could not be matched with any model in the database even when the sketch was aesthetically plausible, which is another limitation that need some attention in future.

Along with storing boxy muscles in a database, the system also stores an OBB representation for each boxy muscle. The OBB tree represents the muscle as a boxy shape. The reason for using an OBB tree representation is that there are certain muscles that can be modelled as a set of separate boxes and then later combined to create a complete muscle. Thus this representation will give a novice artist a clue on how to model complex muscles such as 'sartorius' muscle. The OBB tree representation is later used for cage-based deformation. OBB tree was originally developed to solve the problems vested with collision detection of complex polygonal models. The OBB-tree hierarchy construction method was presented by Gottschalk et al. (1996) and operates in a top-down fashion.

After the system determines the OBB representation of the muscle, it needs to deform the smooth muscle into its boxy muscle. For deformation, cage-based deformation has been used because of its implementation simplicity and suitability with computing correspondences on low polygon mesh. The cage-based deformation algorithm is governed by Mean Value Coordinates. Mean Value Coordinates one of the most efficient cage-based deformation algorithms as concluded by Nieto and Susin (2013) in their survey. The proposed system uses the algorithm presented by Floater (2003) to compute the deformation. For a mesh with vertices V and edges E the Mean-Value Coordinates for each vertex $v_i \in V$ is computed from the Euclidean coordinates of the vertex and its m neighbour vertices v_j , where $(i,j) \in E$.

$$v_i = F_i(V) = \sum_{(i,j)\in E} w_{ij} \left\lfloor v_j - \left(d_i + v_j \bullet n_i\right) n_i \right\rfloor + h_i n_i \tag{5.6}$$

Here h_i is the vertex offset above the projection plane. w_{ij} are the weight functions and normalizing each weight function by the sum of all weight functions gives us the Mean Value coordinates. d_i is the average distance from the origin. To achieve final deformation of the mesh, we solve the following energy minimization deformation functional using the Gauss-Newton algorithm:

$$\arg\min_V G(V) = \frac{1}{2} \sum_{v_i \in V} (v_i - F_i(V))^2 \tag{5.7}$$

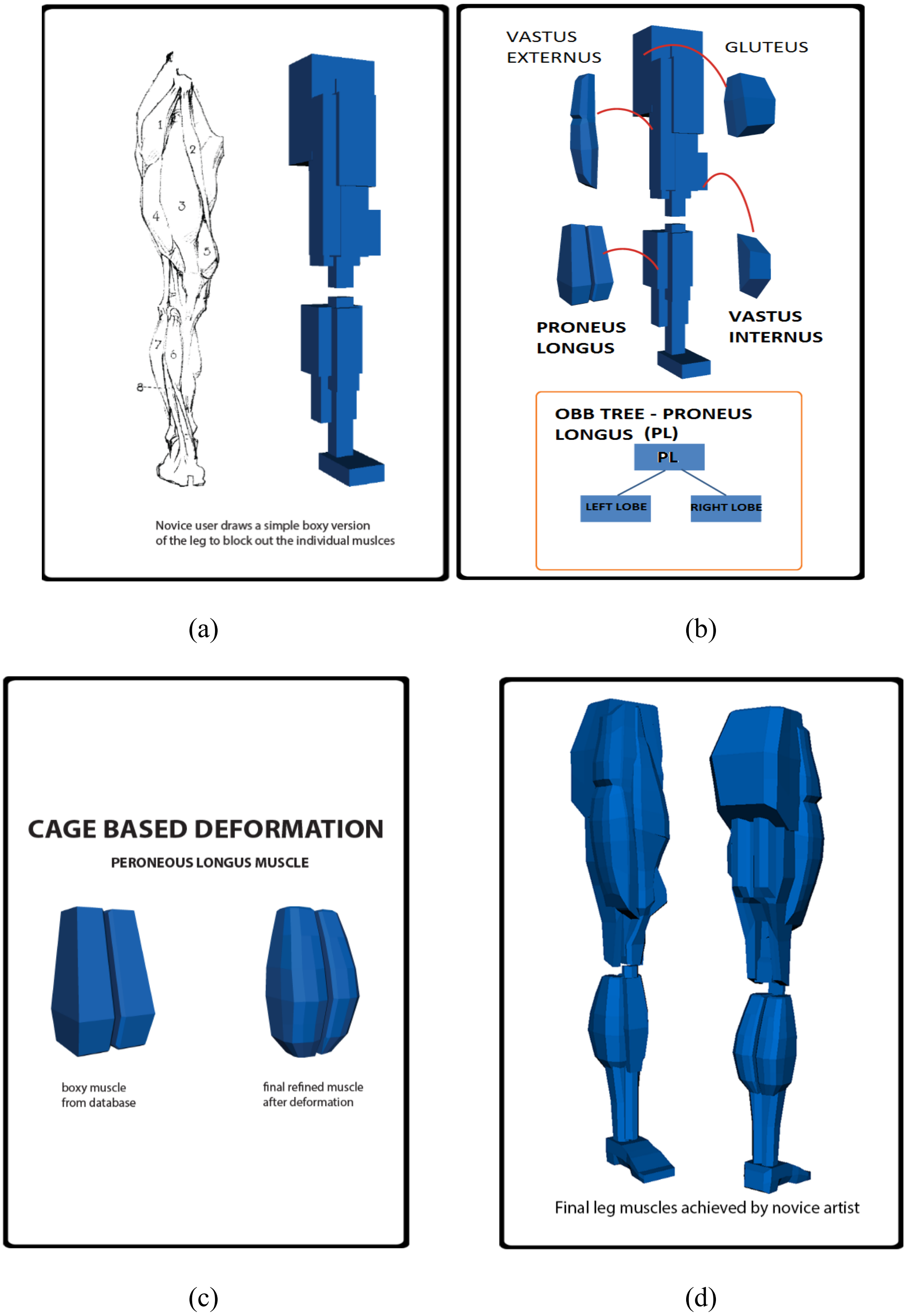


Figure 71: A process showing the steps involved with aiding novice artists to model deltoid muscle.

The computation of MVC based deformation is iterative and it takes a maximum of 5 iterations to achieve reasonable deformations with each iteration taking between 0.01 to 0.03 seconds.

5.3.2.2 Sketch-based muscle sculpting: For a professional artist, this component offers several operations to allow the artist to easily sculpt the boxes to closely match the human body muscles. All these operations are activated via sketching. A brief overview of these operations is given below.

Cut muscle: The main purpose of this tool is to allow the artist to cut a box by placing a single stroke or a couple of strokes directly over a face of the boxy muscle. The Bridgman approach focuses heavily on wedged shaped boxes to indicate the flow of human muscles and how these are connected to each other. Using this operation an artist can cut a portion of the muscle. Cutting the box can be achieved using a single stroke or two strokes. If the artist wants to cut the box in a direction perpendicular to a particular face, he/she can simply draw a single stroke on that face and the system automatically cuts the box perpendicular to the stroked face. Figure 72 shows an example of this operation.

The 'cut muscle' tool creates a 3D polygon which is oriented according to the strokes provided by the artist and then uses a 3D variant of the Greiner–Hormann clipping algorithm to achieve the final shape (Greiner and Hormann, 1998). The Greiner–Hormann clipping algorithm is an efficient algorithm for clipping arbitrary 3D-polygons. The algorithm can handle arbitrary closed polygons, specifically where the clip and subject polygons may self-intersect.

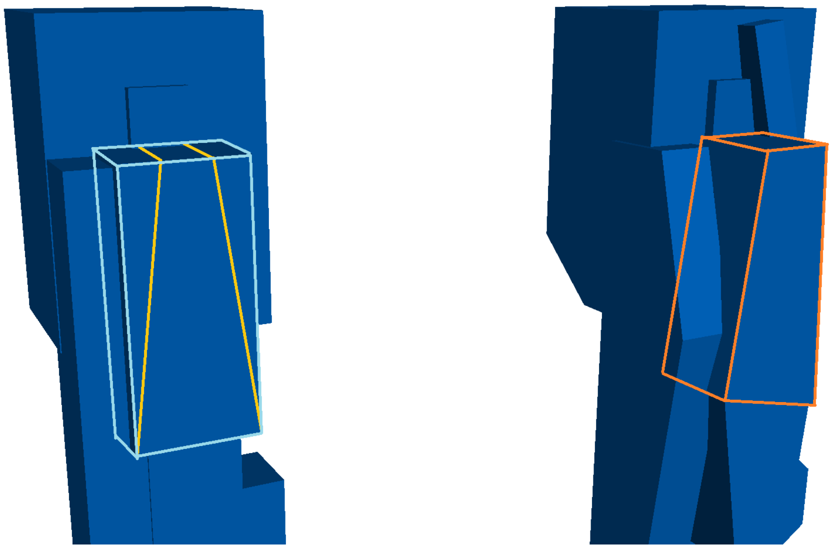


Figure 72: Cutting a mesh using the cutting tool. The left image shows the cutting lines (in yellow) placed by the user on the 3D box (blue highlight). The right image shows the resulting mesh (orange highlight) obtained after the system has performed the cutting operation.

Stretch / Contract face: During the processs of modelling surface anatomy, an artist may want to stretch or contract a specific face of the boxy muscle to achieve the desired shape of the muscle. For example, in modelling a forearm's bicep muscles, an artist can initially draw a 3D box in arbitrary dimensions, and then use this tool to scale a face appropriately as shown in the Figure 73.

This tool was simple to implement as it first modifies the vertex positions of the face and subsequently updates the vertex positions of the adjacent faces that were affected due to this operation.

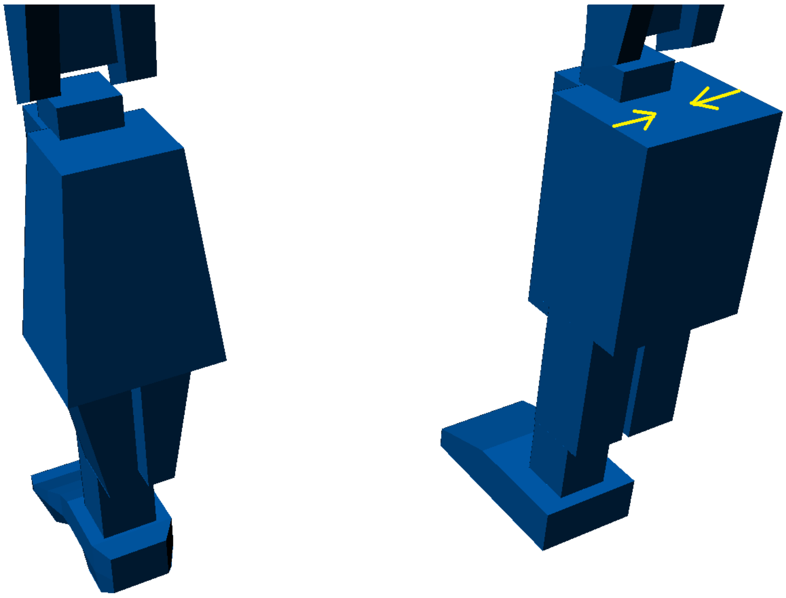


Figure 73: The process of stretching/contracting a face of a box.

Break Muscle: This operation divides a boxy muscle into two pieces by first selecting a particular boxy muscle. The need for this operation arrives when the artist wants to model abdominal muscles, as these muscles are composed of six block shaped muscles arranged in a grid shaped pattern (Figure 74). To utilize this tool, the artist first selects the boxy muscle by drawing a rough ellipse around it, and then draws a line across the boxy muscle (where the artist wants to divide the muscle) in order to break the muscle into two separate muscles.

The line drawn by the artist to break the box was projected into the 3D space and an invisible polygon was created propagating at an arbitrary position in the negative z-axis. After adjusting the length of the invisible plane, a KD-Tree data structure was used to find the nearest vertices on the boxy muscle. The box is then detached into two separate pieces.

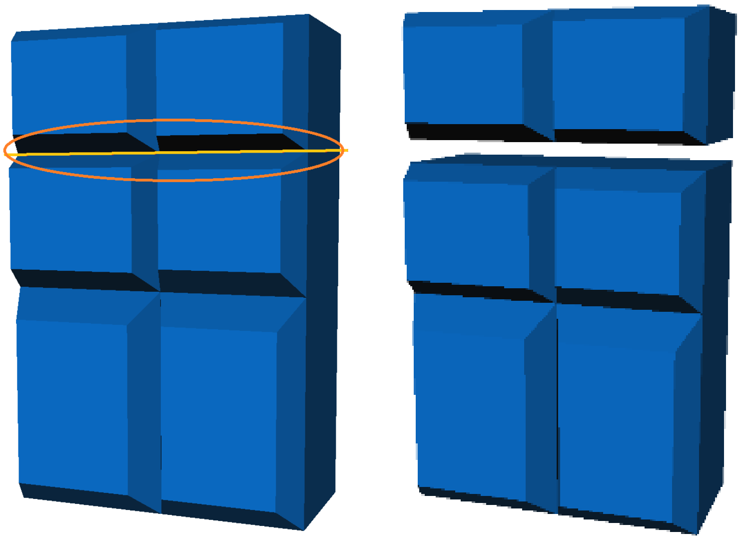


Figure 74: Abdominal muscles modelled using ‘break muscle’ operation.

Join muscles: An artist can initially draw all the 3D boxes representing individual muscles in separate parts, and then later combine these parts to depict one muscle. This tool is useful for modelling the ‘Supinator Longus’ muscle in the human arm. In Figure 75, the image on the right shows the three separate muscles that are first modelled using boxes. The boxes are then cut using the ‘Cut Muscle’ tool to achieve the desired shape. The artist is then required to draw roughly parallel strokes in order to join the separate muscle. These 3 sculpted muscles are then joined by the system to create the entire ‘Supinator Longus’ muscle.

Similar to the ‘break muscle’ tool, this tool constructs invisible polygons using the drawn strokes and then uses the KD-tree data structure to find the nearest vertices on the boxy muscle. The faces to which the vertices belong are then stitched together by simply updating the vertex positions of the affected faces and finally grouping the faces together into one single model.

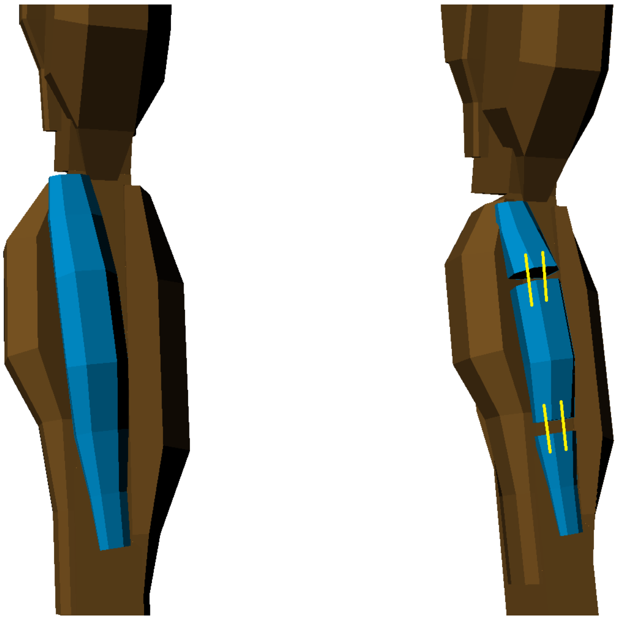


Figure 75: Joining of the Supinator Longus muscle using simple sketching gesture.

Bend muscle: This operation bends the muscle in a boxy fashion and can be used to model the External Oblique muscles that take the form of a belt wrapped around the waist. Another muscle that can be modelled using this feature is 'Sartorius' muscle which is located in the thighs (Figure 76). The artist starts by drawing a box and then scales the box along the desired direction. He/she then sculpts the box using the Cut Tool if necessary, and finally draws a curved stroke to specify the bending direction of the muscle.

The bend muscle tool treats the curvature of the stroke as a polyline and simplifies it. It projects the polyline onto the surface of a boxy muscle on which the artist creates the stroke. The system then uses the break tool and breaks the muscle to be bent into several individual boxy muscles and orients these along the curvature of the stroke/polyline. Finally the system uses the 'join muscle' tool to join all the individual muscles to achieve the final shape.

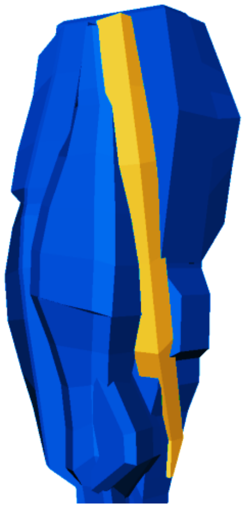


Figure 76: Sartorius muscle (yellow) modelled using the bend muscle operation.

Extrude face: This tool has proven to be useful in modelling different muscles such as Trapezius, Latissimus Dorsi and Abdomical muscles. The user can also use this tool to model the nose and chin of the human model. To use this tool, the artist selects a particular face and extrudes it using a stroke gesture in the direction as shown in the Figure 77.

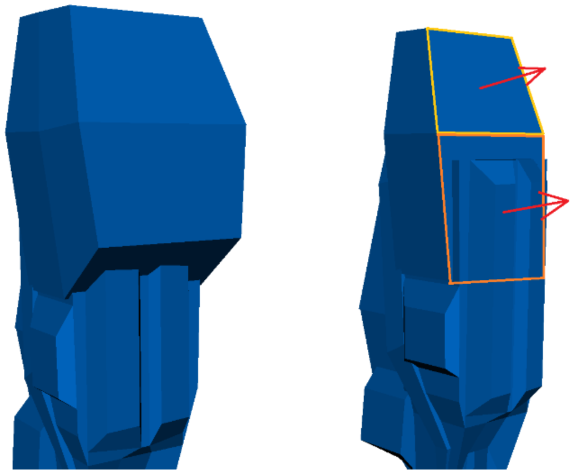


Figure 77: Extrude face operation using simple sketching gesture.

5.4 Results & Discussion

In this section the user interface of the system and a few results are presented to demonstrate the effectiveness of the proposed approach.

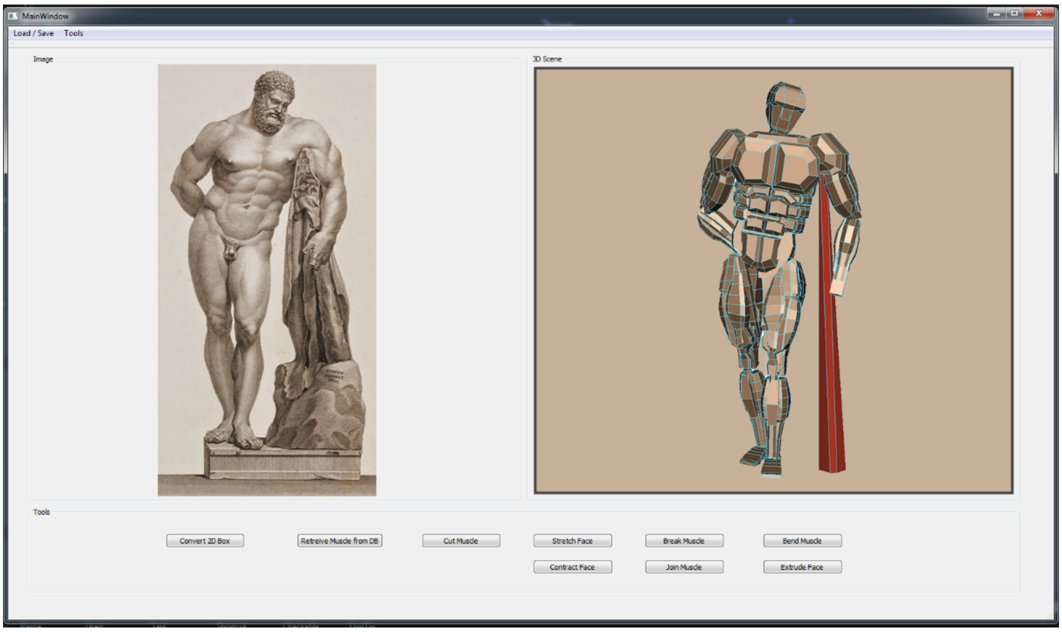


Figure 78: Main interface of the proposed system. *Sketch by Francesco Piranesi c. 1795.*

Figure 78 shows the main interface with a final model of the character from a sketch of the famous Hercules statue. As shown in the figure, the interface provides several controls to the user for sculpting the boxy model into a more defined character model. Figure 79 shows iterative process of a human arm being modelled and refined by a user using the Muscle Sculptor component.

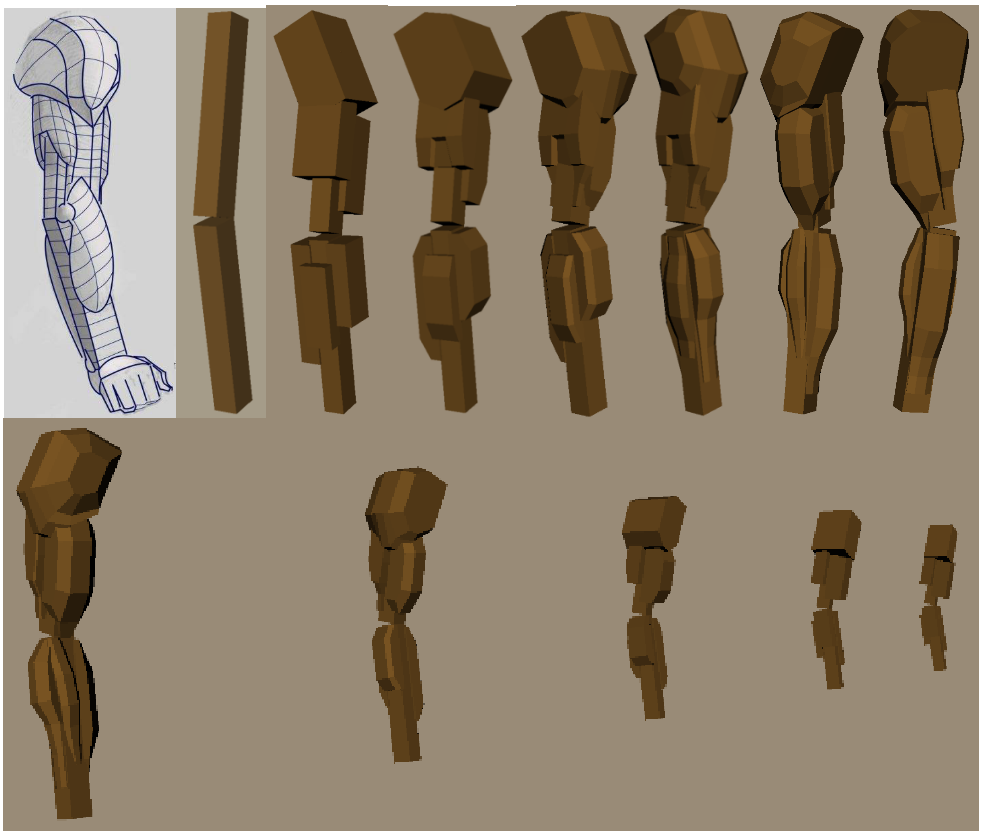


Figure 79: The process of modelling and sculpting a human arm by the user.

Figure 80 shows the human leg being modelled and refined using the muscle sculptor component.

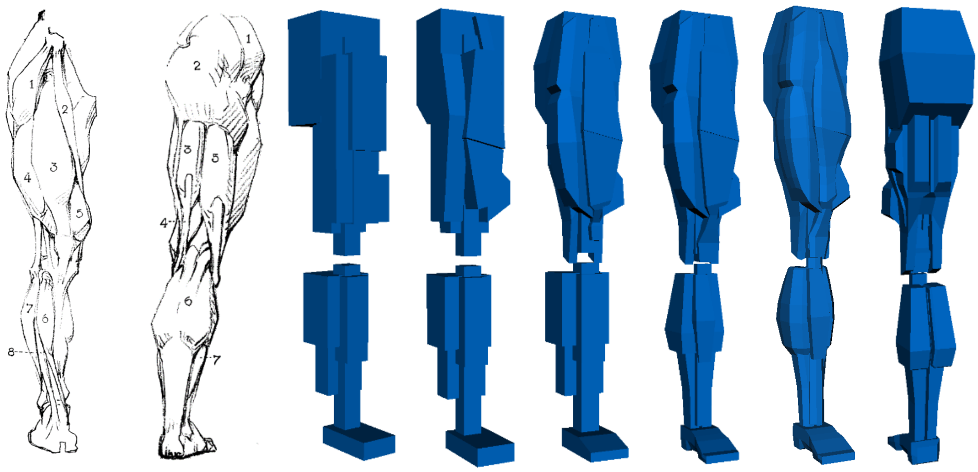


Figure 80: Iterative modelling of the leg using muscle sculptor tool.

The character model presented in Figure 81 was created in over eight hours by a novice user. Although the proposed system aims to support swift creation of models, the creation of models that are more organic looking is difficult to achieve. The tools developed in this system can be refined in future to allow a novice user to construct a character that can achieve an organic look. The system has been tested with three research students so far but in future there is a need to test the tool with artists with varying skill levels.

Table 5: Feedback collated from users of the proposed system.

Question	Strongly Agree	Agree	Undecided	Disagree	Strongly Disagree
It was easy to upload an input sketch using the interface	✓				
It was easy to draw 2D boxes over the input sketch		✓			
It was easy to use the tools provided by ‘Box Modeller’ to generate 3D boxes from 2D boxes		✓			
It was easy to rotate, translate and scale the boxy model using the controls in the 3D window		✓			
It was easy to build a complete boxy model from scratch in the desired pose		✓			
Based on the input sketch of the muscle, it was easy to retrieve a muscle from the database by using the interface features				✓	
It was easy to insert the retrieved muscle in the model				✓	
It was easy to use the cut tool provided by the Muscle Sculptor		✓			
It was easy to use the stretch/contract muscle tool provided by the Muscle Sculptor		✓			
It was easy to use the break muscle tool provided by the Muscle Sculptor			✓		
It was easy to use the bend muscle tool provided by the Muscle Sculptor				✓	
The final model was of acceptable quality		✓			
The final model needs further work to make it high quality	✓				
Overall it was a unique experience using the system	✓				
It was quick to model a 3D character from scratch				✓	
It was more convenient to model a character using this system as compared a professional modelling system such as Maya			✓		
I would recommend this system to other artists		✓			

For testing the system, a survey in the form of a questionnaire was carried out and given to each user of the system to provide feedback on the usability of the system (Table 5). The table shows a set of questions and a set of possible choices. The answers in the table are the average answers provided by all the students. From the table below it is apparent that the data-driven features of the system needs refinement and perhaps re-designing in future. The muscle sculpting tools also needs refinement.

The users also reported that they would like to see more features in the system such as undoing previous actions as well we being able to highlight a set of selected muscles in order to delete or copy these.

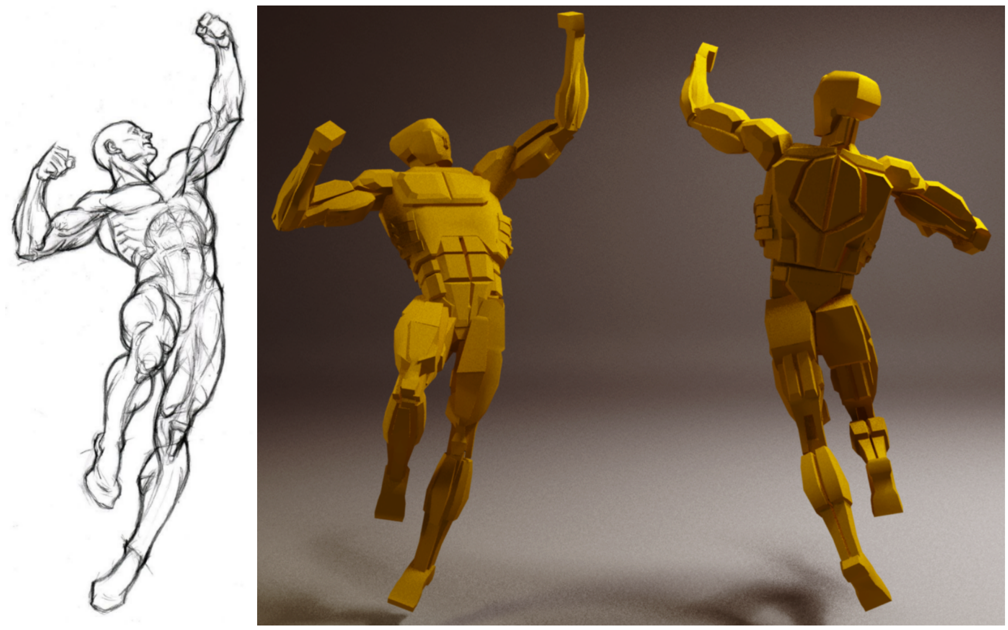


Figure 81: Final result of the character modelled from the source sketch above.

Figure 82 presents yet another result of the proposed system. This model was created by another user in a little over five hours. The noticeable difference between the modelling time of the model in Figure 81 and Figure 82 is due to reusing the arms and legs that were already model for the character in Figure 81.

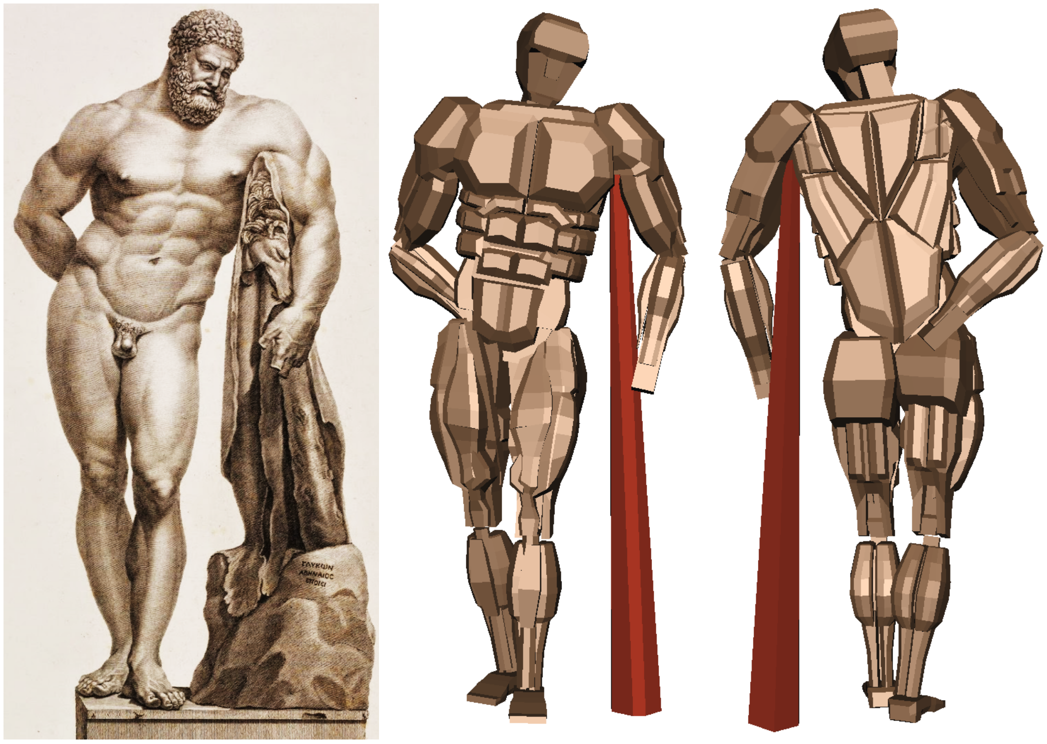


Figure 82: Another result of a modeled character using the proposed system.

Sketch by Francesco Piranesi c. 1795.

5.5 Conclusions and future work

In this chapter, a novel sketch-based modelling system is presented that demonstrates the generation of anatomically correct human characters by sculpting boxes. The proposed system is just an initiating effort in a new direction of sketch-based modelling tools and it is hoped that this system will provide a useful tool for young and professional artists to improve their character modelling skills. As explained in Section 5.4, the system has been tested with a few research students, and mostly positive feedback was received regarding the effectiveness and practicality of this tool. The feedback was gathered in the form of a questionnaire focusing on usability of tools and the quality of models created. The system was deployed on a windows environment and asked the users to draw the boxes using a pen and tablet device.

Due to the anatomical correctness of the models created from the system, these models can be fed into the detailed modelling phase within the animation production pipeline, where the 3D artist will feel comfortable in adding more details to the character with greater ease.

While it can be said with confidence that the proposed system will serve a useful purpose for artists in the early design phase to make prototypical characters, there are aspects that need to improvement in future. At the moment the system supports the creation of human figures which are relatively muscular with body builder physiques. Therefore this system may not be suitable for modelling obese or slim human bodies.

6 Conclusion and Future Work

6.1 Conclusion

3D character modelling is a challenging job for 3D artists, especially for those coming from a traditional arts background. Many 3D artists in particular find it difficult to transform 2D concept characters into 3D characters, as they are not trained in traditional 2D arts. On the other hand, many 2D artists find it difficult to create 3D models as they do not possess the necessary skills in 3D.

Sketch-based modelling is considered to be one of the solutions that provide a bridge between traditional 2D character drawing and modern 3D character modelling. Thus in this thesis, several contributions have been proposed novel approaches that aim to answer the question posed at the beginning of this thesis (Chapter 1 - Introduction).

In terms of the key contributions, the first sketch-based modelling method presented in this thesis (Chapter 3) provides a novel pipeline of sketch-based character modelling, while also overcoming the limitations present in the previous techniques of a similar nature. Particularly, the proposed approach has contributed towards the computation of accurate occluding contours, which is also fast as compared to the work by Kraevoy et al. (2009). This new approach combines the principle behind human vision, intelligence and interactions with computer's powerful computing capacity to achieve accurate and quick correspondences between the 3D template model and user's drawn sketches, and presents a hybrid mesh deformation technique to change the 3D template model into new character models efficiently. The presented hybrid mesh deformation technique maximizes the strength of skeleton-based deformation in dealing with large mesh displacements and deformations globally and that of mesh editing algorithms in tackling small mesh deformations locally. The developed user interface integrates all the functions ranging from creating a character dataset, interactive sketch creation, retrieval of a most similar 3D template model, fast and accurate correspondences, and hybrid mesh deformation. A few examples were presented to demonstrate efficiency and quality of the proposed approach.

The second proposed approach explained in Chapter 4 presents a hybrid sketch-based modelling system combining the features from existing systems such as (Gingold et al., 2009) and shape-from-shading (Yang and Han, 2007) to allow artists to quickly model the base mesh of the character as well as to add details to the base mesh. The system provides intuitive tools to add details to the base mesh. In contrast to the popular commercial animation packages such as Autodesk Maya (Maya), which have a steep learning curve, the system allows the artists to quickly create a base mesh from a single view via simple strokes, and utilizing the power of primitives (generalized cylinders, cubes, and ellipsoids). Moreover, the system allows the artists to quickly add details to the surface of the base mesh by generating a 3D surface from a photo with minimum effort. It is believed that the system makes a fresh attempt at pushing the sketch-based modelling systems beyond their boundaries.

The final proposed approach in Chapter 5 presents a novel sketch-based modelling system that demonstrates the generation of anatomically correct human characters by sculpting boxes. This system is just an initiating effort in a new direction of sketch-based modelling tools and it is hoped that the system will provide a useful tool for young and professional artists to improve their character modelling skills.

6.2 Discussion and future work

Though the results presented in Chapter 3 demonstrate the effectiveness of the proposed system, the quality of deformation achieved varies with the number of polygons present in the mesh. With models having polygon count between 30,000 and 90,000 the overall system performs very efficiently and produces highly plausible results, however with higher sizes of meshes, the deformation process becomes slower and produces slightly inaccurate results. Here ‘size’ refers to the number of polygons in the mesh. This slow behavior is due to the fact that the efficiency of the automatic skeleton based deformation algorithm is directly proportional to the mesh size, which eventually affects the final mesh deformation. It can therefore be concluded that the proposed approach in Chapter 3 is best suited for characters of medium sized meshes mostly used in games, and not suitable for hyper-realistic characters used in movies.

As pointed out in Chapter 3, there are some more limitations of the proposed approach. Firstly, the current dataset only includes male human character models. In future, there is a possibility to evolve the dataset into a larger version involving various female models and non-human character models. Secondly, all the character models in the dataset are currently in either A-pose or T-pose. It would be interesting the add characters in various articulate poses and see the effect it can do template retrieval step. Thirdly, manually drawing occluding contours is slightly difficult for the first time users. This can be avoided by developing an automatic algorithm to generate the occluding contours.

The work in Chapter 3 was presented at 28th Annual Conference on Computer Animation and Social Agents (CASA 2015) at Nanyang Technological University, Singapore. On presenting the work, one of the attendees suggested that one possible future research direction would be to add support for sketch based clothes modeling.

The system proposed in Chapter 4 can be further improved through the following work:

1. More features can be added to control and manipulate the primitives smoothly, and add details to the base mesh. Currently a user can only modify the vertices of the mesh. In future, it would be interesting to add modification controls/gestures for faces and edges.
2. The controls can be added to deform the feature curves of the modelled character to better match those of the input sketch.
3. A feature can be added in the system interface to allow the artist to automatically stitch/transfer the details from SFS algorithm onto the base mesh without any manual interaction.

As pointed out in Chapter 4, this system can also be integrated with the system proposed in Chapter 3 in order to potentially improve the quality of the generated character models even further.

Finally, the work presented in Chapter 5 is still a work in progress. It was initially conceived as a graphic software system to train novice artists the skills of human figure sculpting with a different approach than Autodesk Mudbox (Mudbox). It is believe that the end product of this system will also allow me to conduct studies and survey what approaches novice artists use to learn human figure modelling, and whether the use of drawing principles developed by George Bridgman will be beneficial in making a novel and more efficient sketch-based modelling system. This idea has been discussed with some of the world's leading experts in Computer Graphics and received encouraging feedback. This work was also submitted to SIGGRAPH 2016 as a research paper and received very constructive feedback from the reviewers. As a future work it is planned to improve the quality of this work based on the reviews, and also develop this idea further.

One potential application of the proposed system in Chapter 5 is to be used a teaching tool for fine art students at arts schools/colleges. In a physical setting, this can be visualized in a way that in a fine arts class, instead of using a traditional drawing easel and pencil, drawing tablets such as Wacom Cintiq can be installed running the proposed system, thus allowing the students to create 3D human figures. It is believed that a more enhanced and robust data-driven support to the system can boost the learning process of the user, by allowing the user to accurately retrieve a desired muscle from the muscle database. Moreover, with a more enhanced and user friendly experience, this system will have the potential to be adopted by novice artists in animation studios.

References

Akgül, C.B., 2007. *Density-based shape descriptors and similarity learning for 3D object retrieval* (Doctoral dissertation, Ph. D. thesis, Dept. Signals-Images, Télécom ParisTech, Paris).

Andre, A. and Saito, S., 2011, August. Single-view sketch-based modelling. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modelling* (pp. 133-140). ACM.

Asada, H. and Brady, M., 1986. The curvature primal sketch. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1), pp.2-14.

Bae, S.H., Balakrishnan, R. and Singh, K., 2008, October. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (pp. 151-160). ACM.

Belongie, S., Malik, J. and Puzicha, J., 2002. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24*(4), pp.509-522.

Bénard, P., Hertzmann, A. and Kass, M., 2014. Computing smooth surface contours with accurate topology. *ACM Transactions on Graphics (TOG), 33*(2), p.19.

Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM, 18*(9), pp.509-517.

Bernardini, F. and Rushmeier, H., 2002, June. The 3D model acquisition pipeline. In *Computer graphics forum* (Vol. 21, No. 2, pp. 149-172). Blackwell Publishers Ltd.

Berrada, F., Aboutajdine, D., Ouatik, S.E. and Lachkar, A., 2011, April. Review of 2D shape descriptors based on the curvature scale space approach. In *Multimedia Computing and Systems (ICMCS), 2011 International Conference on* (pp. 1-6). IEEE.

Bium, H., A transformation for extracting new descriptions of shape. In *Symposium on Modeis for the Perception of Speech and Visua1 Form*.

Blender Foundation. 2016. *Blender* (2016). [Software]

Bober, M., 2001. MPEG-7 visual shape descriptors. *IEEE Transactions on circuits and systems for video technology, 11*(6), pp.716-719.

Borgefors, G., 1984. Distance transformations in arbitrary dimensions. *Computer vision, graphics, and image processing, 27*(3), pp.321-345.

Bridgman, G.B., 1920. Constructive anatomy (Vol. 501). Courier Corporation.

Buchanan, P., Mukundan, R. and Doggett, M., 2013, July. Automatic single-view character model reconstruction. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modelling* (pp. 5-14). ACM.

Bustos, B., Keim, D., Saupe, D. and Schreck, T., 2007. Content-based 3D object retrieval. *Computer Graphics and Applications, IEEE, 27*(4), pp.22-27.

Canny, J., 1986. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6), pp.679-698.

Cashman, T.J. and Fitzgibbon, A.W., 2013. What shape are dolphins? building 3d morphable models from 2d images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35*(1), pp.232-244.

Celebi, M.E. and Aslandogan, Y.A., 2005, April. A comparative study of three moment-based shape descriptors. In *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II* (Vol. 1, pp. 788-793). IEEE.

Chaikin, G.M., 1974. An algorithm for high-speed curve generation.*Computer graphics and image processing*, *3*(4), pp.346-349.

Chen, B.Y., Wu, F.C., Lin, T.Y. and Su, M.C., 2009, October. Skeleton constrained dual-resolution modelling for sketch-based deformation. In *Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference* (pp. 1-7). Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, International Organizing Committee.

Chen, D.Y., Tian, X.P., Shen, Y.T. and Ouhyoung, M., 2003, September. On visual similarity based 3D model retrieval. In *Computer graphics forum* (Vol. 22, No. 3, pp. 223-232). Blackwell Publishing, Inc.

Chen, T., Zhu, Z., Shamir, A., Hu, S.M. and Cohen-Or, D., 2013. 3-sweep: Extracting editable objects from a single photo. *ACM Transactions on Graphics (TOG), 32*(6), p.195.

Chuang, G.C. and Kuo, C.J., 1996. Wavelet descriptor of planar curves: Theory and applications. *Image Processing, IEEE Transactions on, 5*(1), pp.56-70.

Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H.S., Finkelstein, A., Funkhouser, T. and Rusinkiewicz, S., 2008, August. Where do people draw lines?. In *ACM Transactions on Graphics (TOG)* (Vol. 27, No. 3, p. 88). ACM.

Cook, M.T. and Agah, A., 2009. A survey of sketch-based 3-D modelling techniques. *Interacting with Computers, 21*(3), pp.201-211.

Daniele, T., 2012. *Poly-modeling with 3ds Max: Thinking Outside of the Box*. CRC Press.

Daras, P. and Axenopoulos, A., 2009, June. A compact multi-view descriptor for 3D object retrieval. In *2009 Seventh International Workshop on Content-Based Multimedia Indexing* (pp. 115-119). IEEE.

Daras, P. and Axenopoulos, A., 2010. A 3D shape retrieval framework supporting multimodal queries. *International Journal of Computer Vision,89*(2-3), pp.229-247.

Daras, P., Tzovaras, D., Dobravec, S., Trnkoczy, J., Sanna, A., Paravati, G., Traphoener, R., Franz, J., Kastrinogiannis, T., Malavazos, C. and Ploskas, N., 2008, November. Victory: a 3d search engine over p2p and wireless p2p networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet (p. 49). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).*

Daras, P., Zarpalas, D., Tzovaras, D. and Strintzis, M.G., 2004. 3D model search and retrieval based on the spherical trace transform. In *Multimedia Signal Processing, 2004 IEEE 6th Workshop on* (pp. 335-338). IEEE.

DeCarlo, D., Finkelstein, A., Rusinkiewicz, S. and Santella, A., 2003. Suggestive contours for conveying shape. *ACM Transactions on Graphics (TOG), 22*(3), pp.848-855.

deMartin, J. 2016. *Jon deMartin's Official Website*. [Online]. [Accessed 5th December 2014]. Available at: www.jondemartin.net

DeRose, T., Kass, M. and Truong, T., 1998, July. Subdivision surfaces in character animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (pp. 85-94). ACM.

di Baja, G.S. and Svensson, S., 2002. A new shape descriptor for surfaces in 3D images. *Pattern Recognition Letters, 23*(6), pp.703-711.

Durou, J.D., Falcone, M. and Sagona, M., 2008. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding, 109*(1), pp.22-43.

Eitz, M., Richter, R., Boubekeur, T., Hildebrand, K. and Alexa, M., 2012. Sketch-based shape retrieval. *ACM Trans. Graph., 31*(4), pp.31-1.

En.wikipedia.org. (2017). *Autodesk 3ds Max*. [online] Available at: https://en.wikipedia.org/wiki/Autodesk_3ds_Max [Accessed 2 Mar. 2017].

En.wikipedia.org. (2017). *Autodesk Maya*. [online] Available at: https://en.wikipedia.org/wiki/Autodesk_Maya [Accessed 2 Mar. 2017].

En.wikipedia.org. (2017). *Autodesk Mudbox*. [online] Available at: https://en.wikipedia.org/wiki/Autodesk_Mudbox [Accessed 2 Mar. 2017].

En.wikipedia.org. (2017). *ZBrush*. [online] Available at: https://en.wikipedia.org/wiki/ZBrush [Accessed 2 Mar. 2017].

Floater, M.S., 2003. Mean value coordinates. *Computer aided geometric design, 20*(1), pp.19-27.

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D. and Jacobs, D., 2003. A search engine for 3D models. *ACM Transactions on Graphics (TOG), 22*(1), pp.83-105.

Gingold, Y., Igarashi, T. and Zorin, D., 2009, December. Structured annotations for 2D-to-3D modelling. In *ACM Transactions on Graphics (TOG)* (Vol. 28, No. 5, p. 148). ACM.

Gottschalk, S., Lin, M.C. and Manocha, D., 1996, August. OBBTree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (pp. 171-180). ACM.

Greiner, G. and Hormann, K., 1998. Efficient clipping of arbitrary polygons. *ACM Transactions on Graphics (TOG)*, *17*(2), pp.71-83.

Grimm, C. and Joshi, P., 2012, June. Just DrawIt: a 3D sketching system. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modelling* (pp. 121-130). Eurographics Association.

Guay, M., Cani, M.P. and Ronfard, R., 2013. The line of action: an intuitive interface for expressive character posing. *ACM Transactions on Graphics (TOG), 32*(6), p.205.

Hahn, J., Qiu, J., Sugisaki, E., Jia, L., Tai, X.C. and Seah, H.S., 2013. Stroke-based surface reconstruction. *Numerical Mathematics: Theory, Methods and Applications, 6*(01), pp.297-324.

Igarashi, T., Matsuoka, S. and Tanaka, H., 1999. Teddy: A Sketching Interface for 3D Freeform Design. In *ACM SIGGRAPH'99.*

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y. and Ramani, K., 2005. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design, 37*(5), pp.509-530.

Jayanti, S., Kalyanaraman, Y., Iyer, N. and Ramani, K., 2006. Developing an engineering shape benchmark for CAD models. *Computer-Aided Design,38*(9), pp.939-953.

Joshi, P., Meyer, M., DeRose, T., Green, B. and Sanocki, T., 2007, August. Harmonic coordinates for character articulation. In *ACM Transactions on Graphics (TOG)* (Vol. 26, No. 3, p. 71). ACM.

Karpenko, O.A. and Hughes, J.F., 2006, July. SmoothSketch: 3D free-form shapes from complex sketches. In *ACM Transactions on Graphics (TOG)*(Vol. 25, No. 3, pp. 589-598). ACM.

Kazmi, I.K., You, L. and Zhang, J.J., 2013, August. A survey of 2D and 3D shape descriptors. In *Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference* (pp. 1-10). IEEE.

Khoje, S. and Bodhe, S., 2012. Performance Comparison of Fourier Transform and its Derivatives as Shape Descriptors for Mango Grading. *International Journal of Computer Applications, 53*(3).

Koenderink, J.J., 1984. What does the occluding contour tell us about solid shape?. *Perception, 13*(3), pp.321-330.

Koenderink, J.J., 1998. Pictorial relief. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 356*(1740), pp.1071-1086.

Kraevoy, V., Sheffer, A. and van de Panne, M., 2009, August. Modelling from contour drawings. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based interfaces and Modelling* (pp. 37-44). ACM.

Ku, D.C., Qin, S.F. and Wright, D.K., 2006. A sketching interface for 3D modelling of polyhedron. *Eurographics.*

Lam, L., Lee, S.W. and Suen, C.Y., 1992. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence, 14*(9), pp.869-885.

Lee, K.M. and Kuo, C.C., 1993. Shape from shading with a linear triangular element surface model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 15*(8), pp.815-822.

Lewis, J.P., Cordner, M. and Fong, N., 2000, July. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of* *the 27th annual conference on Computer graphics and interactive techniques* (pp. 165-172). ACM Press/Addison-Wesley Publishing Co.

Li, B. and Johan, H., 2013. 3D model retrieval using hybrid features and class information. *Multimedia tools and applications, 62*(3), pp.821-846.

Li, B., Godil, A., Aono, M., Bai, X., Furuya, T., Li, L., López-Sastre, R.J., Johan, H., Ohbuchi, R., Redondo-Cabrera, C. and Tatsuma, A., 2012, May. SHREC'12 Track: Generic 3D Shape Retrieval. In *3DOR* (pp. 119-126).

Liao, S.X. and Pawlak, M., 1998. On the accuracy of Zernike moments for image analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20*(12), pp.1358-1364.

Lipman, Y., Levin, D. and Cohen-Or, D., 2008, August. Green coordinates. In *ACM Transactions on Graphics (TOG)* (Vol. 27, No. 3, p. 78). ACM.

Lipp, M., Wonka, P. and Müller, P., 2014. PushPull++. *ACM Transactions on Graphics (TOG), 33*(4), p.130.

Liu, Q., 2012. A survey of recent view-based 3d model retrieval methods.*arXiv preprint arXiv:1208.3670.*

Liu, Y., Zha, H. and Qin, H., 2006, June. The generalized shape distributions for shape matching and analysis. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)* (pp. 16-16). IEEE.

Lowe, D.G., 1999. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). IEEE.

Mattesi, M., 2012. Force: Dynamic Life Drawing for Animators. *CRC Press.*

Mikhail, E.M., Bethel, J.S. and McGlone, J.C., 2001. Introduction to modern photogrammetry. *New York*.

Mokhtarian, F., Abbasi, S. and Kittler, J., 1996, September. Robust and E cient Shape Indexing through Curvature Scale Space. In *Proceedings of the 1996 British Machine and Vision Conference BMVC (Vol. 96)*.

Nam, S. and Chai, Y., 2012. SPACESKETCH: Shape modelling with 3D meshes and control curves in stereoscopic environments. *Computers & Graphics, 36*(5), pp.526-533.

Nealen, A., Igarashi, T., Sorkine, O. and Alexa, M., 2007, August. FiberMesh: designing freeform surfaces with 3D curves. In *ACM Transactions on Graphics (TOG)* (Vol. 26, No. 3, p. 41). ACM.

Nieto, J.R. and Susín, A., 2013. Cage-based deformations: a survey. In *Deformation models* (pp. 75-99). Springer Netherlands.

Ogniewicz, R.L. and Kübler, O., 1995. Hierarchic voronoi skeletons. *Pattern recognition, 28*(3), pp.343-359.

Ohbuchi, R. and Furuya, T., 2010, October. Distance metric learning and feature combination for shape-based 3D model retrieval. In *Proceedings of the ACM workshop on 3D object retrieval* (pp. 63-68). ACM.

Ohbuchi, R., Osada, K., Furuya, T. and Banno, T., 2008, June. Salient local visual features for shape-based 3D model retrieval. In *Shape Modelling and Applications, 2008. SMI 2008. IEEE International Conference on* (pp. 93-102). IEEE.

Olsen, L., Samavati, F.F., Sousa, M.C. and Jorge, J.A., 2009. Sketch-based modelling: A survey. *Computers & Graphics, 33*(1), pp.85-103.

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D., 2002. Shape distributions. *ACM Transactions on Graphics (TOG), 21*(4), pp.807-832.

Papadakis, P., Pratikakis, I., Theoharis, T., Passalis, G. and Perantonis, S., 2008. 3D Object Retrieval using an Efficient and Compact Hybrid Shape Descriptor. In *Eurographics Workshop on 3D object retrieval.*

Ping-Sing, T. and Shah, M., 1994. Shape from shading using linear approximation. *Image and Vision computing, 12*(8), pp.487-498.

Pooja, C.S., 2012. An effective image retrieval system using region and contour based features. In *IJCA proceedings on international conference on recent advances and future trends in information technology* (pp. 7-12).

Prasad, L., 1997. Morphological analysis of shapes. *CNLS newsletter,139*(1), pp.1997-07.

Qian, G., Tang, M., Tong, R., Zhang, X. and Pan, R., 2015. Interactive mesh cloning driven by boundary loop. *The Visual Computer*, pp.1-9.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), pp.257-286.

Reeb, G., 1946. Sur les points singuliers d'une forme de Pfaff completement intégrable ou d'une fonction numérique (On the singular points of a completely integrable Pfaff form or of a numerical function). *CR Acad. Sci. Paris, 222*(847-849), p.2.

Rivers, A., Durand, F. and Igarashi, T., 2010. 3D modelling with silhouettes(Vol. 29, No. 4, p. 109). ACM.

Rogers, D.F., 2000. *An introduction to NURBS: with historical perspective*. Elsevier.

Rudin, L.I., Osher, S. and Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena, 60*(1), pp.259-268.

Schmid, C. and Mohr, R., 1997. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence,19*(5), pp.530-534.

Schmidt, R., Grimm, C. and Wyvill, B., 2006. Interactive decal compositing with discrete exponential maps. *ACM Transactions on Graphics (TOG)*, 25(3), pp.605-613.

Schmidt, R., Khan, A., Singh, K. and Kurtenbach, G., 2009, December. Analytic drawing of 3d scaffolds. In *ACM Transactions on Graphics (TOG)*(Vol. 28, No. 5, p. 149). ACM.

Schmidt, R., Khan, A., Singh, K. and Kurtenbach, G., 2009, December. Analytic drawing of 3d scaffolds. In *ACM Transactions on Graphics (TOG)* (Vol. 28, No. 5, p. 149). ACM.

Sheppard, J., 1991. Drawing the Living Figure:A Complete Guide to Surface Anatomy. *Courier Corporation.*

Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T., 2004, June. The princeton shape benchmark. In *Shape modelling applications, 2004. Proceedings* (pp. 167-178). IEEE.

Singh, C., 2012. Local and global features based image retrieval system using orthogonal radial moments. *Optics and Lasers in Engineering, 50*(5), pp.655-667.

Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C. and Seidel, H.P., 2004, July. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (pp. 175-184). ACM.

Spencer, S., 2011. *ZBrush Character Creation: Advanced Digital Sculpting*. John Wiley & Sons.

Sun, X., Rosin, P.L., Martin, R.R. and Langbein, F.C., 2009. Bas-relief generation using adaptive histogram equalization. *Visualization and Computer Graphics, IEEE Transactions on, 15*(4), pp.642-653.

Takayama, K., Panozzo, D., Sorkine-Hornung, A. and Sorkine-Hornung, O., 2013. Sketch-based generation and editing of quad meshes. *ACM Transactions on Graphics (TOG), 32*(4), p.97.

Takayama, K., Schmidt, R., Singh, K., Igarashi, T., Boubekeur, T. and Sorkine, O., 2011, April. Geobrush: Interactive mesh geometry cloning. In *Computer Graphics Forum* (Vol. 30, No. 2, pp. 613-622). Blackwell Publishing Ltd.

Tangelder, J.W. and Veltkamp, R.C., 2008. A survey of content based 3D shape retrieval methods. *Multimedia tools and applications, 39*(3), pp.441-471.

Tatsuma, A. and Aono, M., 2009. Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. *The Visual Computer, 25*(8), pp.785-804.

Teague, M.R., 1980. Image analysis via the general theory of moments*.*JOSA, 70*(8), pp.920-930.

Thompson, D.W., 1942. On growth and form. On growth and form.

Torrance, K.E. and Sparrow, E.M., 1967. Theory for off-specular reflection from roughened surfaces. *JOSA, 57*(9), pp.1105-1112.

Trimble. 2016. *Sketchup* (2016). [Software]

Von F, Z., 1934. Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode. *Physica, 1*(7-12), pp.689-704.

Vranić, D.V. and Saupe, D., 2004. 3D model retrieval. *Proc. SCCG 2000*, pp.3-6.

Vranić, D.V., Saupe, D. and Richter, J., 2001. Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on* (pp. 293-298). IEEE.

webfox100. 2009. *How to draw comics the marvel way 1*. [Online]. [Accessed 20th May 2015]. Available at: https://www.youtube.com/watch?v=f3y6k_vk_Vg

Wei, C.H., Li, Y., Chau, W.Y. and Li, C.T., 2009. Trademark image retrieval using synthetic features for describing global shape and interior structure. *Pattern Recognition, 42*(3), pp.386-394.

Williams, L.R. and Hanson, A.R., 1994, June. Perceptual completion of occluded surfaces. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on* (pp. 104-112). IEEE.

Williams, L.R., 1994, May. Topological reconstruction of a smooth manifold-solid from its occluding contour. In *European Conference on Computer Vision* (pp. 36-47). Springer Berlin Heidelberg.

Wu, T.P., Sun, J., Tang, C.K. and Shum, H.Y., 2008, December. Interactive normal reconstruction from a single image. In *ACM Transactions on Graphics (TOG)* (Vol. 27, No. 5, p. 119). ACM.

Wu, T.P., Tang, C.K., Brown, M.S. and Shum, H.Y., 2007, August. ShapePalettes: interactive normal transfer via sketching. In *ACM Transactions on Graphics (TOG)* (Vol. 26, No. 3, p. 44). ACM.

Wuhrer, S. and Shu, C., 2012, May. Shape from suggestive contours using 3D priors. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on*(pp. 236-243). IEEE.

Wuhrer, S., Xi, P. and Shu, C., 2012. Human shape correspondence with automatically predicted landmarks. *Machine Vision and Applications, 23*(4), pp.821-830.

Xu, B., Chang, W., Sheffer, A., Bousseau, A., McCrae, J. and Singh, K., 2014. True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Transactions on Graphics, 33*(4).

Yang, L. and Han, J., 2007. Fast algebra algorithm of shape-from-shading with specular reflectance. *Chinese Optics Letters, 5*(4), pp.204-207.

Yang, M., Kpalma, K. and Ronsin, J., 2008. A survey of shape feature extraction techniques. *Pattern recognition*, pp.43-90.

Yang, Y., Lin, H. and Zhang, Y., 2007. Content-based 3-D model retrieval: A survey. Systems, Man, and Cybernetics, *Part C: Applications and Reviews, IEEE Transactions on, 37*(6), pp.1081-1098.

Yap, P.T., Paramesran, R. and Ong, S.H., 2003. Image analysis by Krawtchouk moments. *IEEE transactions on image processing*, *12*(11), pp.1367-1377.

Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B. and Shum, H.Y., 2004. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics (TOG)*, 23(3), pp.644-651.

Zaharia, T. and Preteux, F.J., 2001, May. 3D-shape-based retrieval within the MPEG-7 framework. In *Photonics West 2001-Electronic Imaging* (pp. 133-145). International Society for Optics and Photonics.

Zarins, U., Kondrats, S., 2014. Anatomy for Sculptors, Understanding the Human Figure. *Exonicus LLC*

Zeleznik, R.C., Herndon, K.P. and Hughes, J.F., 2007, August. SKETCH: an interface for sketching 3D scenes. In *ACM SIGGRAPH 2007 courses* (p. 19). ACM.

Zhang, D. and Lu, G., 2001, June. A comparative study on shape retrieval using Fourier descriptors with different shape signatures. In *Proc. International Conference on Intelligent Multimedia and Distance Education (ICIMADE01)*.

Zhang, D. and Lu, G., 2004. Review of shape representation and description techniques. *Pattern recognition, 37*(1), pp.1-19.

Zhang, G., Ma, Z.M., Niu, L.Q. and Zhang, C.M., 2012. Modified Fourier descriptor for shape feature extraction. *Journal of Central South University, 19*(2), pp.488-495.

Zhang, J., Siddiqi, K., Macrini, D., Shokoufandeh, A. and Dickinson, S., 2005, November. Retrieving articulated 3-d models using medial surfaces and their graph spectra. In *Energy minimization methods in computer vision and pattern recognition* (pp. 285-300). Springer Berlin Heidelberg.

Zhang, L., da Fonseca, M.J., Ferreira, A. and e Recuperação, C.R.A., 2007.Survey on 3D shape descriptors. *Technical Report, DecorAR (FCT POSC/EIA/59938/2004).*

Zhang, R., Tsai, P.S., Cryer, J.E. and Shah, M., 1999. Shape-from-shading: a survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(8), pp.690-706.